NOVA IMS Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

COMPARING MULTIMODAL LLMS AND TRADITIONAL NEURAL NETWORKS FOR TABLE EXTRACTION FROM PDFS AND IMAGES

An Evaluation of Structure and Content Extraction from Table in Images

Guilherme Guerra Marques Nunes

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade NOVA de Lisboa

COMPARING MULTIMODAL LLMS AND TRADITIONAL NEURAL NETWORKS FOR TABLE EXTRACTION FROM PDFS AND IMAGES

by

Guilherme Guerra Marques Nunes

Master Thesis presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Specialization in Data Science

Supervised by:

Márcia Lourenço Baptista, PhD, Adjunct Professor, NOVA Information Management School

July, 2025

STATEMENT OF INTEGRITY

I, Guilherme Nunes, hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Portugal, 15/07/2025

Guilherme Guerra Marques Nunes

Comparing Multimodal LLMs and Traditional Neural Networks for Table Extraction from PDFs and Images

An Evaluation of Structure and Content Extraction from Table in Images

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.2.1) (Lourenço, 2021).

Acknowledgements

I would like to express my gratitude to Professor Márcia and Dr Vitor for their guidance throughout my thesis and for their valuable insights on how to write and analyse the data. I would also like to thank Fraunhofer AICOS for providing the research grant that supported my thesis. A special thanks goes to the NLP team—Vitor Rolla, Duarte Pereira, and Vasco Gomes—who taught me a great deal and helped me revise my thesis. One outcome of this work was a paper that was accepted for poster presentation at the 1st Joint Workshop on Large Language Models and Structure Modelling (XLLM 2025) at ACL 2025. This achievement would not have been possible without the collaboration of everyone mentioned above.

Abstract

Extracting tables from images, such as cropped sections from PDFs or screenshots of spreadsheets, remains a challenging task due to the variability in table layouts and the absence of structural metadata. Traditional OCR-based systems, like the Table Transformer (TATR) combined with PaddleOCR, rely on explicit structure detection and text recognition. More recently, multimodal Large Language Models (LLMs) such as GPT-4o, GPT-4o Mini, Granite Vision, and PHI-3 Vision have introduced an alternative approach, generating structured outputs directly from images without relying on traditional OCR pipelines. This thesis compares both strategies using 2,000 annotated tables from the PubTables-1M dataset, evenly split between simple and complex cases. Evaluation focuses on structural accuracy, content fidelity, and layout robustness, with GriTSCon used as a unified metric. Results show that GPT-4o performs best among multimodal LLMs on simple tables (GriTSCon F1 = 89.6%), while TATR-OCR outperforms all models on complex tables (GriTSCon F1 = 85.5%). GPT-4o achieves higher cell-content accuracy at exact-match thresholds on simple layouts but experiences a performance drop of 17 points when handling complex structures. In contrast, TATR-OCR maintains high accuracy across both scenarios, with low failure rates and stable structure recognition. These findings highlight the limitations of current multimodal LLMs in complex visual tasks and support the potential of hybrid approaches that combine the strengths of OCR-based systems with LLM reasoning capabilities.

Keywords: Table Extraction, Optical Character Recognition, Multimodal Large Language Models, Table Transformer, GriTSCon

Sustainable Development Goals (SDG):

Resumo

A extração de tabelas a partir de images, como recortes de PDFs ou capturas de tela de planilhas, continua sendo uma tarefa desafiadora devido à variabilidade dos layouts e à ausência de metadados estruturais. Sistemas tradicionais baseados em OCR, como o Table Transformer (TATR) combinado com o PaddleOCR, dependem da detecção explícita da estrutura e do reconhecimento de texto. Mais recentemente, modelos de linguagem multimodal de grande escala , como o GPT-4o, GPT-4o Mini, Granite Vision e PHI-3 Vision, introduziram uma abordagem alternativa, gerando saídas estruturadas diretamente a partir de imagens, sem utilizar um pipeline OCR tradicional. Esta dissertação compara essas duas estratégias utilizando 2.000 tabelas anotadas do conjunto de dados PubTables-1M, divididas igualmente entre casos simples e complexos. A avaliação foca na precisão estrutural, fidelidade do conteúdo e robustez do layout, utilizando o GriTSCon como métrica unificada. Os resultados mostram que o GPT-4o apresenta o melhor desempenho entre os MLLMs em tabelas simples (GriTSCon F1 = 89,6%), enquanto o TATR-OCR supera todos os modelos em tabelas complexas (GriTSCon F1 = 85,5%). O GPT-4o alcança maior precisão de conteúdo em limiares de correspondência exata em layouts simples, mas sofre uma queda de aproximadamente 17 pontos em estruturas mais complexas. Em contraste, o TATR-OCR mantém alta precisão em ambos os cenários, com baixas taxas de falha e reconhecimento estrutural estável. Esses resultados destacam as limitações dos LLMs atuais em tarefas visuais complexas e apontam para o potencial de abordagens híbridas que combinam as vantagens dos sistemas baseados em OCR com a capacidade de raciocínio dos LLMs.

Palavras-chave: Extração de Tabelas, Reconhecimento Óptico de Caracteres, Modelos Multimodais de Linguagem,Table Transformer, GriTSCon

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

CSV Comma-Separated Values *(p. 8)*

DETR DEtection TRansformer *(p. 5)*
DOM Document Object Model *(p. 3)*

EDD Encoder Dual Decoder *(p. 5)*

FA Functional Analysis *(p. 5)*

GriTS Grid Table Similarity *(p. 11)*
GriTSCon Grid Table Similarity Considering Content *(p. 1)*

HTML HyperText Markup Language *(p. 3)*

LCS Longest Common Subsequence *(p. 13)*
LLMs Large Language Models *(p. 1)*

NL2SQL Natural Language to SQL *(p. 9)*

OCR Optical Character Recognition *(p. 1)*

PDF Portable Document Format *(p. 1)*
PMCOA PubMed Central Open Access *(p. 15)*
POS Plan-of-SQLs *(p. 9)*

SUC Structural Understanding Capability *(p. 10)*

TATR Table Transformer *(p. 1)*
TD Table Detection *(p. 5)*

TEDS Tree-Edit-Distance-based Similarity *(p. 11)*
TSR Table Structure Recognition *(p. 5)*

1

Introduction

The growing number of digital documents, such as reports, scientific articles, and administrative records, has made extracting useful information more complex. Tables are a common element in these documents as they organise data clearly and structurally. However, automatically detecting and extracting both the structure and content of tables from documents, especially when they are stored as images or scanned Portable Document Format (PDF)s, remains a challenging task.

Traditional approaches rely on Optical Character Recognition (OCR) systems combined with deep learning models to detect text and identify table structures. These methods often involve multiple steps, including text detection, layout analysis, and cell parsing. Although these methods have improved over time, they still encounter difficulties when dealing with complex table layouts, such as merged cells and projected rows.

Recently, multimodal Large Language Models (LLMs) have introduced new possibilities for this task. These models can process both text and images simultaneously, generating structured table outputs directly from images. This development could simplify the extraction process and enhance performance, particularly for simpler tables.

This thesis compares two different strategies for extracting tables from images, such as cropped tables from PDFs. The first strategy is based on a deep learning model, Table Transformer (TATR), combined with OCR to detect and read table cells. The second approach utilises four multimodal LLMs—GPT-4o, GPT-4o Mini, Phi-3 Vision, and Granite Vision—to generate structured data directly from table images.

The comparison focuses on accuracy in structure and text content, as well as the ability to handle both simple and complex table layouts. It also examines inference failure and time efficiency. Evaluation metrics include table shape accuracy, F1 score at various similarity thresholds, and the Grid Table Similarity Considering Content (GriTSCon) metric. A subset of annotated table images from the PubTables-1M dataset is used for the experiments.

The second chapter of this thesis reviews existing work on table extraction, covering

both traditional OCR methods and multimodal language models. The third chapter explains data preparation, model setup, and evaluation procedures. The fourth chapter presents the results of the experiments and compares the performance of the models. The final chapter discusses the main findings and proposes future directions for improving table extraction using multimodal models.

This work aims to enhance our understanding of the strengths and weaknesses of traditional OCR-based systems in comparison to newer multimodal LLMs and to demonstrate how these approaches perform in various table extraction scenarios.

2

Literature Review

This literature review is structured in four sections. The first explains table extraction fundamentals. The second covers OCR-based methods. The third examines multimodal LLMs for tables, including reasoning and generation. The fourth presents evaluation metrics and benchmarking datasets. The keywords used to search the citation are in the Table 2.1.

Table 2.1: Keywords used in the search of the sources for this literature review.

	Keywords	
Optical Character Recognition	Large Language Models	Table Extraction
Computer Vision	Evaluation Metrics	Table Detection
Datasets	PDFs	Images

2.1 Fundamentals of Table Extraction

Extracting structured information from tables is critical for fields such as scientific research, financial analysis, and automated data processing. Tables provide a structured layout, making them an effective method for organising and retrieving information (Burdick et al., 2020; L. Chen et al., 2023). Manual extraction of tables from PDFs and images remains a tedious process due to their unstructured nature, prompting the development of automated solutions as summarised by (Musa et al., 2023).

Early foundational work by Pinto et al. (2003) conceptualised table extraction as a six-stage process: locating tables, identifying row and column positions, segmenting into cells, tagging headers, and associating data cells with the appropriate headers. Subsequent analyses categorise extraction from HyperText Markup Language (HTML), where the Document Object Model (DOM) schema aids structure—versus plain-text tables, where layout cues must be inferred (Gatterbauer et al., 2007; Kashinath et al., 2022).

Research on web tables has progressed over nearly two decades from basic extraction and classification to advanced semantic interpretation and integration into knowledge-based applications, as noted by (Zhang & Balog, 2020). Initially focused

on identifying tables on webpages, the scope broadened to include spreadsheets and detailed taxonomies. Improved extraction methods led to large-scale table corpora, though these often become static and outdated.

Currently, research is shifting towards table interpretation, which seeks to identify schema elements and extract relational data. However, challenges remain, including low recall rates, under-represented table types, and difficulties in linking tabular content to external knowledge bases. These issues highlight ongoing research opportunities in semantic modelling and robust interpretation techniques (Zhang & Balog, 2020).

2.2 OCR and Deep Learning models for Table Extraction

This section explores two complementary classes of methods for extracting tabular information from images and scanned documents. First, it introduces OCR as a foundational tool for transcribing visual text into a machine-readable format. Then, it examines deep-learning–based models designed specifically for detecting table structure and content. Finally, it discusses the current limitations of these approaches and highlights emerging trends aimed at improving robustness and structural understanding in table extraction pipelines.

2.2.1 Optical Character Recognition

The ability to extract information from tables within images, such as those found in PDFs or medical records, is crucial across numerous domains, including scientific research, financial analysis, and data entry automation (Y. Li et al., 2024). Traditional table extraction relies heavily on OCR as a fundamental preprocessing step, laying the groundwork for subsequent tasks such as structure recognition, content extraction and evaluation (Y. Li et al., 2024; Smock et al., 2023a, 2023b). OCR, in essence, transcribes image-based text into a machine-readable format, thereby unlocking the information contained within documents, photographs, and digital images (Ranjan et al., 2021; Zhong et al., 2020).

Although OCR technology has been available since the 1990s, with tools such as Tesseract OCR (Smith, 2007), significant challenges remain in extracting information from tables. Traditional OCR generally involves a two-stage process: first, detecting text areas, and then recognising the characters within those areas Ranjan et al. (2021).

Traditional OCR pipelines typically involve separate modules for text detection, text recognition, and parsing as explained in the research review by Surana et al. (2022). As a result, additional effort is required to ensure overall system performance (Y. Li et al., 2024). Such approaches often focus on general text extraction and may not address the specific challenges of extracting tabular information from PDFs or images, given the complexity of table structures as pointed out in the conclusion of Ranjan et al. (2021).

However, modern OCR systems now enable the extraction of bounding box information for words, capturing their precise location within an image (Du et al., 2020). This advancement presents a promising method for integrating OCR with table structure recognition techniques. By combining the spatial data of table cells with the extracted words, it becomes possible to reconstruct tables in a machine-readable format while preserving the original structure.

Applying OCR to tables in unstructured documents, such as PDFs and images, poses distinct challenges. While accuracy has improved, factors such as noisy images, diverse fonts, and complex layouts can still hinder performance (Sharma, 2023). Errors at this stage may cascade, adversely affecting the entire table extraction process and leading to incorrect data retrieval.

2.2.2 Table Structure Detection Models

Several models have been developed to detect tables and their structures. One notable example is TableNet, introduced by Paliwal et al. (2019). This end-to-end deep learning model is designed specifically for table detection and structure recognition. TableNet utilises a base network initialised with pre-trained VGG-19 features and features two decoder branches: one for segmenting the table region and another for segmenting the columns. This design effectively exploits the interdependence between table detection and structure recognition, allowing the model to achieve state-of-the-art performance on benchmark datasets.

Another approach is the Encoder Dual Decoder (EDD) model, proposed by Zhong et al. (2020), which uses an attention-based framework to convert table images into HTML code. The EDD model employs a structure decoder to reconstruct the table architecture and a cell decoder to recognise cell content accurately.

Microsoft introduced a model called TATR, described by Smock et al. (2022) that performs Table Detection (TD), Table Structure Recognition (TSR), and Functional Analysis (FA) in a unified framework. They trained a DEtection TRansformer (DETR) model for TD and a modified DETR model for TSR and FA jointly. These models achieve state-of-the-art performance without requiring task-specific modifications, demonstrating the suitability of the DETR architecture for table extraction. The models are available on Microsoft's Hugging Face repositories: Table Structure Recognition (https://huggingface.co/microsoft/table-transformer-structure-recognition) and for Table Detection (https://huggingface.co/microsoft/table-transformer-detection).

2.2.3 Challenges in Table Extraction

While OCR effectively transcribes text, extracting structured tables introduces unique challenges. Pinto et al. (2003) defined table extraction as a six-step process:

1. Locate the table.
2. Identify row positions and types.
3. Identify column positions and types.
4. Segment the table into cells.
5. Tag headers versus data cells.
6. Associate data cells with headers.

In structured HTML tables, this process is simplified due to predefined markup (Gatterbauer et al., 2007; Kashinath et al., 2022), but unstructured tables in PDFs or images present additional complexities (Zhang & Balog, 2020). For instance, poor markup consistency in HTML can make table localisation difficult, even when the underlying formatting suggests structure.

Table structure recognition models aim to detect and spatially organise table elements, thus overcoming OCR's limitations (Paliwal et al., 2019; Smock et al., 2022, 2023a, 2023b; Zhong et al., 2020).

2.2.4 Discussion

OCR and deep-learning models have significantly advanced the extraction of tabular information from scanned documents. Traditional OCR systems convert images, such as PDFs or medical records, into machine-readable text. Yet, while effective at basic recognition, conventional OCR often struggles with complex structures, especially in tables with irregular layouts, overlapping cells, and nested headers. Recent developments have partially addressed these issues: modern OCR engines (for instance, the advanced OCR toolkit called PaddleOCR, which uses the PP-OCR model developed by Du et al. (2020)) now extract bounding box information to capture the precise spatial location of words. This spatial metadata is crucial for accurately reconstructing the table layout.

Deep-learning models, such as TableNet and Transformer-based approaches (often referred to as TATR), extend the capability of traditional OCR. These models do more than recognise text; they capture the hierarchical structure of tables by identifying rows, columns, merged cells, and headers. As a result, they help reconstruct tables in a format that retains both their structure and content. For example, DETR-based transformers have advanced the state of the art by jointly optimising for table detection and cell segmentation, thereby mitigating issues where OCR might miss subtle layout cues.

