Information Retrieval From Tables in Financial Documents

By

Waseque Chowdhury
23341100
Md. Nahid Abrar
24141098
Tanveer Ahmed Shah
22141014
Md. Sakib Hasan Chowdhury
22101086

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
January 2026.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Waseque Chowdhury

23341100

Md. Nahid Abrar

24141098

Tanveer Ahmed Shah

22141014

Md. Sakib Hasan Chowdhury

22101086

Approval

The thesis/project titled "Information retrieval from tables in Financial documents" submitted by

1. Waseque Chowdhury (23341100)
2. Md. Nahid Abrar (24141098)
3. Tanveer Ahmed Shah (22141014)
4. Md. Sakib Hasan Chowdhury (22101086)

of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 31, 2026.

Examining Committee:

Supervisor:
(Member)

Dr. Swakkhar Shatabda

Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Structured financial data including balance sheets, income statements, risk metrics need tables in order to be presented within financial documents. Tables need correct information extraction for both financial analytic work and monitoring compliance together for generation of reports. The combination of complex table elements that include cells and merged headers as well as irregular layout structures creates problems for standard processing methods. This paper constructs an information retrieval framework which assesses machine learning, computer vision together with natural language processing, detection transformer, large language models, Vision language models, convolutional neural network, optical character recognition, structured points of thought to find the most effective strategies for financial document table detection, table structure recognition and information extraction. The retrieved table relations enable clients to request financial information, historical data and comparisons between different table elements. The analysis of this research examines various tabular documents to understand interdisciplinary table detection methods through both positive and negative aspects as well as their practical utility in real-world contexts. From various case study reports, the efficiency of different models, datasets in processing multi-page financial documents, with robust cell detection and data extraction from complicated tabular layouts will define an optimized solution when addressing practical tasks. This research strengthens the connection between table detection methods and semantic retrieval technology to develop inexpensive automated platforms that retrieve financial data and perform analysis.

Keywords : Financial table data, Information extraction, Table structure recognition, Table Detection, Large language models(LLM's), Computer vision, Machine learning (ML).

Table of Contents

Chapter 1

Introduction

1.1 Background

The exponential increase of the digital financial documentation has provoked an unprecedented demand in automated information extraction systems, which are able to process the complex tabular data structures. There are finances reports such as balance sheets, income statements, cash flow statements and risk assessment reports which have important structured data that must be accurately extracted due to regulatory requirements, financial analysis and making business decisions [9]. The manual processing systems that have been traditionally in use are not only tedious, error prone and time consuming but also incapable of processing the huge amounts of financial documents that are produced by the modern financial institutions.

Financial document processing is difficult due to the variation in the table structures that are found in the documents. Financial tables mainly include uncategorized structure, cell fusion, varied header schema and borderless designs that foil the traditional optical character recognition (OCR) and rule-based extraction approach [2]. The current breakthrough in machine learning, computer vision, as well as natural language processing, has provided new paths to handling such challenges using complex detection transformers, vision-language models and large language models [17].

Hybrid systems which combines many technologies to provide improved performance in table identification and structure recognition, have showed promise in recent research while dealing solving previous problems. The study conducted by Zhou et al. [22] have emphasized the efficiency of Vision Large Language Models (VLLMs) to process financial text and structured data whereas Sheng and Xu [11] brought up a unified solution to extracting tables in both digital and image-based PDF documents. Conventional neural networks or CNN when combined with transformer have also demonstrated greater performance in dealing with complex table components and those with irregular structures [3].

The growing usage of automated document processing systems within the financial sector requires the development of powerful systems that are capable of correctly identifying

table boundaries, cell structures, extracting the textual information and preserving the semantic relations among tabular components. The systems should be very precise with different types of documents and should be able to handle multi-page financial reports with ease. Annotated dataset and evaluation measures that are specially designed to address financial document analysis have also shown to be relevant in advancing the field and ensuring real-world applicability [6].

1.2 Rational of the Study or Motivation

Banks and financial bodies across the globe produce huge data sets of important documents such as balance sheets, income statements, cash flow statements and risk assessment reports, which are structured information vital in the regulation of entities, financial forecasts and company strategic decision-making. The manual processing methods long dominant in this sphere, though labor-intensive necessitated by tradition, time-consuming and often prone to human error and now utterly unable to process the colossal volume of a variety of financial documents and financial information that can be generated by modern financial institutions, are becoming all the more inadequate to this field. This bottleneck in technology is a major drawback in the operations of the financial sector where precision and productivity are the key factors governing the working of the financial markets and where errors in the operations can have long term repercussions on not just individual financial systems but on the entire financial spiraling.

This thesis is motivated by the unique and complex nature of the problems present in tables of financial documents that has not allowed current technologies to address in a satisfactory manner. Tables with financial data have quite abnormal structures such as unusual layouts, joined cells across different rows and columns, diverse header structure, borderless table design and structure of tabular layouts that are always a challenge when it comes to using the traditional optical character recognition technology and regimes that are based on rules. Financial documents are different in nature, they can be an easy to understand income statement or a multi-dimensional risk matrix, which makes them even more complex and requires new solutions. The absence of special and high-quality labeled datasets related to the specific challenge of the analysis of financial documents has additionally undermined the progress of powerful extraction models and the research gap represents and this research paper is expected to fill.

1.3 Problem Statement

Automated systems are currently having a hard time dealing with the technical difficulties that are involved in extracting reliable information from financial documents. Financial records include compound tabular formats with dynamic arrangements, consolidated headers and table-borderless formats that baffle the conventional processing systems [1]. Financial tables are heterogeneous (simple income statements all the way to multi-dimensional risk matrices) which places significant challenges on standard OCR and rule-based extraction engines.

Among the key difficulties is proper detection and recognition of cells in financial tables. Most of the financial documents contain merged cells crossing several rows or columns, resulting in unclear boundaries which the current detection algorithms cannot detect [16]. Moreover, the extraction is also complicated by the existence of nested tables, overlapping content and different cell alignments that result in the incomplete or wrong data retrieval.

Another serious issue of financial document processing is borderless table recognition Financial tables need not include cell partition boundaries, of the sort found in academic or scientific tables and may use less intuitive formatting indicators such as alternate row colors and row height variations or implicit column alignment to structure data [8]. The existing detection algorithms are usually unable to see these visual clues resulting in the partial extraction of the tables or even failure to identify the tables altogether.

The semantic interpretation of the financial table's relationships is even more complicated. Financial documents require to keep the contextual linkages amid assorted tables elements e.g. between footnotes and specific cell, or the order in multi-level heads [14]. Existing systems have been unable to generate these semantic dependencies and hence they cannot be used to perform end to end financial analysis.

Moreover, there is a lack of high quality labeled datasets focused on financial documents analysis which prevents the creation of the strong extraction models [6]. The majority of the available datasets are on academic or general-purpose tables and are not domain-specific and of the complexity that is present in real-world financial documents, thus reducing the applicability of the machine learning techniques in this particular specialized area.

1.4 Research Objectives

The main contribution of the study is to design the extensive information retrieval system of financial documents that will overcome the existing drawbacks in the table identification, structure recognition and data extraction. The detailed research aims are as following:

i) **Design resilient table detection models:** Using powerful detection transformers and computer vision techniques [18] train resilient table detection models that can identify tables with arbitrary structure, no borders design and complex formatting hierarchy as found in financial documents. Being able to retrieve information from table datasets in useable format [1].

ii) **Task-specific evaluation measures and datasets:** Need to design task-specific evaluation measures and corpora to task of financial document analysis that are reflective of the complexity of thereal world, as well as industry standard formatting specifications to make them useful in practice [12].

iii) **Optimized computational efficiency and scalability:** Maximize computing performance and scalability of the information extraction system to handle the high throughput of multi-page financial documents within reasonable time limits and with high degree of accuracy [13].

All of these goals are oriented towards closing the gap between the existing technological capacity and the actual needs of financial document processing and finally attaining automated platforms of full financial data retrieval and analysis.

1.5 Methodology in Brief

The research uses the multi-stage, hybrid methodology to detect, recognize and extract information of the financial tables. In order to deal with the variety in the structure of financial tables, we first utilize upstream applications of detection transformers and the convolutional vision backbones, including the HybridTabNet [3] and the Hierarchical Transformer toward Table Detection (HTTD)[18]. Such models were found to be very useful in identifying boundaries of tables in scanned and digital documents; not to mention irregular tables and borderless tables. We start with raw PDF input which may be subsampled to high-resolution images. Tables can be precisely localized, whether they are of different page arrangements or language using object detection models such as YOLOv8 [8] and detection transformers [22]. The combination with image preprocessing techniques e.g, Skew correction, binarization makes the system sufficient to withstand noise or complicated data of multi-page financial reports [1][4]. This basis ensures that digital table and image-based tables are identified precisely after which the next step is taken.