Despite these promising advances, challenges persist. Errors generated during text detection can propagate through downstream processes (e.g., header tagging and cell segmentation), ultimately undermining the extraction and interpretation of complex

tables. Noisy images, diverse fonts, and inconsistent layouts continue to hamper performance, and the computational cost remains significant, particularly for large-scale or multi-domain document processing. These challenges suggest that relying solely on traditional OCR or any single deep-learning model might not be sufficient.

Future research should thus focus on integrating multiple modalities. A promising direction involves combining OCR's spatial precision with NLP-based reasoning and vision models. Such multimodal approaches could leverage contextual understanding to resolve ambiguities in complex table layouts, improving both the reliability of cell segmentation and the accuracy of text recognition. Domain-specific model training and fine-tuning, particularly for specialised contexts (like financial reports or scientific literature), could further enhance performance.

In summary, while OCR combined with deep-learning models trained for this task has significantly improved table extraction, there remains room for further development. A continued focus on multimodal, hybrid architectures that integrate OCR capabilities with semantic reasoning will be essential to overcome current limitations and meet real-world document processing demands; for example, Soric's recent master's thesis demonstrated novel pipelines for extracting structured tables from unstructured PDFs while preserving uncertainty (Soric, 2025).

2.3 Multimodal LLMs for Table Extraction

This section is organised into three parts. First, a review of studies that explore the use of multimodal LLMs for table extraction and related tasks, such as table reasoning, is presented. Next, key challenges encountered when applying these models to extract and interpret complex tabular data are examined. Finally, a discussion is provided that synthesises current findings and suggests promising directions for future research.

2.3.1 Multimodal Large Language Models

The limitations of traditional OCR-based table extraction have spurred the exploration of alternative methods, with multimodal LLMs emerging as a particularly promising avenue (Sui et al., 2024). These models can process both textual and visual information, enabling them to analyse table images or their textual representations directly, and thereby potentially bypass the need for a separate OCR stage. This paradigm shift promises extraction solutions that are more efficient, accurate, and robust.

Multimodal LLMs have attracted considerable attention not only for table extraction but also for a range of related automatable tasks. For instance, Cheng et al. (2025) provided an extensive review on table mining with LLMs, demonstrating that these models are effective in both table extraction and table reasoning. Also, Dong and Wang (2024) offers a comprehensive survey of advances, challenges, and opportunities in applying state-of-the-art LLMs to tabular data. They introduce methods for prompting

and training LLMs on table interpretation, processing, reasoning, analytics, and generation, equipping researchers and practitioners with the tools to fully unlock LLMs' potential for table-centric tasks.

A key advantage of LLMs is their capacity for generalisation and the effective use of contextual cues for complex reasoning. However, they can struggle with intricate table structures, especially when cell layouts are disrupted (T. Liu et al., 2023).

In a related line of work, (Nam, Song, et al., 2024) introduces Prompt to Transfer (P2T), a tabular transfer-learning framework that exploits LLMs' in-context learning to bridge heterogeneous source and target tables. P2T first uses the LLM to pinpoint the column most predictive of the downstream task, then generates "pseudo-demonstrations" by serialising those source rows into natural-language prompts alongside few-shot examples. Although P2T consistently outperforms both PDF extractors and conventional few-shot learners, it remains bounded by LLM prompt-length limits and can exhibit regressions, especially when asked to produce precise numerical values.

In another study (Nam, Kim, et al., 2024), the same author proposes OCTree, a tabular feature generation framework that leverages LLMs not just for classification but as optimisers for rule-based feature construction. OCTree uses decision tree reasoning extracted from model training to iteratively guide the LLM in generating meaningful new column features, enabling language-guided exploration without requiring a predefined rule space. This optimisation loop, powered by natural language prompts and scored feedback, demonstrates strong performance across classification and regression tasks, even in datasets lacking semantic labels. OCTree highlights a shift toward LLMs as collaborative agents in automated data preparation pipelines, extending their utility far beyond traditional inference.

Further, (Zha et al., 2023) introduced TableGPT, which integrates a dedicated table encoder that captures semantic and numerical patterns across entire tables. It transforms user instructions into structured command sequences that facilitate tasks like chart generation, natural language responses, or analytical queries. Building upon this foundation, TableGPT2 by (Su et al., 2024) demonstrates improvements in encoding both schema-level and fine-grained cell content, thereby enhancing the model's performance on complex tables, ambiguous queries, and tables with missing metadata. Trained on hundreds of thousands of real-world examples, it demonstrates strong capability across a wide range of tabular reasoning tasks, outperforming general-purpose LLMs on several evaluation benchmarks. Together, these models represent a shift toward LLMs that are not only text-aware but table-aware, bridging the gap between natural language understanding and structured data interpretation.

Singha et al. (2023) investigated how the representation of tables in prompts affects the performance of LLMs. Their study evaluated eight standard table formats, including DFLoader, JSON, Data-Matrix, Markdown, Comma-Separated Values (CSV), and Tab-Separated Values, across a suite of self-supervised structural understanding tasks, such as navigation, column lookup, and table transposition. They also introduced

eight noise-inducing operations, inspired by real-world messy or adversarial data, such as shuffled headers, merged columns, and serialised rows. Their findings revealed that both the choice of table format and the presence of structural noise significantly impact the performance of LLMs. This highlights the brittleness of LLMs in response to changes in layout and representation.

Another study by T. Liu et al. (2023) examined three dimensions of LLM performance on tabular data: the impact of structural perturbations, the comparative performance of textual versus symbolic reasoning, and the benefits of ensemble prompting strategies. Their findings revealed that even minor structural disruptions could notably degrade performance, especially for tasks requiring arithmetic or symbolic reasoning, although their exclusive reliance on GPT-3.5 poses limitations for generalisation.

In addition to these insights, Lu et al. (2025) offers a comprehensive survey on the applications of LLMs and visual language models in table processing tasks. This includes traditional applications such as table question answering and Natural Language to SQL (NL2SQL), as well as newer areas like spreadsheet manipulation and image-based table extraction. The study categorises methods into four key dimensions: data representation, training strategies, prompting techniques, and agent-based planning. It also highlights several challenges faced in this field, including the structural complexity of tables, the diversity of user inputs, inefficiencies in current prompting methods, and the lack of robust real-world benchmarks.

W. Chen (2023) was among the first to demonstrate that LLMs can act as effective few-shot reasoners for table-related tasks. His work showed that even a single example, when combined with chain-of-thought prompting, significantly enhances a model's capacity to reason over complex table structures without extensive specialised training. Interpretability is also an important consideration. Nguyen et al. (2025) introduced the Plan-of-SQLs (POS) approach, a new Table Question and Answering method that makes the model's decision-making process interpretable, to clarify how LLMs make decisions when processing tabular data, achieving up to 90% agreement with human evaluators. Also,I. Deng et al. (2025) investigated how LLMs can handle temporally evolving tables, thereby addressing an additional level of complexity in dynamic data environments.

Multimodal LLMs have generated significant interest in tasks that go beyond table extraction. As reviewed by M. Zheng et al. (2024), these models can handle various tabular tasks and even bypass traditional OCR pipelines for more accurate extraction (Sui et al., 2024). For instance, models such as LLaVA (H. Liu et al., 2023) and GPT-4o (OpenAI et al., 2024) integrate image and text processing to enhance their table recognition capabilities. In their analysis of the GPT family, (Yenduri et al., 2023) compare and explain each version from GPT-1 to GPT-4, summarising their findings in a table that highlights the challenges, including the difficulty of understanding unstructured data. Advanced prompting techniques, such as chain-of-thought, have further refined these models' abilities to understand structure (N. Deng et al., 2024; Sui

et al., 2024).

GPT-4 Omni (GPT-4o), launched by OpenAI in May 2024, represents a significant leap forward with a parameter count estimated to exceed 1 trillion—vastly outnumbering GPT-3 (175 billion) and GPT-1 (117 million) (Shahriar et al., 2024). Its integrated architecture processes text, audio, and images at high speeds and is pre-trained on extensive data up to October 2023. This enables GPT-4o to generate detailed, structured outputs that capture complex table information. Another notable model is Table LLaVA (M. Zheng et al., 2024), which is the LLava model (H. Liu et al., 2023) fine-tuned on the MMTab dataset to perform table-based question answering and data interpretation. However, Table LLaVA is primarily designed for single-table scenarios in English and operates on relatively low-resolution inputs. In contrast, MiniCPM-V (Yao et al., 2024) supports high-resolution images (up to 1.8 million pixels) and offers robust OCR capabilities along with multilingual support.

The Phi-3 Vision model (Microsoft, 2024) from Microsoft can process both visual and textual cues simultaneously. Although primarily trained on English and low-resolution images (similar to Table LLaVA), it often outperforms earlier models in key benchmarks. Moreover, Granite Vision 3.2 (GraniteVision, 2025), an open-source model trained on approximately 13 million images with 80 million multimodal instructions, has demonstrated exceptional performance in extracting structured information from tables, charts, and infographics by processing medium-resolution images that strike a balance between detail and efficiency.

2.3.2 Representation, Prompting, and Structural Understanding

A critical element in leveraging multimodal LLMs for table extraction is the method of data representation and the design of input prompts. Research indicates that whether tables are converted into text-based formats (like JSON or CSV) or maintained as images enriched with visual cues (such as colour highlighting), the chosen representation profoundly influences the model's ability to extract both structure and content (N. Deng et al., 2024; Sui et al., 2024). Techniques like chain-of-thought prompting enable models to reason through the table extraction process step-by-step. In contrast, self-augmented prompting, where the model generates intermediate structural cues based on its latent knowledge,further enhances accuracy in tasks like cell lookup, row retrieval, and overall layout reconstruction.

Evaluations often utilise benchmarks such as the Structural Understanding Capability (SUC) benchmark (Sui et al., 2024), which assesses performance across dimensions, including table partitioning, size detection, and cell-level reasoning. Although many studies have utilised GPT-3.5 for comprehensive experiments, GPT-4 is often employed for validation due to its superior performance, despite higher computational demands.

2.3.3 Discussion

Despite significant progress, preserving the fine-grained structural integrity of complex tables remains a challenge. While multimodal LLMs perform remarkably well in generating detailed cell content and maintaining precise structural configurations, particularly in the presence of merged or spanning cells, maintaining precision remains a significant hurdle. A promising future direction is the development of hybrid architectures that merge the spatial precision of traditional deep learning (TATR) methods with the generative and reasoning strengths of multimodal LLMs. Incorporating a TATR-based pre-processing step to extract spatial cues could guide the LLMs output, thereby reducing structural hallucinations and improving overall accuracy.

Additionally, exploring cross-modal pretraining strategies that jointly embed visual and textual features from the outset may reduce the fine-tuning burden and enhance overall robustness. Establishing unified, benchmark-driven evaluation frameworks that concurrently assess both structural fidelity and semantic accuracy will be crucial for effective evaluation. Such comprehensive frameworks can ultimately drive improvements in prompt design, model architecture, and real-world applications.

In conclusion, while multimodal LLMs hold significant promise for revolutionising table extraction and reasoning, challenges, particularly those associated with representation, complex structural interpretation, and computational scalability, remain. Addressing these issues through innovative prompting strategies, hybrid model architectures, and integrated evaluation frameworks is essential for fully harnessing the potential of these advanced models.

2.4 Evaluation Metrics

This section is divided into three subsections. The first subsection discusses traditional metrics used for evaluating table extraction. The subsequent subsections describe advanced metrics, Tree-Edit-Distance-based Similarity (TEDS), and Grid Table Similarity (GriTS). Finally, a discussion compares the advantages and limitations of these approaches.

2.4.1 Traditional Metrics

Traditional metrics provide a straightforward evaluation of table extraction performance and serve as useful baselines. One commonly used measure is table shape accuracy, which assesses whether the predicted table dimensions (i.e., the number of rows and columns) match those of the ground truth. Table shape accuracy is typically computed as the harmonic mean of row accuracy and column accuracy. Row accuracy penalises cases where rows are omitted or added, and column accuracy does likewise for columns. In some evaluations, these differences are also reported directly as structural errors, such as the number of missing or extra rows and columns, to offer more

interpretable insights into over- or under-segmentation issues. Although these metrics do not capture finer cell-level errors, they offer an essential gauge of the overall layout preservation (Reddy, 2025).

In addition to layout-based metrics, standard evaluation measures such as precision, recall, and the F1 score are commonly applied at the cell level. Precision quantifies the proportion of correctly extracted cells among all cells predicted by the model, while recall measures the proportion of ground-truth cells that are successfully extracted. The F1 score, defined as the harmonic mean of precision and recall, provides a balanced evaluation that considers both false positives and false negatives. The relevance and application of the F1 score across various extraction tasks have been well documented by Hand et al. (2021).

Moreover, to capture the nuances of text extraction accuracy, F1 scores are computed under different threshold settings. A threshold of 0 focuses solely on structural layout (ignoring cell content), while thresholds such as 60, 80, and 100 measure increasingly strict levels of textual matching. For example, a threshold of 60 tolerates minor variations or typographical errors, whereas a threshold of 100 requires exact text matches. This threshold-based approach permits a more detailed analysis of both the structural and content-based performance of table extraction systems.

2.4.2 Tree-Edit-Distance-based Similarity

Zhong et al. (2020) introduces TEDS, a metric designed to evaluate how well a model can recognise tables from images by comparing the recognised table to the original table. Tables are commonly used to summarise data in a structured format, making it easier for people to find and compare information. However, it can be challenging for machines to comprehend tables, particularly in unstructured formats such as images. Table recognition is, therefore, the process of converting tables into a machine-readable format.

TEDS measures the similarity between two tables by calculating the difference between them. The goal is to determine how accurately a model has reconstructed a table's structure and content from an image. Traditional metrics for table recognition had shortcomings. Some metrics only checked the immediate relationships between table cells, making it difficult to detect errors caused by misaligned cells or empty cells. Other metrics lacked a reliable way to measure the performance of cell content recognition.TEDS was therefore developed to address these limitations.

TEDS works by comparing the structure of the recognised table to the original table. It represents both tables as tree structures that capture the relationships between table elements such as headers, rows, and cells. It then calculates the "edit distance" between the two trees by counting the number of changes (insertions, deletions, substitutions) needed to transform one tree into the other. The cost of substitutions is determined by how different the content of the cells is and whether the row and column spans of the

cells differ. Finally, the TEDS score is calculated based on this edit distance, resulting in a similarity score between the two tables.

TEDS examines the entire table structure, rather than just immediate neighbours, enabling it to identify errors related to the overall layout and alignment. It evaluates the cell content using a string similarity measure, allowing for partial matches and capturing recognition errors that are not exact matches. The source indicates that TEDS is superior to the adjacency relation metric because it effectively captures both table structure and cell content errors, whereas the adjacency relation metric either underreacts to structural errors or overreacts to cell content errors.

2.4.3 Grid Table Similarity

Smock et al. (2023b) introduce GriTS, a novel class of metric for TSR evaluation. The key feature of GriTS is that it evaluates the correctness of a predicted table directly in its natural matrix form. To facilitate the comparison of matrices, the researchers generalise the two-dimensional Longest Common Subsequence (LCS) problem to the two-dimensional most similar substructures problem, proposing a polynomial-time heuristic for solving it. This algorithm provides both upper and lower bounds on matrix similarity, and empirical evaluations demonstrate that there is very little practical difference between these bounds.

Unlike prior metrics that use alternative abstract representations for tables, GriTS unifies the assessment of cell topology (GriTSTop), cell content (GriTSCon), and cell location recognition (GriTSLoc) within the same framework. This modular approach simplifies evaluation and enables more meaningful comparisons across different TSR approaches. The authors validate GriTS empirically using a large real-world dataset, confirming that it exhibits more desirable behaviour than alternative metrics for TSR evaluation. Specifically, they demonstrate that GriTS assigns equal importance to rows and columns, and that cells are credited equally regardless of their position within the table.