The second step follows the detection of tables, which is structure recognition and cell segmentation utilizing multi-modal and CNN-transformer structure hybrids. TableVLM [5] and OmniParser [21] methods oblige our intuitive way of analyzing both visual and textual information, so the system is able to deduce the correct mapping of rows, columns and cells, even when dealing with merged or nested cell layouts. Specifically, the feature extraction architectures take a visual cue of the table, positional encoding and layout-consistent transformers improve the segmentation to maintain structural consistency [14][19]. Each cell is read by Optical Character Recognition (OCR) engines (such as optimized versions of Tesseract with additional domain-specific post-processing application). In order to make sure that cell boundaries correspond to real content, the models are dynamically up-dated, by means of feedback loops based on contextual alignment and language indicators [2][16]. This recognition key-stage is important in maintaining the semantic relationships in some of the financial figures, headers and footnotes which traditional OCR and rule-based extractors often fail.

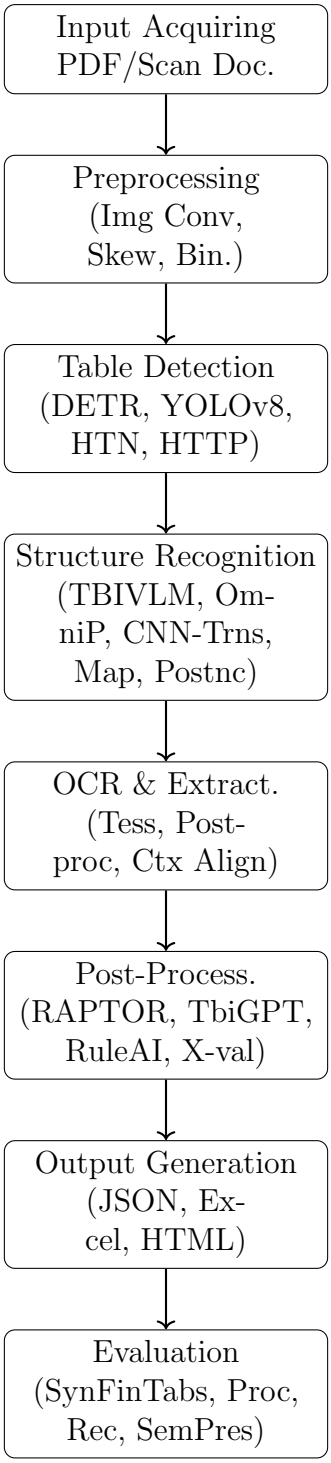


Figure 1.1: Table Detection & Information Extraction Flow

The last step converts the identified and detected table structures to machine readable languages such as JSON, Excel or HTML. This is achieved by a post-processing layer, customized to this problem, which improves the original detections and fixes misalignments applying rule-based and AI-powered heuristics. The extracted figures are cross-validated within our system against known financial schemas e.g. balance sheets and income statements in order to detect anomalies and promote fidelity [9][12]. To determine the precision, recall and semantic preservation, an evaluation component uses domain-specific

test sets like SynFinTabs [6]. Using these various but related technologies, the methodology manages to circumvent the frequent disasters in the form of uncertain table edges, repeated table cells and style inconsistencies. This pipeline system that has many stages in the chain of detection, structure recognition, content extraction and output validation constitutes a robust end-to-end that is well designed toward automating financial documents. It is not only less manpower consuming but also opening up the prospect of large volume and reliable analysis of financial data, a great requirement of the intelligent, domain-specific document processing systems [13][15][17].

1.6 Scopes and Challenges

This research problem is narrowed down to the development of a solid, automated scheme of capturing, identifying and extracting tabular-based data of various financial documents comprising balance sheets, income statements, cash flow reports and risk assessments among others. The emphasis is on Detection transformer design, multimodal vision-language models and hybrid frameworks that were able to work with any table layout, borderless organization, merged cells and nested formats that are common in financial reporting [1][2][3]. The system proposes to adopt the most modern methods, such as hierarchical transformers and CNN-transformer hybrids [3][18], to cope with multi-page reports and maintain semantic relationships between their layouts. Domain-specific corpus construction and bespoke evaluation metrics formed part of the research due to the shortage in high-quality labeled corpuses in the related application area, which is one of the main limitations of developing and testing financial systems [6]. Moreover, study limitations include PDF financial records printed and scanned but not handwritten notes and very poorly-structured notes which will be considered in future research.

Nevertheless, despite these ambitions, there remains a number of issues that prevent the solution in question to be complete. The financial tables have a reputation of inconsistent formats and fused cells and both rows and columns are irregularly aligned and contribute to a loss of clarity in detecting the boundaries and segmenting the cells, particularly on the tables with no visible borderlines [2][8]. Current algorithms have poor capabilities to parse the unobtrusive layout signals or maintain logical connections such as footnotes and multi-level headings that are vital in subsequent analysis [14]. Also, real-world financial reports have enormous differences in their language, format and noise levels and require highly flexible models with high computation efficiency and scalability to work on large batches at a minimal time [13]. Finally, the need to rely on bulk information in a labeled type of data presents a feasibility issue concerning data collection and labeling that may limit the system generalizability and cross-domain adaptation even though the future of Vision Large Language Model and unified toolkits brightens [11][22]. These limitations form the research boundary of this research and drive it ultimately to full automatization of a dependable financial information extraction.

Chapter 2

Literature Review

2.1 Preliminaries

To begin with we started our search with keywords for instance “Table Detection methods”, “Table Detection in financial documents”, “Document table structure detection”, “ Table Structure detection, “Table Content Recognition” in IEEE, Springer, ArXiv, Google Scholar etc. The searching range of the papers were from 2024 and 2025 so that we can work with the most recent findings containing the latest, well performing technologies like LLM’s, VLM’s,Transformer based, Deep CNN, OCR etc. In the beginning we had collected an immense mass of papers; then came the abstract and the conclusions, were read carefully, in order to screen it. This assisted to short list a total of twenty papers that have been most applicable to our research issue. These choices were inter-Balanced with thorough readings of every paper to make sure that we had obtained valuable information that would have directed our future work and degree of investigation. This step-by-step method was useful in trying to make sure that we are only considering more recently published articles which presented the best and best findings ever in the field.

2.2 Review of Existing Research

In a paper [22] by Yitong Zhou et al., they wanted to address the underexplored application of Vision Large Language Models (VLLMs) to table recognition tasks. They were determined to develop detailed open source financial language models that can process text and structured data in many different types of application. Their SST consists of three models developed using financial corpora, instruction datasets and pairs of images and texts and these models handle working with text, tables, charts and time series data. They use a reliable three-step method in financial AI to start with domain data, upgrade instruction with various financial works and incorporate image abilities. The system has high domain flexibility since it uses AI tools in various areas such as finance, including analyzing feelings, sorting data into groups, answering questions, trading, scoring credit, identifying fraud and processing different types of tables and charts. They encountered problems because current LLMs for finance have small corpora, lack the ability to work with different datasets and there are not enough proper measures to

prove their effectiveness. They used a method that consisted of three main steps: they trained LLaMA3-8B on 52 billion items of financial data, upgraded it through instructions covering many types of task and allowed it to process visual data by training with almost 1.43M sets that combined images and texts. To create the system architecture, they chose a modular pipeline using seven types of data for pre-training, many training datasets for fine-tuning, a CLIP vision encoder for multimodal abilities and a wide range of evaluation tools on 30 different datasets covering 14 financial tasks and trading. The groups discovered that FinLLaMA excelled in various areas, as it performed better than LLaMA3-8B, LLaMA3.1-8B and BloombergGPT in the majority of benchmarks and accomplished outstanding outcomes, such as 82.10 F1 score in NER, 85.54 in Headlines classification, 1.41 in the average Sharpe ratio for trading, 32. 65% average return in the portfolio and FinLLaMA-Instruct, a version of FinLLaMA, was better than GPT-They are significant in making the first open source LLM models for finances by adding advanced multimodal features to them, creating many benchmarks to evaluate their 14 financial uses with 30+ datasets and showing superior performance in various trading situations.