2.4.4 Discussion

Evaluating Table Structure Recognition is crucial for developing models that extract and interpret tables from images. Traditional metrics, such as table shape accuracy (direct count of correctly extracted rows and columns) and F1 score, have been limited, leading to the creation of more advanced methods. One significant approach is the Tree-Edit-Distance-based Similarity metric. TEDS treats table structures as hierarchical trees, enabling detailed assessments of structural differences and recognition of cell content. Although TEDS improves on earlier methods by calculating "edit distance" to identify alignment errors and mismatches, it relies on an abstract representation that may overlook real-world complexities.

The GriTS metric evaluates tables in their natural matrix form. This approach unifies cell topology, content, and positioning within a flexible framework. By utilising the two-dimensional most similar substructures (2D-MSS) problem, GriTS provides a more intuitive and efficient way to compare predicted and actual tables. Empirical studies indicate that GriTS yields more reliable and interpretable results, treating rows and columns equally and distributing cell importance consistently. Given its comprehensive evaluation capabilities, GriTS may be the optimal choice for assessing table recognition models, enhancing our understanding of their effectiveness in real-world applications.

2.5 Benchmarking Datasets

This section is divided into five subsections. Four subsections describe the different publicly available benchmarking datasets and their characteristics: SciTSR, TableBank, PubTabNet, and PubTables-1M. The last subsection discusses and compares them.

2.5.1 SciTSR

The SciTSR dataset, created by Chi et al. (2019), is a large-scale resource for table structure recognition in PDF files, comprising 15,000 tables with corresponding structure labels collected from scientific papers in LaTeX format on arXiv [1]. These snippets are extracted using regular expressions and compiled into PDFs.

The structure labels include cell content and coordinates (start and end row and column), stored in JSON format. The dataset is divided into 12,000 tables for training and 3,000 for testing. A subset, SciTSR-COMP, features complicated tables with spanning cells; the test set includes 716 such tables.

SciTSR addresses the lack of training data for complicated table structure recognition, distinguishing itself from the ICDAR-2013 dataset, which contains only 156 tables. On average, each table contains about 9 rows, 5 columns, and 48 cells, with 24 per cent considered complicated. The dataset is publicly available and designed to evaluate a novel graph neural network model for this task.

2.5.2 TableBank

M. Li et al. (2020) introduced TableBank, a large image-based dataset for table detection and recognition. Utilising a novel weak supervision approach, TableBank includes 417,234 high-quality labelled tables and their original documents to advance deep learning models for table analysis.

The dataset consists of tables from Word and LaTeX documents, featuring multiple languages in Word and primarily English in LaTeX. Bounding boxes are generated by

[1] arXiv is an open-access repository for scholarly articles in fields such as physics, mathematics, computer science, and more. It hosts preprints—research papers shared publicly before formal peer review and is widely used by researchers to disseminate findings quickly.

modifying the source code of these documents, with annotations included for training table detection models and providing 145,463 instances for table structure recognition.

Covering various domains, including business documents, official filings, and research papers, the paper establishes baseline models using the Faster R-CNN algorithm and an image-to-text model for table recognition. The dataset is split into training, validation, and test sets, with performance metrics like precision, recall, F1 score, and BLEU scores calculated for evaluation. TableBank is publicly available to promote research in table detection and recognition.

In summary, TableBank is a comprehensive dataset designed to support the development of effective deep-learning models for table analysis.

2.5.3 PubTabNet

Zhong et al. (2020) introduced PubTabNet, a large dataset for image-based table recognition containing 568,000 table images with structured HTML representations. This dataset was automatically generated by matching XML and PDF formats from scientific articles in the PubMed Central Open Access (PMCOA), making it the largest publicly available dataset of its kind. The tables come from over 6,000 journals, ensuring a diverse range of styles.

Each image is annotated with its structure and cell content in HTML format, where cells are classified as either headers or body cells. The dataset prioritises a tree structure for web application integration. Table images extracted from PDFs based on XML nodes were created at 72 pixels per inch. To improve data quality, tables with cells spanning more than 10 rows or columns or with inconsistent XML representations were removed. PubTabNet is a resource for enhancing image-based table recognition using deep learning and is useful for pre-training recognition models.

2.5.4 FinTabNet

FinTabNet is a large-scale dataset developed by X. Zheng et al. (2021) to support table structure recognition in visually complex financial documents. It comprises over 110,000 annotated tables extracted from more than 70,000 pages of annual reports published by S&P 500 [2] companies. Unlike tables found in scientific or governmental sources, financial tables often exhibit sparse use of graphical lines, wider spacing between elements, and greater stylistic variation, making them particularly challenging for automated extraction systems.

[2] S&P 500 (Standard & Poor's 500) is a stock market index that tracks the performance of 500 of the largest publicly traded companies in the United States. It is widely used as a benchmark for the overall health of the U.S. equity market.

2.5.5 PubTables-1M

Smock et al. (2022) introduced PubTables-1M, a comprehensive dataset for table extraction from unstructured documents, containing nearly one million tables from scientific articles in the PubMed Central Open Access (PMCOA) database. It supports multiple input modalities and provides detailed annotations for headers and cell locations, making it ideal for various modelling approaches.

This dataset is nearly twice the size of comparable datasets and addresses all three table extraction subtasks: table detection (TD), table structure recognition (TSR), and functional analysis (FA). It features robust annotations, a novel canonicalization process to correct over-segmentation, and quality verification measures to ensure accurate ground truth.

Transformer-based object detection models trained on PubTables-1M have achieved excellent results without special customisation. The dataset also includes spatial information, which provides rich ground-truth dilated bounding boxes to prevent gaps or overlaps, offering more complexity and diversity in table structures compared to previous datasets.

2.5.6 Discussion

PubTables-1M stands out as the most suitable dataset for this thesis due to its large scale. It comprises nearly one million tables, almost double the size of comparable datasets. This extensive collection ensures a comprehensive representation of table structures.

The dataset provides rich ground-truth annotations, including precise cell positions, headers, and content, making it beneficial for training and evaluating models focused on complex table structures. Additionally, it includes OCR ground truth for the entire image, allowing models to learn both the structural and textual aspects of tables.

These attributes make PubTables-1M an ideal resource for advancing research in table extraction and recognition. Given its recent release and robust annotations, it is expected to aid significantly future deep-learning models.

3

Methodology

This chapter details the methodology adopted in this thesis to compare traditional neural network-based approaches and multimodal LLMs for table extraction from PDFs and images. The proposed pipeline integrates data preprocessing, model application, and evaluation metrics to provide a comprehensive assessment of each system's performance.

The chapter begins by describing the dataset selection and preprocessing steps, where raw annotated data is transformed into structured formats suitable for model input and ground truth generation. It then explains the sampling strategy used to create balanced subsets of simple and complex tables for experimentation.

Following this, the chapter outlines the table extraction methods assessed in this study: a conventional OCR-based model (TATR-OCR) and four multimodal LLMs equipped with visual capabilities (GPT-4o, GPT-4o Mini, Granite Vision, and Phi-3 Vision). The discussion covers the implementation pipeline, which includes standardised output formatting and validation, to guarantee fairness and reproducibility in the comparison of models.

Finally, the chapter introduces the evaluation metrics applied to assess structural accuracy and content extraction quality. These include failure rate, structural error analysis, accuracy measures (row, column, and table shape accuracy), F1 score variations across different thresholds, and GriTSCon. Together, these components provide a robust framework for analysing the strengths and limitations of different table extraction approaches.

3.1 Dataset Preprocessing

This section describes the source of the data, the steps taken to combine structural annotations with textual content, and the creation of a clean and well-organised ground truth. The preprocessing pipeline ensures that the models evaluated in this thesis are provided with consistent and accurate input data.

The dataset utilised in this thesis is known as PubTables-1M, as proposed in the

paper by Smock et al. (2022), which provides details on the creation of the dataset, including the combination of manual annotation and automated processes employed to generate the bounding boxes for words and table structures, i.e. the maximum and minimum value for the x and y axis that creates a rectangle. The dataset is available on Hugging Face: https://huggingface.co/datasets/bsmock/pubtables-1m. It is divided into multiple tar.gz files, which are types of compressed archives in a folder; all the descriptions and information of the folders are in Table 3.1.

Table 3.1: Datasets of structure recognition model (947,642 total cropped table instances).

PubTables 1M Structure Dataset Folders (PubTables-1M-Struture_)			
Name Extension	Total Files	Type of File	Description
Table_Words.tar.gz	947,642	JSON	Files containing bounding boxes and text content for all of the words in each cropped table image
Annotations_Test.tar.gz Images_Test.tar.gz	93,834	XML PNG	Files containing bounding boxes in PASCAL VOC format Files containing the cropped image instances
Annotations_Train.tar.gz Images_Train.tar.gz	758,849	XML PNG	Files containing bounding boxes in PASCAL VOC format Files containing the cropped image instances
Annotations_Validation.tar.gz Images_Validation.tar.gz	94,959	XML PNG	Files containing bounding boxes in PASCAL VOC format Files containing the cropped image instances

This section describes the pipeline code to generate the final JSON and CSV files. For this thesis, only the data from the test dataset was processed to create the ground truth. To do so, a code was developed to extract only the files in the validation set from the Structure Table Words files. This preprocessing pipeline was designed to merge table structure information from XML files, specifically the bounding box information, including table columns, table rows, table column header, table spanning cell, and table projected row, with textual content from JSON files to create a combined ground truth for the PDF images. The aim is to generate a structured JSON output that contains relevant table features from the PDF image, making the annotated data suitable for further analysis and model training. For each cell, there is the text content, the bounding box of the cell, and the information about the cell type.

The preprocessing consists of multiple steps: extracting the annotated table structure, assigning text, and computing features. The Figure 3.1 shows the pipeline steps:

(1) - The first step involves parsing XML files that contain table annotations. These XML files store bounding box information for key table elements, such as columns, rows, headers, spanning cells, and projected row header cells. The code extracts these elements by reading their coordinates and classifying them based on predefined categories. This structured extraction facilitates the identification of individual table components.

(2) - Once the annotated table structure has been extracted, the script determines individual table cells by computing intersections between detected rows and columns. Each intersection represents a potential table cell.

(3) - If the XML file contains bounding boxes identified as spanning cells or a

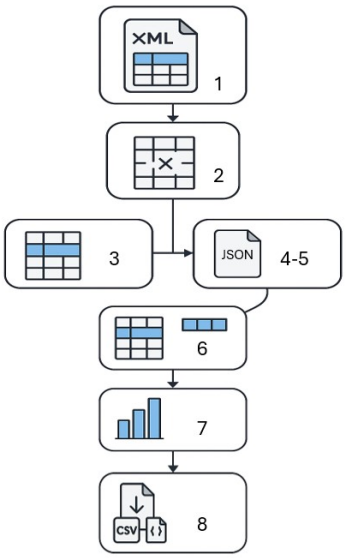

Figure 3.1: Workflow pipeline for ground truth. (1) Load and parse XML; (2) Calculate cells; (3) Process spanning cells and projected row headers; (4) Load words JSON; (5) Round bounding boxes; (6) Match text to cells; (7) Compute features; (8) Save processed data.

projected row header cell, the code uses their bounding box since they represent multiple cells inside just one cell, instead of using the cells generated by the intersections. Otherwise, it uses the calculated cells. This step helps refine the table structure by managing complex layouts.

(4) - At the same time, the script processes textual information extracted from document images stored in JSON files. These JSON files contain words and their corresponding bounding boxes for all words within the image, including those that are outside the table. They are loaded to be processed.

(5) - The bounding box coordinates are rounded to the same decimal number as the cells, and the files are now prepared for integration with the table structure.

(6) - After processing both XML and JSON data, the script merges them by assigning words to table cells. This is performed by checking if a word's bounding box falls within the boundaries of a detected cell. A small tolerance is applied to include words near the edges of a cell, ensuring that all relevant text is captured. The final result is a structured dataset containing each cell's assigned text and characteristics, i.e the ground truth of the PDF images.

(7) - Once the table structure and textual content are combined, the script extracts key features to characterise each table file. These features include the number of columns, rows, and the total number of detected cells. Additionally, the script determines whether the table contains column headers and computes complexity indicators such as spanning cells and projected row headers. Another important feature created is the character density, which represents the average number of characters per cell and

provides insights into text distribution within the table.

(8) - The processed data is saved as a structured JSON file that contains the bounding box position of the cell, the text and the type of cell (which can be a normal cell, spanning cell, column header cell, projected row header cell, as shown in Figure 3.2).

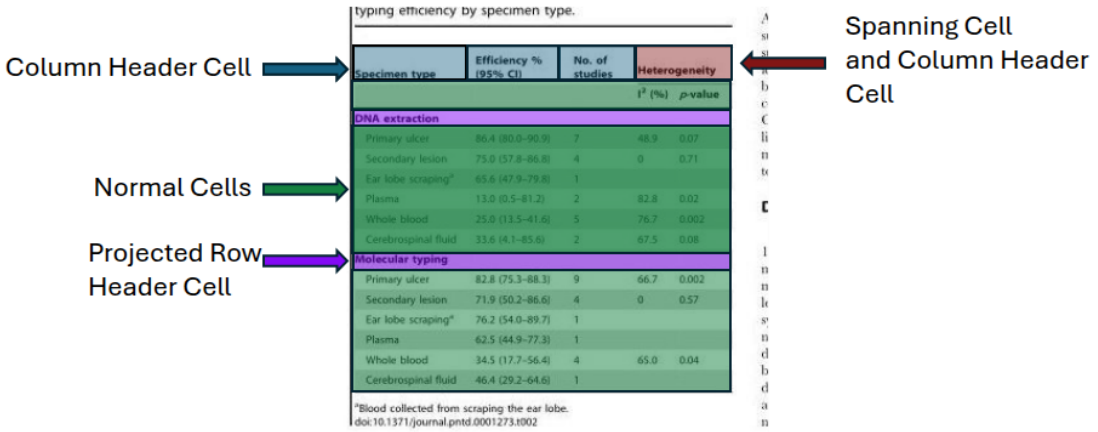

Figure 3.2: The different types of cells.

Ultimately, each JSON file created from the original files is converted to a CSV file, following a logic based on cell position and cell type. Because CSV files require a closed grid or matrix, the total number of cells must be the product of the number of rows and the number of columns. Using this information, a CSV file is created and populated with the total number of cells, along with the text from each cell, extracted from the JSON within the cells. For the complex tables that have spanning cells and projected rows, a different logic was applied to replicate them as explained in Table 3.2.

Table 3.2: Logic to transform the JSON into a CSV based on the type of cell.

Type of Cell	Logic
Normal Cell	Use the bounding box positions to recreate each cell into csv
Spanning Cell	Replicate the text for all the cells that are occupied by the spanning cell bounding box
Projected Row Header	Write the text in the leftmost cell and create empty cells for the rest of the row

This preprocessing pipeline is crucial in structuring raw table annotations and text data into a well-organised format. Merging table structure with textual content and computing meaningful features facilitates the automatic analysis of tables in scanned documents and other structured data sources. With that, we have the original images and the ground truth of the annotated cells.

3.2 Simple and Complex Sampling

To enable fair and insightful evaluation, the dataset was divided into simple and complex table subsets based on structural characteristics. This section outlines the

sampling strategy applied, the criteria for defining table complexity, and the validation steps taken to ensure the sampled data is representative of the full dataset.

A new dataset was created with the final JSON files, where each row is related to a JSON-extracted cell and content information from an image; some features were created using this information and are explained below, respectively:

- **filename**: Name of the file containing the table data.
- **num_columns**: Number of columns in the table.
- **num_rows**: Number of rows in the table.
- **num_cells**: Total number of cells.
- **has_column_headers**: Boolean value indicating if the table has column headers.
- **is_complex_table**: Indicates if the table has a complex structure (e.g., spanning cell or projected row).
- **num_spanning_cells**: Number of cells that span across multiple rows or columns.
- **num_projected_rows**: Number of projected row header cells.
- **character_density**: Ratio of text characters to total number of cells.
- **cells**: Structured data representing individual cells within the table.

The dataset was split into simple and complex categories; the complex category is considered when a table has a spanning cell or a projected row header. The simple table dataset contains 44342 values, and the complex dataset contains 49474 values. Due to budget, time, and computational constraints, a random sample of 1000 examples was chosen for each dataset. A comparison of the means of each feature was done to show that the 1000 samples have values close to the total dataset, as shown in Table 3.3 where the difference between the complete set and the subsample is less than one.

Table 3.3: Comparison of the mean value between the datasets based on their complexity (simple and complex) and their subsample.

Complexity	Dataset	Features (mean values)								
		Total	num_columns	num_rows	num_cells	has_column_headers	is_complex_table	num_spanning_cells	num_projected_rows	character_density
Simple	Complete	44342	4.79	10.34	51.73	0.87	0	0	0	14.76
	Subsample	1000	4.70	10.48	51.90	0.87	0	0	0	16.06
Complex	Complete	49474	6.09	16.09	83.04	0.97	1	4.13	1.34	10.75
	Subsample	1000	6.11	16.21	83.05	0.97	1	4.11	1.48	10.49

3.3 OCR for Table Extraction

The first table extraction approach evaluated in this thesis leverages a traditional neural network model combined with OCR technology. This section describes the TATR model, its integration with OCR tools, and the steps taken to generate structured table outputs from document images.

3.3.1 TATR-OCR

The first approach utilises the TATR, which is based on DETR. DETR comprises a convolutional backbone followed by an encoder-decoder transformer, offering an end-to-end training approach for object detection. The specific model employed was TATR-v1.1-All, available at: https://huggingface.co/bsmock/TATR-v1.1-All. This model was trained using the PubTables-1M and FinTabNet.c datasets, with details of the training procedure described in (Smock et al., 2023a). The pipeline is designed to process cropped tables from PDFs as images (PNG format) and extract meaningful structural information. The process begins by loading a list of image file names, followed by preparing the necessary environment, which includes importing libraries and loading the pretrained table recognition model. The model is configured to detect and classify table elements, including rows, columns, column headers, spanning cells, and projected row headers. When the input consists of PDF files, each page is first converted into an image for further processing.

The target image is resized to a consistent maximum dimension, normalised, and converted into a format compatible with the model. Once preprocessed, the image is passed through the model, which detects table elements and outputs bounding box coordinates, labels, and confidence scores. The detections are filtered to retain only those with confidence scores higher than 0.95 to ensure precision. These predictions are saved as XML files, where each table element is recorded with its label and coordinates.

In the post-processing stage, the XML files are parsed to extract structural components and compute a grid of cells by intersecting row and column bounding boxes. The grid is refined to incorporate spanning cells and projected row headers, and cells are annotated to identify column headers where applicable. The output is converted into JSON format for subsequent processing.

Cell-level OCR is then performed. Each cell is cropped from the original image with slight padding and enhanced through grayscale conversion, contrast adjustment, and resizing. PaddleOCR is applied to extract text from each individual cell, and the recognised text is integrated into the JSON file. The cells are then organised into rows and columns based on their coordinates, and the data is exported as CSV files.

Finally, the resulting CSV files can be compared with ground truth data using the final JSON files, enabling quantitative evaluation of structure recognition and text extraction performance. To facilitate qualitative analysis, a separate code module was developed to overlay bounding boxes on the original images, highlighting the components detected by the model. This visualisation serves as a crucial tool for interpreting the model's predictions and for manual inspection of its performance.

Figure 3.3 illustrates the model output on a sample table. The first image shows the combined grid structure generated by intersecting rows and columns. The second image displays the detected rows, while the third highlights the columns. The fourth image shows the bounding boxes corresponding to column headers. Although spanning

cells and projected row headers were not present in this specific example, the model is capable of detecting them and will annotate them accordingly when present.

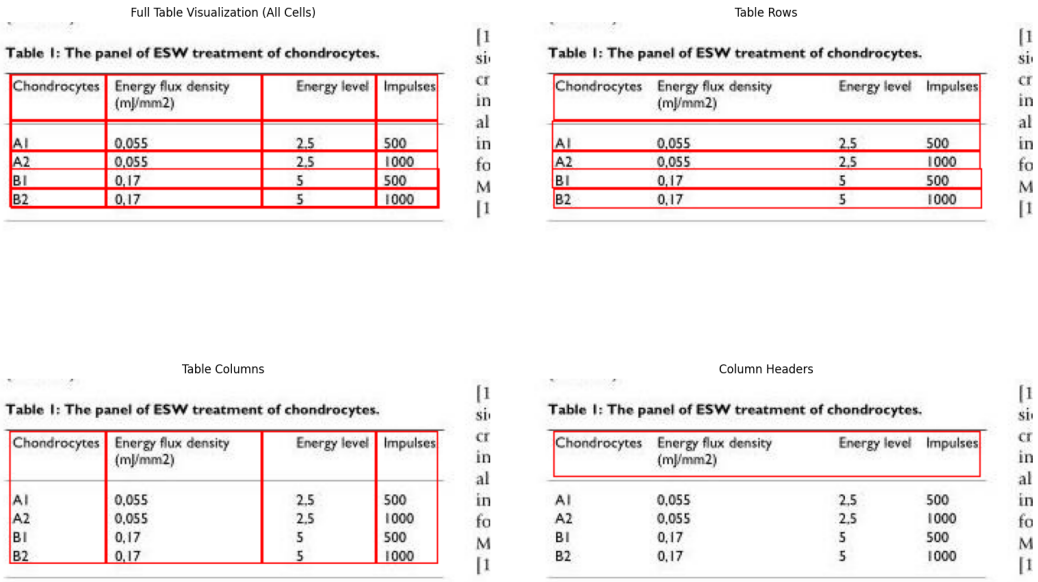

Figure 3.3: Output of the model TATR showing various table components detected.

3.4 Multimodal LLMs for Table Extraction

In contrast to the traditional OCR approach, this section presents the application of multimodal LLMs with vision capabilities for table extraction. The models evaluated, their configurations, and the unified processing pipeline are described, with an emphasis on the use of consistent prompts and output validation to ensure comparability.

3.4.1 Models

This thesis evaluates four large multimodal language models with visual reasoning capabilities: Granite Vision, Phi-3 Vision, GPT-4o, and its lightweight variant GPT-4o Mini. These models were selected due to their strong performance in structured document processing tasks, including table recognition, visual understanding, and multimodal question answering. The purpose of this thesis differs, as it involves extracting structured data from an unstructured data source.

Granite Vision, (GraniteVision, 2025), is an open-source visual document understanding model developed by IBM Research. It has been trained on a multimodal

dataset of approximately 13 million images and 80 million visual-text instructions. The model is optimised to extract structured information from medium-resolution inputs, such as forms, tables, plots, and infographics, and incorporates dedicated architectural components for spatial layout interpretation.

Phi-3 Vision, (Microsoft, 2024), released by Microsoft, combines a ViT-L/14 vision encoder with a transformer-based text decoder in a compact phi-3.5-mini configuration. The model has been instruction-tuned on approximately 500 billion tokens and is optimised for low-resolution visual inputs. While primarily designed for English-language scenarios, Phi-3 Vision has shown strong performance on document and layout understanding tasks.

GPT-4o, (OpenAI et al., 2024), also known as GPT-4 Omni, is a state-of-the-art multimodal model introduced by OpenAI in 2024. It integrates text, vision, and audio capabilities in a unified architecture and is pre-trained on a combination of publicly available and proprietary datasets. GPT-4o supports high-resolution inputs and demonstrates strong reasoning abilities across visual and tabular domains. Its outputs are notable for precision in structural alignment and content fidelity.

GPT-4o Mini (OpenAI et al., 2024) is a smaller variant of GPT-4o, designed to operate in more constrained computational environments. While it shares architectural similarities with its larger counterpart, it emphasises efficiency and inference speed, making it a practical choice for cost-sensitive applications. Despite its reduced size, GPT-4o Mini retains competitive performance in layout-sensitive tasks such as table extraction.

Together, these four models offer complementary strengths across accuracy, resolution handling, and computational efficiency, enabling a comparative analysis of their suitability for table recognition in complex real-world documents.

3.4.2 Pipeline

All models are integrated into a unified codebase in which only the API configuration and model selection vary, it was used OpenAI API for the GPT 4 models, Fireworks API for Phi-3-Vision and Ollama API for Granite. This approach ensures that all models are evaluated under identical conditions, using the same input data and processing logic.

The implementation is designed to be modular and reusable. A configuration variable ("MODEL CONFIG") dynamically selects the model for each execution, while the remainder of the pipeline—from image preprocessing to data export—remains consistent regardless of the chosen model. This design guarantees fairness in the comparison between models and simplifies the experimental setup.

Structured output, example in Figure 3.4(a), is adopted to ensure consistency in the responses and to facilitate quantitative evaluation. Each model is prompted to produce output in the form of a standardised JSON object containing two keys, called a schema

as shown in Figure 3.4(b): headers, representing a list of column names, and rows, representing the table's data as a list of lists of strings. The system prompt provided in Figure 3.4(c) to the models specifies detailed instructions on how to format the output, and the requirement that all rows contain the same number of columns. Empty cells must be represented explicitly as empty strings (""). The prompt also emphasises that no additional formatting, commentary, or metadata should appear in the JSON output.

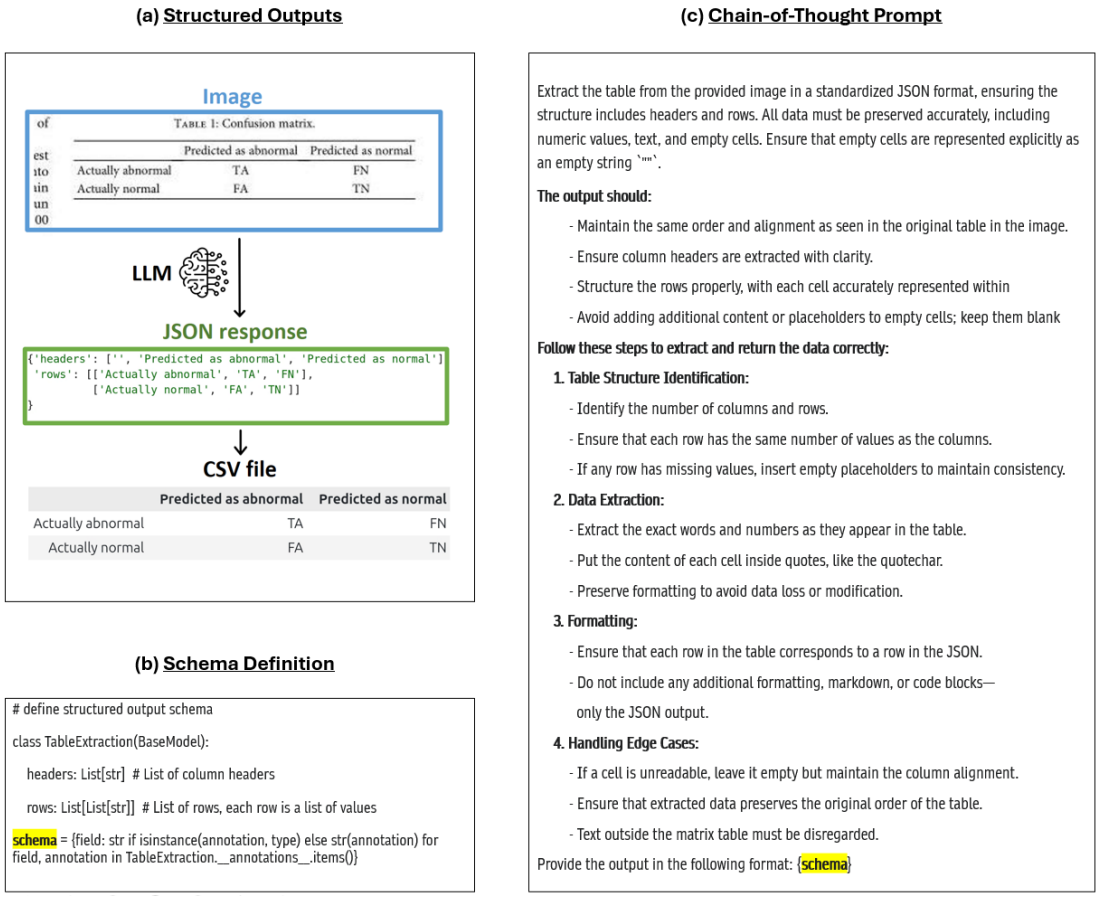

Figure 3.4: (a) Example of table extraction with LLM structured output. The LLM converts the input image into a structured JSON response, extracting the table attributes - headers and rows - whilst ignoring content outside the table. This JSON is then converted into a CSV file for evaluation. (b) Structured output schema definition. (c) The chain-of-thought prompt is used with the structured output technique.

This prompt was used for the simple tables dataset, but when trying to use it for the complex dataset, it failed to extract. An improved system prompt was developed to handle special cases such as horizontal and vertical spanning cells, projected row headers with some direct example in it, as shown in Figure 3.5.

The expected output structure is defined and validated using the Pydantic library. A class named TableExtraction is defined as a subclass of BaseModel, specifying that the headers field contains a list of strings and the rows field contains a list of lists of strings. This schema enables the automatic validation of LLM responses, ensuring

```
SYSTEM_PROMPT = f"""Extract the table from the provided image and output a standardized JSON that represents the CSV structure. The JSON must have exactly two keys:{schema}

All data must be preserved exactly as it appears in the image, including numeric values, text, and empty cells (which must be represented explicitly as an empty string "").
Each row in the final output must have the same number of cells as there are header columns.
Follow these instructions exactly:

1. Table Structure:
   - The table may contain multiple header rows. In our examples, there are two header rows.
   The final output must have a uniform number of columns (for instance, 12 columns as in our ground truth).
   - Every data row must have the same number of cells as the header row.

2. Spanning Cells:
   - **Horizontal Spanning (across columns):**
     If a header or data cell spans more than one column, replicate its value into each affected column.
     Example:
     Visual header:
     ┌─────┬─────────────────────┐
     │  X  │          Y          │  <-- The header cell "Y" spans two columns
     └─────┴─────────────────────┘
     Expected JSON header:
      "headers": ["X", "Y", "Y"]

   - **Vertical Spanning (across rows):**
     If a cell spans multiple rows, then subsequent rows may show an empty cell where the spanning cell applies.
     In that case, backfill the empty cell with the value from the previous row.
     Example:
     Visual table data:
     ┌──────────┬────────┐
     │ Group A  │ Value1 │
     ├──────────┼────────┤
     │          │ Value2 │   <-- The first cell is empty because "Group A" spans vertically.
     └──────────┴────────┘
     Expected JSON rows:
     [
       ["Group A", "Value1"],
       ["Group A", "Value2"]
     ]

3. Projected Rows:
   - In some cases the table is designed so that only the leftmost cell of a row carries text and all subsequent cells in that row are meant to be empty.
     Example:
     Visual projected row:
     ┌──────────────────┬─────┬─────┐
     │ Department: Sales │     │     │  <-- Only the first cell contains text.
     └──────────────────┴─────┴─────┘
     Expected JSON row:
     ["Department: Sales", "", ""]

4. Uniformity of Columns:
   - Every header and data row must have the same fixed number of columns.
   If the ground truth (or target CSV) has 12 columns, then make sure that if a row is missing any element due to spanning,
   you replicate or pad with empty strings so that the final row length is 12.

5. Output Requirements:
   - Do not include any extra formatting, markdown, or commentary in the output—only valid JSON.
   - The final JSON must strictly adhere to the following structure:{schema}
```"""

Figure 3.5: Improved system prompt to handle spanning cells, projected row headers with some examples.

compliance with the expected format and facilitating the early detection of errors or inconsistencies.