To resolve the problems they found in current table extraction toolkits, [11] Lei Sheng and Shuai-Shuai Xu made a unified solution for extracting tables from PDF files. They designed the toolkit so that it could process PDF files, no matter if they were digital or image-based and support connecting to tables via wired or wireless devices in any language. These tools read PDF files from an image or digital source and give a result in the form of HTML, DOCX and Excel files that include tables. The way Devices4Life is created is by using modules that join many open-source systems, turning them into a regular PyTorch framework meant for table extraction. Thanks to the system support for many types of PDF, tables (WLAN, WWAN) and language (English, Chinese), it is possible for businesses to define data analysis models from different domains. Handling the range of pages with different languages and formatting, as well as tables with many types, proved tough because none of the tools developed simply could not deal with all these things. They found it essential to divide this process into five: preparing the treated data, checking the document's structure, extracting its contents, recording the data and running extra applications on the tables. They consisted of data(models) for recognizing table structures (LineCell, SLANet, LGPMA, TableMaster, Cycle-CenterNet, LORE, MTL-TabNet), tools to convert text to machine data(OCR) (PaddleOCR, DuGuangOCR, TesseractOCR, EasyOCR) and systems to analyze the layout of documents (PicoDet LC-Net, DocxLayout, LayoutParser), all of which could be run interactively using standard APIs. By their type, PDFs are identified, images with wrong orientation are corrected, tables are found based on their layout, many types of tables are sorted by creating rules, suitable algorithms are selected for table structure recognition, OCR is used to get the data and it is eventually put into required data formats. The tests showed that the accuracy of LineCell is 98. 4% F1 score and 99. 5% TEDS-Struct on electronic PDFs. In the case of images, the F1 score reached 84. 2% and the TEDS-Struct reached 94. 7% at the same time. In the PubTabNet corpus, SLANet recorded 76. 31% precision, 97.01 in TEDS-Struct and completed the inference in 798 ms. All results indicated better performance and very little change from the previous toolkit. They proved the toolkit is successful by putting it to work on various sets of data.

In the article written by [6] Ethan Bradley et al., they aimed to tackle the issue of scarcity of good labeled data for using tables in finance since most datasets focus on scientific tables. This tool made it possible for people to handle various types of PDF files, such as digital and image types and fetch tables from Internet or Wi-Fi connections even in different languages. They handle PDFs and make HTML, DOCX and Excel files with all tables and their details intact. Taking this approach, turning tables into data requires several algorithms written in open-source and using PyTorch too. The system is adaptable because it handles different situations such as choices for PDFs, types of tables (wired or wireless) and languages English, Chinese and others, as picked by the model depending on the company's needs. It is challenging for crowds to deal with several table types, difficult forms, various languages and tell apart different situations, as options such as Camelot, pdfplumber and PP-StructureV2 are only capable of certain things. They set up their process in separate phases: data is prepared, the format is examined, its structure is read, all the text is taken out and then different upper-level jobs are done, all going smoothly one by one to convert numerous documents at once. The system's architecture was made by joining seven table detection models, four optical character recognition engines and three layout analysis models. Using APIs, all of these 14 tools are able to communicate with each other. There are different actions needed: recognizing the kind of PDF, deciding where the images should be, spotting tables, recognizing which tables are wired or wireless, selecting the proper table scanning method, scanning through OCR and converting data into several formats. According to what they evaluated, LineCell obtained nearly 99% precision in digital PDFs (F1 score: 98. 4%, TEDS-Struct: 99. 5%) and 8% F1 score and 95% TEDS-Struct in image-based PDFs. In a similar way, SLANet got 76.31% accuracy, excelled at TEDS-Struct with 97.01% and did inference in just 798 ms on the PubTabNet dataset. All Their main achievement is a complete deep-learning-based toolkit for parsing tables from PDF records, including scanned ones, which relies on several useful open-source libraries and is straightforward to use.

In a paper by [17] Qianqian Xie and colleagues, they wanted to address the limitations of existing financial LLMs that lack sufficient financial knowledge and struggle with multimodal inputs including tables and time-series data. The developers planned to build a group of financial language models using open-source development and these models should handle both regular financial text and well-formatted information in multiple applications. Using financial corpora, training datasets and multimodal data, their system ends up with the following models: FinLLaMA handles basic tasks, FinLLaMA-Instruct handles guiding instructions and FinLLaVA handles content combined with images and text data. They have designed an approach where AI models are continuously trained on specific information, adjusted with several financial tasks and enhanced to see pictures as well. The system is able to support various important financial use cases including analysis of attitudes and feelings, classifications, asking and answering questions, market activities, scoring credit ratings, detecting fraud and analyzing tables and charts without being trained for each job and also does so with increased generalization power. One challenge is that existing financial LLMs work on a confined set of data, miss out on using multimodal information, are mostly tested on narrow topics, do not evaluate the base

model and do not have visited test bed evaluation for trading and fintech applications. They divided their method into three main steps: constant pre-training LLaMA3-8B using massive financial data, fine-tuning the model with a large number of financial instructions related to various tasks and adding ability to handle charts, tables and other visual financial data using 1.43M pairs of images and their explanations. For the system structure, they made a modular pipeline handling seven types of pre-training data and many datasets for training purposes. The pipeline can also bring in visual data for multimodal support and it was properly tested on a wide range of datasets involving 14 financial tasks and trading options. Most of the time, FinLLaMA performed much better than LLaMA3-8B, LLaMA3.1-8B and BloombergGPT when it came to important tasks such as headline classification, trading tasks and others. They introduced a new series of financial LLMs that use several modes of input, showed how to assess their performance on 14 tasks with over 30 datasets and demonstrated that they work well for diverse applications such as trading.