The processing pipeline consists of several stages. A predefined list of image file names is loaded from a text file, ensuring control over the dataset used for evaluation. Each image is loaded, resized to a maximum dimension of 1000 pixels while preserving its aspect ratio, and converted to JPEG format before being encoded as a base64 string. The encoded image and the system prompt are submitted to the selected model via the corresponding API. The response is parsed and validated against the Pydantic schema. Valid responses are converted into pandas DataFrames and saved as CSV files for subsequent analysis.

This methodology enables consistent, controlled, and reproducible evaluation of table extraction performance across different LLMs, with strict output specification and automated validation of results.

3.5 Evaluation Metrics

To assess the performance of the different table extraction approaches, a comprehensive set of evaluation metrics was defined. This section introduces these metrics,

which include failure rates, structural error analysis, row and column accuracies, table shape accuracy, F1 score variations, and grid similarity scores. These metrics offer both quantitative and qualitative insights into the models' abilities to reconstruct table structures and content accurately.

3.5.1 Failure Rate

The first metric used was the model's failure rate. The failure rate is defined as the percentage of images for which the model output does not represent a valid CSV. This means that the predicted output cannot be successfully parsed into a valid two-dimensional grid structure. This failure might occur due to output format inconsistencies, parsing errors, or other issues that prevent the system from generating a coherent table structure.

3.5.2 Structural Errors

To better understand the errors that the models may encounter when extracting a valid grid, four types of structural errors were considered: missing rows, extra rows, missing columns, and extra columns.

Missing rows occur when the model fails to detect rows that are present in the ground truth table, indicating under-segmentation. Extra rows arise when the model predicts additional rows that do not exist in the ground truth, representing over-segmentation. Missing columns reflect situations where the model does not capture actual columns. Extra columns occur when the model introduces spurious columns that are not present in the true structure.

These errors are calculated by comparing the predicted table dimensions to the ground truth:

- *Missing Rows* = $\max(0, \text{Ground Truth Rows} - \text{Predicted Rows})$
- *Extra Rows* = $\max(0, \text{Predicted Rows} - \text{Ground Truth Rows})$
- *Missing Columns* = $\max(0, \text{Ground Truth Columns} - \text{Predicted Columns})$
- *Extra Columns* = $\max(0, \text{Predicted Columns} - \text{Ground Truth Columns})$

The total and mean values of these errors are reported, along with their percentage relative to the total number of rows and columns in the ground truth. This allows for a quantitative analysis of the model's tendency towards over- or under-segmentation.

3.5.3 Accuracies

Row accuracy evaluates the predicted versus actual row counts, penalising both extra and missing rows. It normalises errors relative to the larger count, helping identify structural extraction issues that can impact readability.

Column accuracy assesses whether the predicted number of columns aligns with the actual count. Misidentified or skipped columns reduce accuracy. High column accuracy is crucial for maintaining data alignment and structural integrity in table extraction.

Table shape accuracy measures how well predicted table dimensions match actual ones, calculated as the harmonic mean of row and column accuracy. Significant errors in either dimension greatly affect overall accuracy.

$$\text{Shape Accuracy} = \frac{2}{\frac{1}{\text{Row Accuracy}} + \frac{1}{\text{Column Accuracy}}} \tag{3.1}$$

3.5.4 F1-Score Variations

One of the key metrics used in this analysis is the F1 Score, which balances Precision and Recall to assess the correctness of the extracted tables. The F1 Score is a harmonic mean of Precision and Recall, ensuring that a model's success in extracting table data considers both accuracy and completeness.

Precision: assesses how many extracted cells are correct, compared to the total number of extracted cells.

Recall: determines how many ground truth cells were correctly matched, ensuring a high retrieval rate.

F1 Score combines both metrics to provide an overall measure of the model's ability to extract tables correctly:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \text{where } P \text{ is Precision and } R \text{ is Recall.} \tag{3.2}$$

Different thresholds were applied to measure the similarity between predicted and ground truth cell content to assess text-based table extraction accurately.

Threshold = 0 $\rightarrow$ Measures structural layout accuracy only, ignoring text content.

Thresholds = 60, 80, 100 $\rightarrow$ Measure content similarity using fuzzy matching, requiring increasing levels of textual accuracy.

For instance, a threshold of 60 allows major text variations (typos), 80 allows minor text variations (typos), while a threshold of 100 requires an exact match between predicted and ground truth content.

3.5.5 GriTSCon

GriTSCon, created by (Smock et al., 2023b), is a robust metric for assessing the similarity between two grid-like tables of text, they also created for topology and location but this thesis focused on the cell content. It works by first computing a similarity score for every possible pair of cells from the two grids, using a function based on the LCS technique. This LCS-based score, which ranges from 0 to 1, determines how much textual content the respective cells have in common—even recognizing partial

matches. Instead of simply comparing each pair of cells in fixed positions, the method searches for the best matching substructures by considering various combinations of rows and columns.

To better understand the LCS, consider two strings: "AGCAT" and "GAC". Using a dynamic programming approach, their LCS is computed. The following table visualises the computation process, where the columns correspond to the characters of "AGCAT" (with an initial empty column) and the rows correspond to the characters of "GAC" (with an initial empty row). Each cell shows the length of the LCS for the corresponding prefixes.

		A	G	C	A	T
	0	0	0	0	0	0
G	0	0	1	1	1	1
A	0	1	1	1	2	2
C	0	1	1	2	2	2

The value in the bottom-right cell is 2, indicating that the LCS of "AGCAT" and "GAC" has a length of 2. By backtracking the table, it is found that one valid LCS is "AC". Using the LCS similarity formula,

$$\text{similarity} = \frac{2 \times \text{LCS length}}{\text{length of string1} + \text{length of string2}}$$

, the similarity is calculated as

$$\text{similarity} = \frac{2 \times 2}{5 + 3} = \frac{4}{8} = 0.5.$$

This means that the two strings are 50% similar according to the LCS-based metric. Once all cell-level similarity scores are computed, the algorithm independently aligns the rows and columns of both grids to find the overall optimal matching configuration. The aligned scores are then aggregated and normalised to produce a final evaluation expressed in terms of an F-score, along with precision and recall. This approach not only handles local mismatches but also captures the broader structural similarities between tables, making GriTSCon especially effective for applications in table recognition and data extraction.

4

Results

This chapter presents a detailed analysis of table extraction performance across five models —TATR-OCR, GPT-4o, GPT-4o-mini, Granite, and PHI-3-Vision—on two datasets comprised of simple and complex table images, each containing 1000 images. The results are structured to compare failure rates, structural accuracy, and error types, distinguishing between simple and complex tables. Additionally, the Structural F1 Score, which evaluates cell placement accuracy without considering text content, and the Cell-Content F1 Score, which measures text correctness at multiple similarity thresholds (60%, 80%, and 100%). The GriTSCon metric is also applied to assess both structure and text together, providing a comprehensive evaluation of the quality of table extraction. Each metric is examined to determine how well models preserve grid integrity, handle spanning cells, and extract textual content. The findings highlight differences between traditional OCR-based methods and multimodal LLMs, providing insights into their strengths and limitations in table extraction tasks.

4.1 Failure Rate

The failure rate provides an initial measure of each model's reliability in generating valid table grids. It is defined as the percentage of cases where the model fails to produce a structured output that can be parsed into a valid two-dimensional table. Failures typically result from output format inconsistencies, parsing errors, or incomplete predictions that prevent the grid from being reconstructed.

Figure 4.1 summarises the failure rates for simple and complex tables. TATR-OCR achieved a 0% failure rate across both table types, demonstrating high robustness and consistency in generating valid outputs. GPT-4o exhibited a low failure rate of 2.5% on simple tables, which increased to 13.9% for complex tables. This indicates that while GPT-4o reliably handles simpler layouts, its performance degrades when spanning cells or projected headers are present.

Granite and GPT-4o-mini exhibited greater sensitivity to structural complexity, with

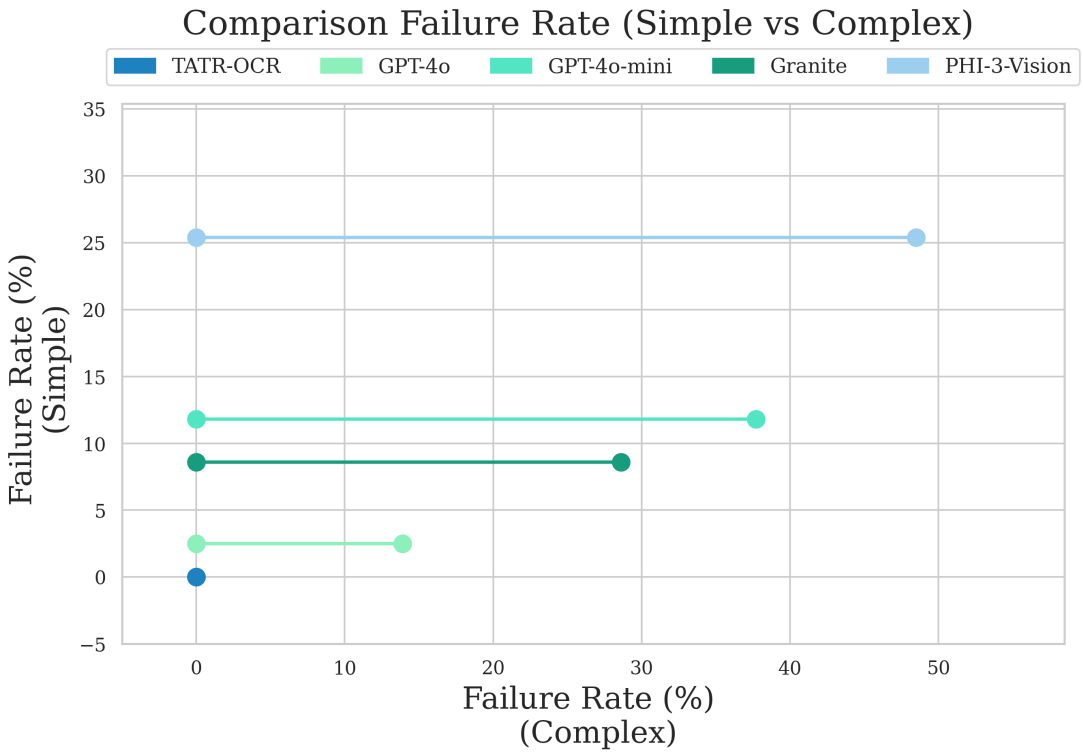

Figure 4.1: Model failure rates, difference between simple and complex table.

failure rates rising significantly for complex tables 8.6 to 28.6% and 11.8 to 37.7% respectively. PHI-3 Vision was the least reliable model, with 25.4% failure on simple tables and 48.5% on complex tables, highlighting challenges in generating valid structures for more intricate layouts. These failures are better represented in the next subsection, where the size of the original image and the inference time are plotted, and the failures are marked as red dots.

4.2 Inference Time and Variability

Table 4.1 presents a comparative analysis of the time performance for generating final CSV files for simple and complex table subsets. TATR-OCR demonstrates the most consistent and efficient performance, with a minimal time range (0.4–0.6s to 77–49.5s) and a low mean time (5.2–6.4s), resulting in a total processing duration between 1 hour and 27 minutes to 1 hour and 47 minutes. PHI-3-Vision exhibits similar stability, with only slightly higher mean and maximum processing times, indicating good scalability and efficiency.

GPT-4o and GPT-4o-mini exhibit greater variability, particularly in maximum processing times (e.g., GPT-4o reaching up to 516.1 seconds and GPT-4o-mini up to 454.6 seconds), with higher standard deviations (up to 23 seconds and 18.8 seconds, respectively). This suggests intermittent delays that may be attributed to model response time or server-side latency, despite having relatively moderate mean processing times.

Table 4.1: Comparison of time of inference for each model and total time for the 1000 tables sample.

Time between creation of final CSV files (1000 Simple Tables -1000 Complex Tables)					
Model	Min(s)	Max(s)	Mean(s)	Standard Deviation(s)	Total Time(min)
TATR-OCR	0.4 - 0.6	77 - 49.5	5.2 - 6.4	5.29 - 4.9	1h 27min - 1h 47min
GPT-4o	1.7 - 1.7	516.1 - 89.1	10.9 - 13.6	23 - 9.6	3h 01min - 3h 47min
GPT-4o-mini	1 - 1.4	378.6 - 454.6	6.5 - 10.1	15.3 - 18.8	1h 49min - 2h 48min
Granite Vision	1.6 - 1.3	1801.7 - 3605.8	23.3 - 66.9	170.1 - 340.4	6h 28min - 18h 31min
PHI-3-Vision	1.1 - 1.2	24 - 33.6	5.4 - 5.9	4.5 - 7.4	1h 30min- 2h 4min

Granite Vision, in contrast, displays the most significant variation and the highest processing times across all metrics. Its total processing time ranges from over 6 to more than 18 hours, with an exceptionally high standard deviation (170.1–340.4s), indicating inconsistency and potential inefficiency when handling larger or more complex input data. This highlights a key trade-off between model sophistication and runtime performance.

Overall, while larger vision-language models such as Granite Vision may offer richer representations, their practicality for large-scale table extraction workflows is limited by prohibitive processing times. Conversely, lightweight models like TATR-OCR and PHI-3-Vision strike a better balance between performance and speed.

Figure 4.2 provides a visual overview of the relationship between image size and inference time for each model, separately plotted for simple and complex tables. All subplots are presented on a log-log scale to capture the growth behaviour of inference time better relative to image size. In this context, larger image sizes often correspond to more content-rich tables, such as those with a greater number of rows, columns, or text density—factors that typically increase the complexity of the extraction task.

TATR-OCR (Figure 4.2a) exhibits a clear and consistent scaling behaviour, with inference time increasing smoothly with image size and a negligible number of failures across both table types. GPT-4o (Figure 4.2b) follows a similar trend, although some failed cases appear, particularly for complex tables with larger sizes. GPT-4o-mini (Figure 4.2c) displays a similar scaling pattern but with a noticeably higher failure rate, especially in the mid-to-high image size range.

Granite Vision (Figure 4.2d) shows the highest variability in inference time, including several outliers exceeding 1000 seconds. Failures occur across a wide range of image sizes, and the relationship between size and inference time is less consistent. PHI-3-Vision (Figure 4.2e) demonstrates a relatively coherent scaling behaviour, but with a high number of failures on complex tables, especially as image size increases.

Red markers in each subplot represent failed predictions—cases where the output could not be parsed into a valid table. While these failures are more frequent with larger and more complex inputs, some also occur at smaller image sizes, suggesting that

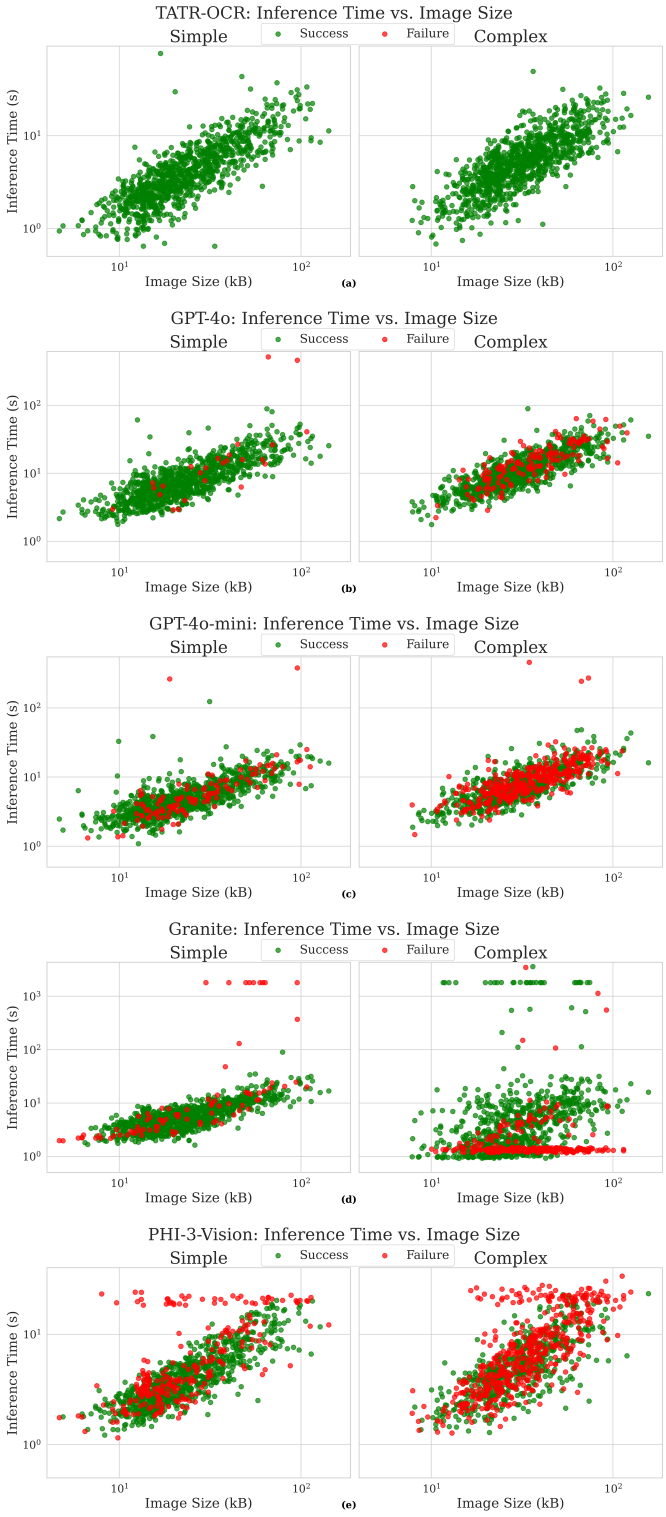

Figure 4.2: Time vs. Image Size for each model on simple and complex tables.

structural characteristics (e.g., spanning cells or irregular layouts) may be more critical factors than size alone. This visualisation reinforces the trends observed in the failure rate statistics and execution time measurements, offering additional insight into the inference behaviour and robustness of each model under varying table complexities.

These results underscore the advantage of models explicitly designed for structure detection, such as TATR-OCR, and highlight the difficulty LLM-based models face in generating reliable grid outputs for complex tables.

4.3 Structural Errors

To understand where and how our models falter in recognising table layouts, we examine two complementary error modes: missing and extra row/column detections. Missing elements reveal under-segmentation—cells, rows or columns the model failed to identify—while extra elements indicate over-segmentation, where the model "hallucinates" grid lines or cells that do not exist. Together, these errors expose the balance each model strikes between conservative and liberal structure inference.

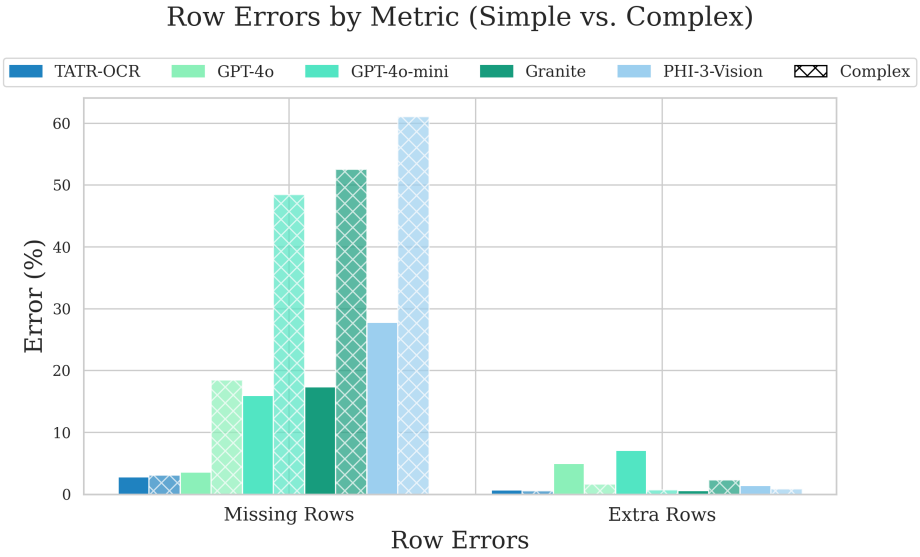

Figure 4.3: Comparison between the error in rows extraction: the left side shows the percentage of missing rows, and the right side shows the percentage of extra rows. Hatched bars represent complex tables.

Figure 4.3 contrasts missing- and extra-row rates on simple versus complex tables. On simple layouts, TATR-OCR misses only 2.83% of rows, closely followed by GPT-4o at 3.62%. GPT-4o-mini and Granite already lag behind at 16.01% and 17.39%, respectively, while Phi-3-Vision jumps further to 27.81%. When complex tables are compared, only TATR-OCR remains stable, rising marginally to 3.09%. By contrast, GPT-4o's missing-row rate inflates nearly fivefold to 18.48%, and GPT-4o-mini surges to 48.47%. Granite and Phi-3-Vision perform worst, missing 52.55% and 61.06% of rows. These

jumps underscore that, without specialised training on multi-row or projected-header structures, LLM-based extractors severely under-segment in the face of layout intricacy.

Extra-row errors (Figure 4.3, right) paint a more nuanced picture. Some models—Granite and PHI-3-Vision—keep their over-detection rates below 3% regardless of table type, demonstrating a conservative bias that avoids spurious rows. GPT-4o-mini, by contrast, reduces its extra-row rate from 7.1% in simple tables to under 1% in complex ones, hinting at unstable segmentation heuristics when confronted with merged regions. TATR-OCR and GPT-4o also maintain extra-row rates under 5%, indicating a good balance between recall and precision.

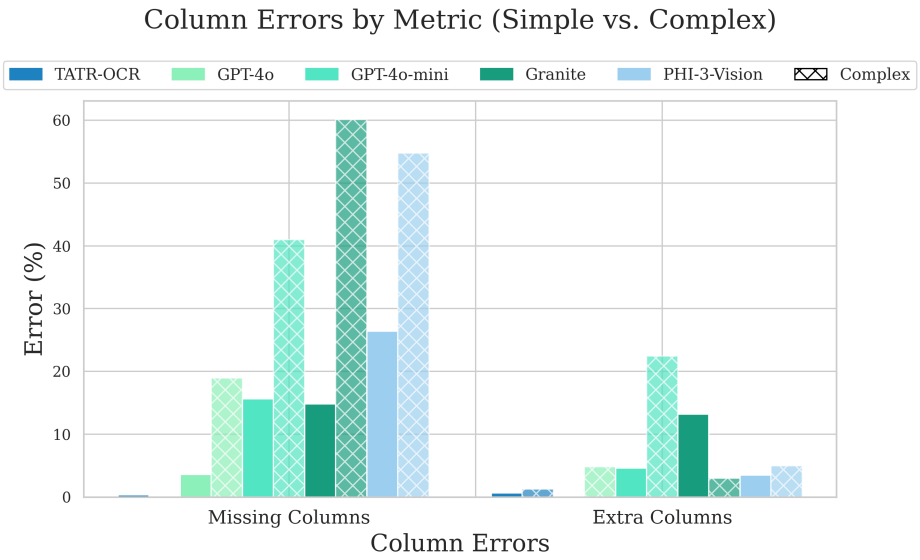

Figure 4.4: Comparison between the error in column extraction: the left side shows the percentage of missing columns, and the right side shows the percentage of extra columns. Hatched bars represent complex tables.

A similar pattern emerges for column errors (Figure 4.4. All models start with low missing-column rates on simple tables—TATR-OCR at 0.8%, GPT-4o at 3.6%—but only TATR-OCR remains under 1% for complex layouts. GPT-4o's missing-column error climbs to 18.9%, and Granite's soars to 60.1%, mirroring its poor row performance. Extra-column errors generally remain lower than 15% for all models, yet GPT-4o-mini and PHI-3-Vision show erratic shifts when complexity increases, again suggesting that LLMs without specialised structural training tend to misplace columns in the face of nested or spanned cells.

Collectively, these error profiles illuminate each model's extraction style. TATR-OCR exhibits minimal deviation between simple and complex tables, underlining its robust handling of grid topologies. GPT-4o strikes a middle ground—its row and column recall degrade in complex scenarios, but it avoids rampant over-segmentation. By contrast, GPT-4o-mini and Granite, while serviceable on straightforward tables, consistently underperform on cells with span and projection. PHI-3-Vision, with

the highest missing-cell rates, appears to be ill-suited for tasks that demand precise boundary recognition.

4.4 Structural Accuracy Comparison

The structural accuracy of the models was evaluated through three key metrics: row accuracy, column accuracy, and table shape accuracy (the harmonic mean of row and column accuracy). The results, shown in Figure 4.5, offer insight into how well each model reconstructs the overall layout of tables.

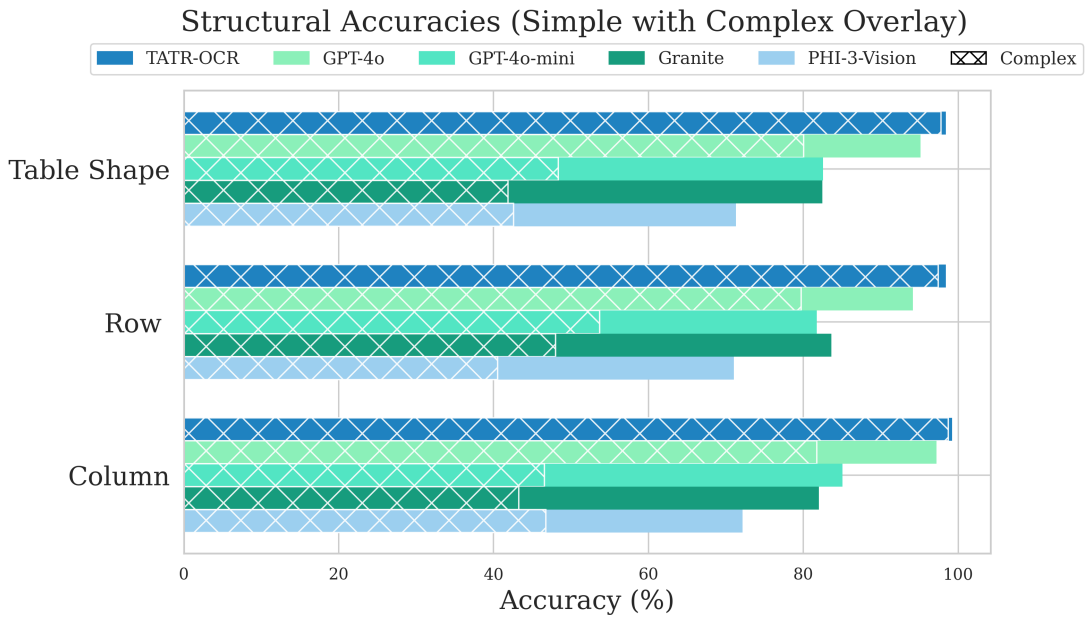

Figure 4.5: Table shape accuracy, row accuracy, columns accuracy for simple and complex (hatched) tables.

On simple tables, TATR-OCR achieved near-perfect performance, with table shape accuracy of 98.4%, row accuracy of 98.4%, and column accuracy of 99.2%. These results confirm the model's ability to consistently reproduce basic table grids with minimal error. GPT-4o followed with strong performance (shape accuracy 95.6%, row accuracy 94.2%, column accuracy 97.2%), although a modest gap remained compared to TATR-OCR.oth GPT-4o-mini and Granite performed moderately well on simple tables, with table shape accuracies between 82% to 85%. PHI-3-Vision lagged behind, achieving shape accuracy of approximately 71%.

As table complexity increased, performance disparities became more pronounced. TATR-OCR exhibited remarkable robustness, with table shape accuracy decreasing only marginally to 97.8% and both row and column accuracies above 97%. This stability demonstrates the model's resilience in handling more challenging table configurations, including those with spanning cells and projected row headers. In contrast, GPT-4o experienced a noticeable drop in structural accuracy, with table shape accuracy falling

to 80.6%, driven by declines in both row and column precision. GPT-4o-mini and Granite were significantly affected by complexity, with shape accuracy dropping to 46.6% and 41.9% respectively, reflecting their limited capacity to cope with multi-row, merged, or irregular cells. PHI-3-Vision showed the weakest performance overall, with table shape accuracy dropping to 40.5% on complex tables.

These results suggest that while LLMs can deliver acceptable structural accuracy on simple table layouts, their performance deteriorates substantially when applied to complex table structures. GPT -4O, in particular, demonstrates moderate resilience, with a decline of approximately 15 percentage points in table shape accuracy when transitioning from simple to complex scenarios. GPT-4o-mini and Granite, on the other hand, exhibit high sensitivity to complexity, suffering substantial performance losses of 30 to 40 percentage points. PHI-3-Vision's consistently low accuracy indicates that it is not well-suited for structural table extraction tasks without additional adaptation. In contrast, TATR-OCR consistently outperforms the other models, offering reliable structural reconstruction regardless of table complexity and confirming the advantages of OCR-based approaches for table extraction.

4.5 Structural-Layout F1 Score

Figure 4.6 compares each model's ability to recover cell boundaries and overall table geometry, using the Structural-Layout F1 metric on both simple (solid bars) and complex (hatched bars) tables.

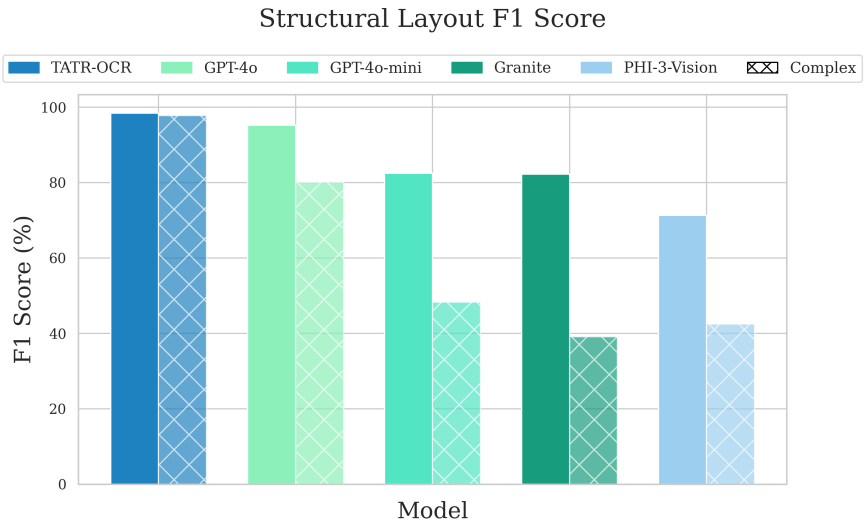

Figure 4.6: Structural-Layout F1 scores on simple and complex (hatched) tables.