In the study [9], Muhammad Azmain Mahtab and Jannatim Maisha focused on easing out the extraction of financial statements from annual reports. That is because manual work on PDF files is a major obstacle in financial monitoring organizations. They wanted to automate a schedule that would take PDF tables from financial statements and change them to Excel spreadsheets with great accuracy and in little time. The system converts PDFs in the annual reports into Excel and organizes tables into various folders in Excel automatically to help officials review them at ease. The method changes the task of extracting financial reports into a three-part pipeline with models for detection, classification and structure recognition, making the whole process automatic. It is adaptable to different areas of finance since it organizes income statements, statements of financial position, cash flow statements and statements of changes in equity for Public Interest Entities (PIEs) annual reports. It was difficult for regulators to look at numerous financial statements in a timely manner because much of the extraction had to be done manually, they had to differentiate among financial tables and useful automated systems for this type of work did not exist. For this work, they developed a three-module life cycle: YOLOv8 for detecting and classifying financial tables to using PP-OCR for processing texts and SLANet for mapping out the arrangement of data in the identified tables. To support detection, classification and structure recognition, they provide three focused datasets (FinDet for detecting tables, FinCls for classification and FinRec for recognizing structures) along with YOLOv8s (has 225 layers and 11.1M parameters) , YOLOv8n-cls (has 99 layers and 1.4M parameters) and SLANet powered by the PP-LCNet backbone and create an efficient FastAPI for PDF to Excel conversion. According to what they discovered, YOLOv8 delivered outstanding results such as 99.5% accuracy, 98.4% precision and 100% recall for financial table recognition, in just 7.2 milliseconds. Similarly, both classifications on the financial reports reached 100% accuracy within 3.4 milliseconds. At the same time, SLANet scored 0.876 TEDS points for the financial statements, finished steps 24 times faster (652ms) than CascadeTabNet (8230ms for Cascade By using a new approach, they introduced an automated pipeline for finding financial statements in annual reports, a new layer of AI classification to reduce computing time by removing irrelevant sections and also gathered three unique sets of data for analyzing financial tables with the first being a specialized dataset to classify financial statements.

In their study[14], Iftakhar Ali Khandokar and Priya Deshpande designed a computer vision system to automatically handle the extraction of content and the structure from tables in financial documents so that the data can be extracted as machine-understood rows and columns. The system works with pictures of financial documents and delivers output in JSON format containing maps of their tables, maintaining the structure of relationships between them, showing usefulness for invoices, statements and business reports, mainly in the financial domain. There were obstacles in the research involving a small number of invoice images, numerous layouts in several languages, different patterns of rules in tables and difficulties maintaining the semantic relationships between table cells. The key to their detection is a four-part pipeline, where transfer learning, deep learning and advanced models are used, notably DETR for table region detection, custom models for cell detection, a TR-OCR system for reading text from cells and Bidirectional Auto-Regressive Transformers for organizing the information collected. The process has four modules that perform one after another: it uses various public datasets in transfer learning to spot tables, a transformer model for finding table cells, an encoder-decoder architecture with ViTransformer with Roberta for reading text and finally extracts structured data with modern transformer tools using both textual and positional features. The first phase of the process is feature extraction with ResNeXt-101; then DETR finds table regions; afterward DetCNN detects and refines the cell boundaries; following this neural OCR recognizes text in cell areas using special models trained on generated data; lastly bidirectional transformers build up the JSON structure that maintains relationships between table elements. The results display a 85% success rate in identifying invoices which sets the AI apart from older OCR systems, yet less information is available compared to generalised OCR systems since this was a finance-specific dataset. The evaluation method resorted to precision, recall, F1-score and custom structural preservation measures on a financial dataset of 4,000 images as well as public dataset such as Marmot, UNLV, ICDAR and PubTabNet with approximately 509,000 images together, recording an accuracy of 85%. The main point of the work is to integrate transfer learning with transformer models for working with financial documents, use innovative techniques to create data for model learning and preserve the consistency of tables after extraction. Nevertheless, the use of these tools is limited to mainly handling financial documents, they base much of their success on prior experience with the same problem, they are computationally demanding and have challenges with several types of documents.