On the simple tables dataset, TATR-OCR achieves the highest F1 score of 98.4%, followed by GPT-4o at 95.2%. Granite and GPT-4o-mini perform similarly in the low-80% range (82.2% and 82.4%, respectively), while PHI-3-Vision trails at 71.3%. Introducing multi-row headers, spanning cells, and projected rows affects each model's performance

to varying degrees. TATR-OCR remains remarkably stable, declining by just 0.6 points to 97.8%. GPT-4o falls more sharply, losing 15.2 points to 80.0%. The compact models suffer the most significant drops: GPT-4o-mini drops to 48.3% (–34.1 points), Granite to 39.1% (–43.1 points), and PHI-3-Vision to 42.5% (–28.8 points).

These results reveal three key lessons. First, TATR-OCR's near-flat performance curve confirms its robustness in detecting cell boundaries even under complex layouts. Second, GPT-4o delivers strong simple table performance but shows only moderate resilience when cell structures become irregular, making it suitable for applications where a trade-off between flexibility and grid precision is acceptable.. GPT-4o-mini, Granite, and PHI-3-Vision exhibit high sensitivity to layout complexity, confirming that without extensive structural training, pure generation approaches under-segment or over-segment in complex scenarios.

Interestingly, Structural-Layout F1 closely mirrors the simpler "table-shape accuracy" metric. For every model and difficulty level, the two scores differ by no more than 0.3 points. This near-perfect correspondence suggests that layout extraction performance is mainly dependent on the accurate identification of boundaries and cell segmentation. Because F1 balances precision and recall—whereas raw accuracy counts only total correct cells—the minor discrepancies reflect cases where models over-segment or under-segment by a handful of cells. To discriminate further among approaches, it becomes essential to incorporate content-aware evaluations or detailed cell-type analyses rather than relying solely on boundary-based measures.

4.6 Cell-Content F1 Score Across Multiple Similarity Thresholds

Figure 4.7 presents cell-content F1 scores at three text-similarity thresholds (60, 80, and 100) for both simple and complex tables. By varying the threshold, we show how strictly each model must reproduce the original cell text: lower thresholds tolerate minor deviations (e.g. typos), whereas a 100% threshold requires an exact match.

In the simple-table setting, every model's performance degrades as matching becomes more exacting. TATR-OCR leads at the 60% threshold with an F1 of 88.7<%, but its score falls sharply to 46.4% under perfect-match conditions. By contrast, GPT-4o begins slightly lower (85.5%) yet declines more gradually—settling at 75.1% at 100% similarity—thus overtaking TATR-OCR when exact reproduction is mandated. The smaller variants (GPT-4o-mini, Granite, PHI-3-Vision) never breach 70% even at the most lenient threshold (70.3%, 69.0%, and 62.0% respectively at 60%) and each drops by roughly 10–15 points as the bar rises to 80% and by 30–40 points at full strictness.

When table complexity increases—with multirow headers, spanning cells, and projected rows—the overall F1 scores contract further (right panel of Figure 4.7). TATR-OCR again tops the field at 81.8%, 74.8%, and 48.8% for thresholds of 60%, 80%, and

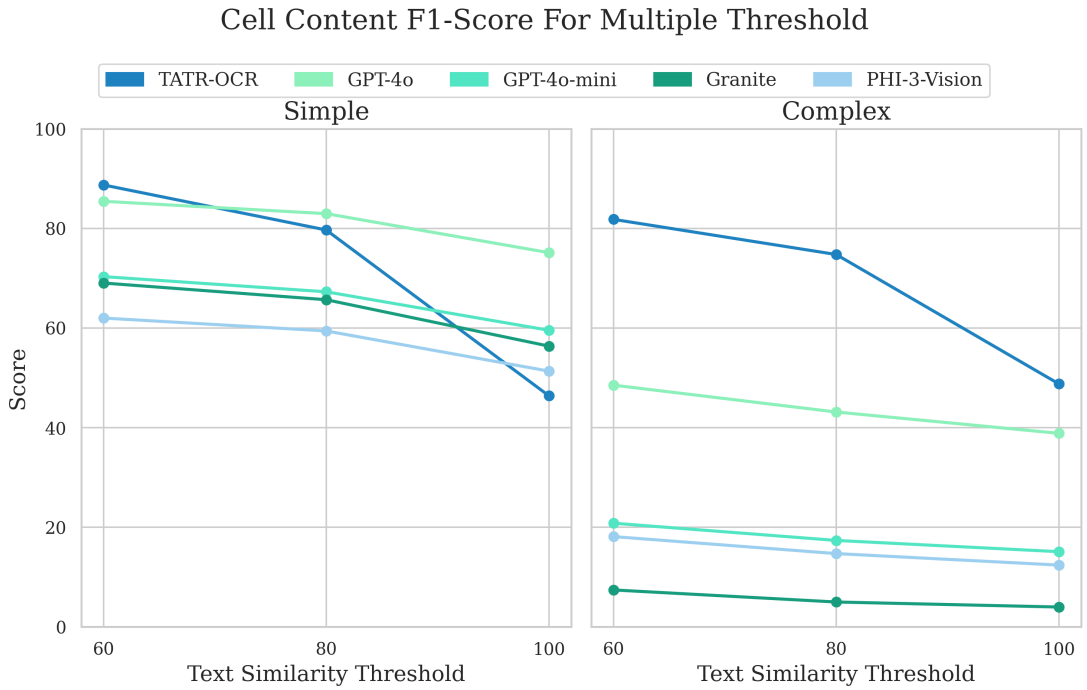

Figure 4.7: F1 score for cell content for different similarity thresholds for simple tables and complex tables.

100%, but the gap between simple and complex tables remains modest (a 6.9-point drop at 60%, 4.9 points at 80%, and just –2.4 points at 100%). GPT-4o's complex-table curve is flatter yet sits well below its simple-table line (48.5% → 43.2% → 38.9%). The three smaller models collapse to single-digit F1 scores at 100% similarity, with Granite reaching only 4.0% and PHI-3-Vision 12.4%.

Taken together, these results illustrate two key findings. First, TATR-OCR exhibits the highest tolerance to minor deviations and the best stability between simple and complex layouts. However, it suffers a steep relative falloff when exact text reproduction is required. Second, GPT-4o demonstrates superior precision under exact-match constraints, outperforming all other models at 100% similarity. The compact models, by contrast, struggle with both textual fidelity and structural intricacy, never exceeding 70% under relaxed criteria and dropping below 20% on complex tables.

4.7 GriTSCon

Figure 4.8 reports GriTSCon F1 scores, an evaluation that jointly considers cell topology, position and text similarity, for five models on both simple and complex tables.

On simple layouts, GPT-4o leads with 89.6%, closely followed by TATR-OCR at 87.8%.the others models can't even reach 80%, Granite reaching 76.3%, GPT-4o-mini at 74.6% and PHI-3-Vision at 65.2%. When tables include spanning cells and projected

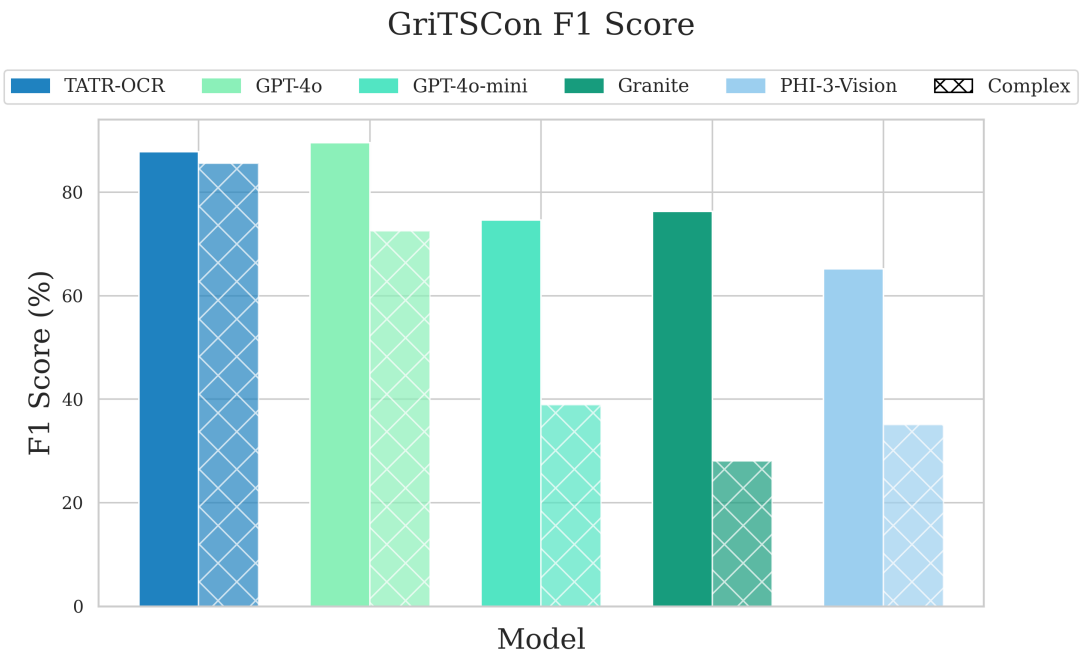

Figure 4.8: GriTSCon F1 score considering the failures and not considering failures and different similarity thresholds for simple table and complex(hatched) table.

rows, TATR-OCR remains most stable at 85.5%, whereas GPT-4o drops to 72.5%, GPT-4o-mini to 38.9%, PHI-3-Vision to 35.1% and Granite to 28.1%.

The simple–to–complex performance gap is only 2.3 points for TATR-OCR, but ranges from 17.1 points (GPT-4o) to 48.2 points (Granite) for the others. This minimal decline highlights TATR-OCR's robust integration of OCR and structural parsing, as it recovers both the correct grid and the cell contents with high fidelity, even under complex layouts. By contrast, models relying primarily on language generation exhibit larger performance losses when strict cell-boundary alignment is required, demonstrating that structure-only or content-only metrics would overstate their practical effectiveness.

4.8 Evaluation Summary

In table 4.2 is the comparison of the best and worst models across the metric and the complexity. Taken together, the information from the previous graphs indicates that TATR-OCR consistently yields the lowest failure rates, the highest structural accuracies, the strongest cell-content F1 across all similarity thresholds, and the smallest simple–to–complex performance gaps (less than 3 points). GPT-4o strikes a balance between content reproduction and structure alignment, outperforming TATR-OCR at exact-match thresholds but losing almost 17 points when tables become complex. In contrast, GPT-4o-mini, Granite, and PHI-3-Vision suffer steep drops (more than 30 points) under both layout and strict text demands. GriTSCon confirms that only

an OCR-driven pipeline can maintain grid integrity and text fidelity simultaneously, whereas pure LLM approaches excel in fuzzy content but falter on precise cell-boundary alignment. These findings underscore the necessity of unified metrics that capture both structure and content, and point toward hybrid OCR+LLM architectures as the most promising direction.

Table 4.2: Best and worst model for each metric (simple vs complex).

Metric (Simple)	**Best Model**	**Worst Model**
Failure Rate	**TATR-OCR (0.0%)**	PHI-3-Vision (25.4%)
Table Shape Accuracy	**TATR-OCR (98.4%)**	PHI-3-Vision (71.3%)
Structural Layout F1	**TATR-OCR (98.4%)**	PHI-3-Vision (71.3%)
Cell Content F1 (Threshold 60)	**TATR-OCR (88.7%)**	PHI-3-Vision (62.0%)
Cell Content F1 (Threshold 80)	**GPT-4o (83.0%)**	PHI-3-Vision (59.4%)
Cell Content F1 (Threshold 100)	**GPT-4o (75.1%)**	PHI-3-Vision (51.4%)
GriTSCon F1	**GPT-4o (89.6%)**	PHI-3-Vision (65.2%)
Metric (Complex)	**Best Model**	**Worst Model**
Failure Rate	**TATR-OCR (0.0%)**	PHI-3-Vision (48.5%)
Table Shape Accuracy	**TATR-OCR (97.8%)**	Granite (41.9%)
Structural Layout F1	**TATR-OCR (97.8%)**	Granite (39.1%)
Cell Content F1 (Threshold 60)	**TATR-OCR (81.8%)**	Granite (7.4%)
Cell Content F1 (Threshold 80)	**TATR-OCR (74.8%)**	Granite (5.0%)
Cell Content F1 (Threshold 100)	**TATR-OCR (48.8%)**	Granite (4.0%)
GriTSCon F1	**TATR-OCR (85.5%)**	Granite (28.1%)

The next chapter interprets these results, discusses their implications for end-to-end table-extraction systems and outlines avenues for future work.

5

Conclusion and Future Work

5.1 Conclusion

This study systematically compares the performance of multimodal LLMs and traditional neural networks for table extraction from PDFs and images. Across multiple structural and content-based metrics, TATR-OCR emerged as the most robust model, consistently achieving high accuracy in detecting table structures and transcribing text with minimal degradation between simple and complex tables. By contrast, LLMs,particularly GPT-4o and its mini variant,demonstrated strong capabilities in content reproduction, excelling in fuzzy matching but faltering when strict structural alignment was required. The GriTSCon metric further validated these findings, revealing that OCR-based approaches better preserve grid integrity while language models struggle with precise boundary recognition. These results highlight the trade-off between structural accuracy and content accuracy, emphasising the need for hybrid models that leverage both visual recognition and contextual language understanding.

A key contribution of this study is the generation of a structured ground truth dataset by merging JSON files (containing extracted text content) and XML annotations (bounding box details). This pipeline ensures a highly accurate comparison between predicted table structures and their original formats, enabling precise benchmarking across models. Beyond evaluation, this curated dataset presents an opportunity for fine-tuning multimodal models, allowing the LLMs to learn explicit table segmentation, hierarchical cell relationships, and structured layout reconstruction, which may enhance their performance in complex scenarios.

Another contribution from this thesis was an article of the preliminary work that compares only the simple tables, which was accepted by ACL for the First Workshop on Large Language Models and Structure Modelling, titled " Benchmarking Table Extraction: Multimodal LLMs vs Traditional OCR ". Also, a demo paper presenting an application named TableEyes was submitted to the 28th European Conference of Artificial Intelligence 2025. The demo, called TableEyes, utilises either the TATR-OCR or the GPT-4o pipelines, combined with named entity recognition models, to extract

tables from images and anonymise sensitive data.

5.2 Future Work

While this research establishes a strong comparative foundation, several avenues remain for further investigation:

- **Hybrid Architectures** – Combining the TATR pipeline to get the structure and the multimodal LLMs to get the text could improve text extraction while maintaining structural integrity. Future work should explore post-processing strategies that enable LLMs to refine OCR outputs, correcting text errors while preserving the table layout.
- **Exploring Other Models** - For future work, a comparison with other OCR models and deep learning models like Docling by Livathinos et al. (2025) that is powered by state-of-the-art specialised AI models for layout analysis (DocLayNet) and table structure recognition (TableFormer). Also, validate the LLMs that were fine-tuned for table extraction, such as TableLlava (M. Zheng et al., 2024).
- **Fine-Tuning Multimodal LLMs** – With the curated ground truth dataset, future experiments could fine-tune models on domain-specific table extraction tasks to improve their structural comprehension and text fidelity by using the images and their structured representation as a CSV file.
- **Exploring Additional Metrics** – While GriTSCon offers a comprehensive view, future studies could incorporate semantic evaluation metrics to determine whether models maintain the meaning of tables beyond just matching raw structure and content.
- **Understand Multimodal LLMs Fails** -Understand why the multimodal LLMs fail to extract valid outputs.

Taken together, these directions could pave the way for more accurate, adaptable, and scalable table-extraction systems that bridge the gap between OCR precision and LLM flexibility.

References

Burdick, D., Danilevsky, M., Evfimievski, A. V., Katsis, Y., & Wang, N. (2020). Table extraction and understanding for scientific and enterprise applications. *Proc. VLDB Endow.*, *13*(12), 3433–3436. https://doi.org/10.14778/3415478.3415563

Chen, L., Huang, C., Zheng, X., Lin, J., & Huang, X. (2023). TableVLM: Multi-modal pre-training for table structure recognition. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2437–2449). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.137

Chen, W. (2023). Large language models are few(1)-shot table reasoners. https://arxiv.org/abs/2210.06710

Cheng, M., Mao, Q., Liu, Q., Zhou, Y., Li, Y., Wang, J., Lin, J., Cao, J., & Chen, E. (2025, April). A Survey on Table Mining with Large Language Models: Challenges, Advancements and Prospects. https://doi.org/10.36227/techrxiv.174352282.22844759/v1

Chi, Z., Huang, H., Xu, H.-D., Yu, H., Yin, W., & Mao, X.-L. (2019). Complicated table structure recognition. https://arxiv.org/abs/1908.04729

Deng, I., Dixit, K., Roth, D., & Gupta, V. (2025). Enhancing temporal understanding in llms for semi-structured tables. *Findings of the Association for Computational Linguistics: NAACL 2025*, 4936–4955. https://doi.org/10.18653/v1/2025.findings-naacl.278

Deng, N., Sun, Z., He, R., Sikka, A., Chen, Y., Ma, L., Zhang, Y., & Mihalcea, R. (2024, August). Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Findings of the association for computational linguistics: Acl 2024* (pp. 407–426). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.23

Dong, H., & Wang, Z. (2024). Large language models for tabular data: Progresses and future directions. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2997–3000. https://doi.org/10.1145/3626772.3661384

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., & Wang, H. (2020). Pp-ocr: A practical ultra lightweight ocr system. https://arxiv.org/abs/2009.09941

Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B., & Pollak, B. (2007). Towards domain-independent information extraction from web tables. *Proceedings of the 16th International Conference on World Wide Web*, 71–80. https://doi.org/10.1145/1242572.1242583

GraniteVision, T. (2025). Granite vision: A lightweight, open-source multimodal model for enterprise intelligence. https://arxiv.org/abs/2502.09927

Hand, D. J., Christen, P., & Kirielle, N. (2021). F*: An interpretable transformation of the f-measure. *Machine Learning*, *110*, 451–456. https://doi.org/10.1007/s10994-021-05964-1

Kashinath, T., Jain, T., Agrawal, Y., Anand, T., & Singh, S. (2022). End-to-end table structure recognition and extraction in heterogeneous documents. *Applied Soft Computing*, *123*, 108942. https://doi.org/10.1016/j.asoc.2022.108942

Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., & Li, Z. (2020, May). TableBank: Table benchmark for image-based table detection and recognition. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the twelfth language resources and evaluation conference* (pp. 1918–1925). European Language Resources Association. https://aclanthology.org/2020.lrec-1.236/

Li, Y., Wei, Q., Chen, X., Li, J., Tao, C., & Xu, H. (2024). Improving tabular data extraction in scanned laboratory reports using deep learning models. *Journal of Biomedical Informatics*, *159*. https://doi.org/10.1016/j.jbi.2024.104735

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 1516. https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf

Liu, T., Wang, F., & Chen, M. (2023). Rethinking tabular data understanding with large language models. https://arxiv.org/abs/2312.16702

Livathinos, N., Auer, C., Lysak, M., Nassar, A., Dolfi, M., Vagenas, P., Ramis, C. B., Omenetti, M., Dinkla, K., Kim, Y., Gupta, S., de Lima, R. T., Weber, V., Morin, L., Meijer, I., Kuropiatnyk, V., & Staar, P. W. J. (2025). Docling: An efficient open-source toolkit for ai-driven document conversion. https://arxiv.org/abs/2501.17887

Lourenço, J. M. (2021). *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. https://github.com/joaomlourenco/novathesis/raw/main/template.pdf

Lu, W., Zhang, J., Fan, J., Fu, Z., Chen, Y., & Du, X. (2025). Large language model for table processing: A survey. *Frontiers of Computer Science*, *19*(2), Article 192350. https://doi.org/10.1007/s11704-024-40763-6

Microsoft. (2024). Phi-3 technical report: A highly capable language model locally on your phone. https://arxiv.org/abs/2404.14219

Musa, U., Adebiyi, M. O., Adebiyi, A. A., & Adebiyi, A. A. (2023). Development of a machine learning model for big data analytics. *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG)*, *1*, 1–6. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10124592

Nam, J., Kim, K., Oh, S., Tack, J., Kim, J., & Shin, J. (2024). Optimized feature generation for tabular data via LLMs with decision tree reasoning. *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=APSBwuMopO

Nam, J., Song, W., Park, S. H., Tack, J., Yun, S., Kim, J., Oh, K. H., & Shin, J. (2024). Tabular transfer learning via prompting llms. https://arxiv.org/abs/2408.11063

Nguyen, G., Brugere, I., Sharma, S., Kariyappa, S., Nguyen, A. T., & Lecue, F. (2025). Interpretable llm-based table question answering [arXiv preprint arXiv:2412.12386]. *Transactions on Machine Learning Research*. https://doi.org/10.48550/arXiv.2412.12386

OpenAI, Hurst, A., Lerer, A., Goucher, A. P., et al. (2024). Gpt-4o system card. https://arxiv.org/abs/2410.21276

Paliwal, S. S., Vishwanath, D., Rahul, R., Sharma, M., & Vig, L. (2019). Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 128–133. https://doi.org/10.1109/ICDAR.2019.00029

Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003). Table extraction using conditional random fields. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 235–242. https://doi.org/10.1145/860435.860479

Ranjan, A., Behera, V. N. J., & Reza, M. (2021). Ocr using computer vision and machine learning. In S. K. Das, S. P. Das, N. Dey, & A.-E. Hassanien (Eds.), *Machine learning algorithms for industrial applications* (pp. 83–105). Springer International Publishing. https://doi.org/10.1007/978-3-030-50641-4_6

Reddy, Y. (2025, November). *The ultimate guide to assessing table extraction*. Nanonets. https://nanonets.com/blog/the-ultimate-guide-to-assessing-table-extraction/

Shahriar, S., Lund, B. D., Mannuru, N. R., Arshad, M. A., Hayawi, K., Bevara, R. V. K., Mannuru, A., & Batool, L. (2024). Putting gpt-4o to the sword: A comprehensive evaluation of language, vision, speech, and multimodal proficiency. *Applied Sciences*, *14*(17). https://doi.org/10.3390/app14177782

Sharma, P. (2023). Advancements in ocr: A deep learning algorithm for enhanced text recognition. *International Journal of Inventive Engineering and Sciences*, *10*, 1–7. https://doi.org/10.35940/ijies.F4263.0810823

Singha, A., Cambronero, J., Gulwani, S., Le, V., & Parnin, C. (2023). Tabular representation, noisy operators, and impacts on table structure understanding tasks in LLMs. *NeurIPS 2023 Second Table Representation Learning Workshop*. https://openreview.net/forum?id=Ld5UCpiT07

Smith, R. (2007). An overview of the tesseract ocr engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, *2*, 629–633. https://doi.org/10.1109/ICDAR.2007.4376991

Smock, B., Pesala, R., & Abraham, R. (2022). Pubtables-1m: Towards comprehensive table extraction from unstructured documents. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4624–4632. https://doi.org/10.1109/CVPR52688.2022.00459

Smock, B., Pesala, R., & Abraham, R. (2023a). Aligning benchmark datasets for table structure recognition. In G. Fink, R. Jain, K. Kise, & R. Zanibbi (Eds.), *Document analysis and recognition - ICDAR 2023* (pp. 371–386, Vol. 14191). Springer, Cham. https://doi.org/10.1007/978-3-031-41734-4_23

Smock, B., Pesala, R., & Abraham, R. (2023b). Grits: Grid table similarity metric for table structure recognition. In G. Fink, R. Jain, K. Kise, & R. Zanibbi (Eds.), *Document analysis and recognition – ICDAR 2023* (pp. 535–549, Vol. 14191). Springer, Cham. https://doi.org/10.1007/978-3-031-41734-4_33

Soric, M. (2025). *Understanding and extracting table information from brgm documents* [Master's thesis]. École Centrale de Lyon. https://inria.hal.science/hal-04974179v1

Su, A., Wang, A., Ye, C., Zhou, C., Zhang, G., Chen, G., Zhu, G., Wang, H., Xu, H., Chen, H., Li, H., Lan, H., Tian, J., Yuan, J., Zhao, J., Zhou, J., Shou, K., Zha, L., Long, L., . . . Xiao, Z. (2024). Tablegpt2: A large multimodal model with tabular data integration. https://arxiv.org/abs/2411.02059

Sui, Y., Zhou, M., Zhou, M., Han, S., & Zhang, D. (2024). Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 645–654. https://doi.org/10.1145/3616855.3635752

Surana, S., Pathak, K., Gagnani, M., Shrivastava, V., R, M. T., & Madhuri G, S. (2022). Text extraction and detection from images using machine learning techniques: A research review. *2022 International Conference on Electronics and Renewable*

Systems (ICEARS), 1201–1207. https://doi.org/10.1109/ICEARS53579.2022.9752274

Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., Chen, Q., Zhou, H., Zou, Z., Zhang, H., Hu, S., Zheng, Z., Zhou, J., Cai, J., Han, X., ... Sun, M. (2024). Minicpm-v: A gpt-4v level mllm on your phone. https://arxiv.org/abs/2408.01800

Yenduri, G., M, R., G, C. S., Y, S., Srivastava, G., Maddikunta, P. K. R., G, D. R., Jhaveri, R. H., B, P., Wang, W., Vasilakos, A. V., & Gadekallu, T. R. (2023). Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. https://arxiv.org/abs/2305.10435

Zha, L., Zhou, J., Li, L., Wang, R., Huang, Q., Yang, S., Yuan, J., Su, C., Li, X., Su, A., Zhang, T., Zhou, C., Shou, K., Wang, M., Zhu, W., Lu, G., Ye, C., Ye, Y., Ye, W., ... Zhao, J. (2023). Tablegpt: Towards unifying tables, nature language and commands into one gpt. https://arxiv.org/abs/2307.08674

Zhang, S., & Balog, K. (2020). Web table extraction, retrieval, and augmentation: A survey. *ACM Trans. Intell. Syst. Technol.*, *11*(2). https://doi.org/10.1145/3372117

Zheng, M., Feng, X., Si, Q., She, Q., Lin, Z., Jiang, W., & Wang, W. (2024, August). Multimodal table understanding. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 9102–9124). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.493

Zheng, X., Zhong, X., Burdick, D., Popa, L., & Wang, N. X. R. (2021). Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 697–706. https://openaccess.thecvf.com/content/WACV2021/papers/Zheng_Global_Table_Extractor_GTE_A_Framework_for_Joint_Table_Identification_WACV_2021_paper.pdf

Zhong, X., ShafieiBavani, E., & Yepes, A. J. (2020). Image-based table recognition: Data, model, and evaluation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12366 LNCS*, 564–580. https://doi.org/10.1007/978-3-030-58589-1_34

A

Ethics Committee Approval

This appendix includes the official ethics approval received from the NOVA IMS Ethics Committee and MagIC Research Center for the research project titled:

COMPARING MULTIMODAL LLMS AND TRADITIONAL NEURAL NETWORKS FOR TABLE EXTRACTION FROM PDFS AND IMAGES

Project No.: DSCI2025-3-134790
Principal Researcher: Guilherme Guerra Marques Nunes
Date of Approval: 14 March 2025

The Ethics Committee confirmed that the project met the ethical requirements necessary for approval. The approval confirms the appropriate use of the PubTables-1M dataset, pending adherence to its licensing and transparency guidelines.

For full details, a printed copy of the ethics approval email is included on the following pages.

Outlook

RE: NOVA IMS | Ethics Committee - NEED REVIEW

De Ethics Committee <ethicscommittee@novaims.unl.pt>
Data Sex, 14/03/2025 15:41
Para Márcia Lourenço Baptista <m.baptista@novaims.unl.pt>; Guilherme Guerra Marques Nunes <20230754@novaims.unl.pt>
Cc Ethics Committee <ethicscommittee@novaims.unl.pt>

Dear Guilherme Nunes,
Dear professor Márcia Baptista,

Thank you for submitting your Research Ethics Checklist for review. After careful consideration, we confirm that your study meets the ethical requirements necessary for approval.

Your study compares Deep Learning models (TATR) and multimodal LLMs (ChatGPT-4o, Phi-3 Vision) for table extraction from PDFs and images. Since the dataset **PubTables-1M** is publicly available on **Hugging Face**, there are no ethical concerns regarding data access. However, please ensure the following:
• **Compliance with Dataset Licensing**: Verify that the dataset's terms of use permit academic research and publication.
• **Transparency in Data Usage**: Clearly document the source of data and any preprocessing steps in your research.
• **Bias & Fairness Considerations**: If the dataset contains structured biases (e.g., certain table formats overrepresented), acknowledge and mitigate these in your analysis.

Taking these suggestions into account, we confirm that you may proceed with the study, as we do not foresee any significant ethical concerns with the project in its current form.

Project No.: **DSCI2025-3-134790**

Project Title: **COMPARING MULTIMODAL LLMS AND TRADITIONAL NEURAL NETWORKS FOR TABLE EXTRACTION FROM PDFS AND IMAGES**

Principal Researcher: **Guilherme Guerra Marques Nunes**

according to the regulations of the Ethics Committee of NOVA IMS and MagIC Research Center this project was considered to meet the requirements of the NOVA IMS Internal Review Board, being considered **APPROVED** on 14/03/2025.

It is the Principal Researcher's responsibility to ensure that all researchers and stakeholders associated with this project are aware of the conditions of approval and which documents have been approved.

The Principal Researcher is required to notify the Ethics Committee, via amendment or progress report, of

- Any significant change to the project and the reason for that change;

- Any unforeseen events or unexpected developments that merit notification;

- The inability of the Principal Researcher to continue in that role or any other change in research personnel involved in the project.

Lisbon, 14/03/2025

NOVA IMS Ethics Committee

ethicscommittee@novaims.unl.pt

Cristina Oliveira
Gestora executiva do centro de investigação MagIC | *Executive manager of the Information Management Research Center (MagIC)*
Find out more about our research at https://magic.novaims.unl.pt/en/
Team member of RM Roadmap - Co-creating the future of Research Management (https://rmroadmap.eu/)
https://orcid.org/0000-0002-0887-7961

NOVA Information Management School
Universidade Nova de Lisboa

Campus de Campolide, 1070-312 Lisboa
Tel. +351 213 828 610

www.novaims.unl.pt

De: Qualtrics Survey Software <noreply@qemailserver.com>
Enviado: quinta-feira, 13 de março de 2025 10:58:40 (UTC+00:00) Dublin, Edinburgh, Lisbon, London
Assunto: NOVA IMS | Ethics Committee - NEED REVIEW

Thank you for submitting your ethics form and additional materials for the NOVA IMS Ethics Committee evaluation.

According to the regulations of the Ethics Committee of NOVA IMS and MagIC Research Center this project was considered to **NEED REVIEW** in order to meet the requirements of the NOVA IMS Internal Review Board.

For this reason, the NOVA IMS Ethical Board Committee will review your proposal and send you a message later on. You can expect to hear from us within **1-2 weeks** (during term time). For any clarification, or if you do not receive the e-mail in that period, please contact directly us by e-mail, including the following information:

Project No.: **DSCI2025-3-134790**

Project Title: **COMPARING MULTIMODAL LLMS AND TRADITIONAL NEURAL NETWORKS FOR TABLE EXTRACTION FROM PDFS AND IMAGES**

Principal Researcher: **Guilherme Guerra Marques Nunes**

Lisbon, 3/13/2025

NOVA IMS Ethics Committee
ethicscommittee@novaims.unl.pt

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade NOVA de Lisboa