In the paper [3] Danish Nazir et al., where they introduced HybridTabNet, a new technique for detecting tables in scanned documents that combines a hierarchical vision foundation with up-to-date Transformer-based methods, hoping to overcome frequent difficulties involving many table layouts and frequent mistakes in identifying the right areas. It works by scanning documents and giving out accurate tablet boundary lines and segmentation masks, performing the same across several types of papers like academic ones, financial and business reports. Their research ran into several challenges, for example, it was hard for objects on different pages to stand apart from each other, there was too much similarity within tables, the chosen model needed time to learn and converge and the research had to work well even with documents of many different complexity levels. They used a strategy with two parts: first, ResNeXt-101 with deformable convolutions

was employed and then improved by using Hybrid Task Cascade network, which came with contrastive denoising for better one-to-one matching, mixed query selection for better anchor initialization and two rounds of refining bounding boxes. The system design includes a main vision path with ResNeXt-101 having cardinality=64, as well as encoders and decoders powered by Multilayer Transformers. It also has a Feature Pyramid Network for taking different scales into account and works with multiple deformable attention heads to keep computing simple. Multi-level features are extracted using ResNeXt-101, the network proceeds with Transformer improvement that adds positional data, then starts the decoder with queries based on dynamic anchors, then improves the results by refining layers and finishes with output creation by optimizing bipartite matching. It shows that the program achieved a success rate of 95.3% on ICDAR-2019-TrackA-Modern, 93.6% on ICDAR-17-POD, 100% on ICDAR-13 and 96.6% on TableBank-Latex, 96.5% on TableBank-Word and 93.5% on Marmot dataset. A comprehensive evaluation was done with precise metrics, including precision, recall, F1-score and IoU, at thresholds ranging from 0.5 to 0.9. Then the model was tested by leaving out a dataset at a time, which added up to six major benchmarks comprising over 10,000 images. As a result, the model achieved over 27% less relative error and more than 42% improved accuracy over other top models on ICDAR-2019 and TableBank-Latex. The workstands out because it unites deformable convolutions and Transformer, creates a new leave-one-out simulation system, makes it possible to do without extra processing images and helps generalize results across several datasets. At the same time, scientists found that the approach needs massive computational capability, faces difficulties because related classes are too close in ICDAR-17-POD, relies on big training datasets and could struggle in real-time use.

Jieqiong Han, Wenjing Fan, Chengxin Huo, Youcai Ma and Yang Gao designed a new model in a study [16] to improve how features of tables in power metering test reports are handled. They wanted to overcome challenges of recognizing tables that have merged cells, missing data and crowded text areas. The system receives test documents for power metering and turns them into results with better cell detection, mainly applicable in the electrical utility sector. The study team faced issues such as high numbers of unanswered and merged cells in tables, very close text spacing that concealed table edges in real time, the recognizing of facts for real-time metrological tests and main features extraction compared to present light networks mostly made for image recognition and not designed to work on measurement tasks. To achieve better results they combined CvT for better feature extraction with CSP-PAN to enhance features, even though their architecture has quadratic ROI+CVT+SCNN, with fully connected layers doing the role of semantic segmentation, making the model run faster. The system architecture has blocks called CvT networks, which introduce convolutional layers and multi-head attention, use the CSP-PAN module to process text, use spatial CNNs to look at rows and columns with direction processing and use cell vector extraction networks that combine ROI and spatial contrast. Firstly, convolutional-transformer based feature extraction takes place with CvT and the information is refined through CSP-PAN features at every spatial frequency, Next, CNN processing helps identify and outline cell boundaries and cell merging determination relies on adjacent cell data refinement implemented inside the network. Using the TEDS score, performance insights show that the model achieved an impressive 97.65%, which is more than TSRFormer (95.87%) and TRUST (95.78%) and also worked faster, averaging 685 ms per document. TEDS was used as the central measurement while including testing

on ICDAR2013 dataset having 238 images and specialized power metering sets, leading to benchmark figures with around 1.78-point better performance than TSRFormer and 1.87-point higher results than TRUST. It helps a lot through the introduction of a new method that handles complex tables, a modern PAN approach to replace FPN in feature transformation, skipping complicated post-processing steps and allowing more efficient cell merging with a better quad architecture. However, there are drawbacks involved, it is computationally resource intensive and not being able to work in all industrial domains due to specialized data needs.

In a study [18] by Mohamed Mahmoud, Bilel Yagoub, Mostafa Farouk Senussi, Mahmoud Abdalla and Hyun-Soo Kang, researchers built HTTD (Hierarchical Transformer for Table Detection) to tackle three problems: dealing with multiple document layouts, boosting how quickly the model trains on Images of documents go into the system and they generate accurate edges of tables along with scores representing confidence, making the system flexible in various areas like academic writing, businesses, or medical fields. Working on transformer models to detect tables was not easy because they met diverse table structures, experienced sluggish training, required better computing and were needed to be reliable regardless of the quality of documents. They developed new strategies, for example, contrastive denoising for solid training, mixing different query choices between DETR-like and traditional methods, twice look-forward for using precise details from later layers and a hierarchical vision backbone to keep track of various object sizes. The architecture is made up of Swin-L Transformer for hierarchical extraction of features, a Transformer encoder/decoder in multiple layers, a dynamic way of setting the bounding boxes, various prediction modules and further includes contrastive denoising and mixed query selection. First, Swin-L backbone is used to gather multiscale features then Transformer encoder improvement takes place, followed by decoding with upgraded anchor boxes, then by contrastive denoising for better training, then prediction generation with specialized layers occurs to end the process. Performance insights suggest the results are top-notch with 96.98% precision on ICDAR-2019 cTDaR, 96.43% on TNCR and 93.14% accuracy on TabRecSet and fill the camera with 25 pictures per second. Strict evaluation was carried out using precision, recall, F1 while comparing accuracy across IoU values and testing against typical benchmark data as well as running non-standard tests, such as those for breast cancer detection, where the accuracy was 56.38% and receipt key detection (36.5% IoU). The research has a major influence by using a new Transformer design, new denoising and selection methods, demonstrating how it works well for a range of detection cases and ensuring it is computationally efficient. Still, using good computation takes a lot of time, training the model on poor documents might make it less accurate, it mostly depends on large sets of training data and it has difficulties with datasets that are different from the ones used in training.

Through a study [20], Eliott Thomas et al. improved existing DETR-based models and introduced RAPTOR (Refined Approach for Product Table Object Recognition). This system helps enhance the table extraction of product tables in business documents and improves performance by detecting incorrect regions and overlapped columns. What happens is that it feeds the predictions from DETR and TATR models with table data into the

system, which outputs polished table borders and a better structure, staying best suited for viewing products and managing other types of tables decently. The research group came across some major problems like using different table formats, finding few business datasets annotated due to privacy issues and detecting errors such as above/below noise (22% of occurrences), incorrect table detection (18% frequency) and missing some data (13% prevalence). They used a wide framework that mixed modules for perfecting Table Detection with modules for recognizing Table Structure and adjusting parameters using Genetic Algorithm. Separate pipelines for TD and TSR refinement, confidence-based selection, iterative validations, clusters for noise elimination, detection using IoU and processing of spatial coordinates are parts of the system architecture. First, predictions are made based on the model, then the Table Chooser is used with multi-feature evaluation, after that Header Refiner is checked, then over segmentation is managed with clustering and the process finishes with Refining Tables for Shared Rows and handling column processing. The use of hybrid models gives a major boost in precision for product tables and the business tools meet increased standards for quality work amid stable performances on classic benchmarks, but performance drops on recall for academic datasets due to cautious content removal. Both evaluation tasks were performed with Purity, Completeness, GIoU for detection and GRITS-CON for structure recognition across five collections, made up of two business-owned datasets and three popular industry-standard datasets, proving that using a modular approach is more effective when there is not much training data to fine-tune upon. The research achieves results by reporting widespread issues in business tables, enhancing modular editing, releasing a DocILE dataset sample and setting up practical GA-based parameter tuning with little training data. Yet, some challenges are that the model has been optimized for business documents, it is less likely to spot irrelevant tables, it supports just one table per page and the focused design for business documents limits its applicability across diverse domains.

In their paper [12], Trivedi et al. wanted to introduce a system able to collect transaction details from bank statements by finding and reading tables automatically. The first thing they accomplish is locating and grouping tables in bank statements and the second step is analyzing the exact layout of tables to capture all the little transaction facts. BankTabNet was made by annotating images of bank statements, which can be given by financial institutions and used to teach the AI how to identify and organize tables in statements. Since their system aims at banking documents, it shows that it can deal with different financial documents in a similar manner. It was difficult for them to handle and mark the data in bank statements because these contain sensitive information that needs to be properly hidden. To build the model and design the system, they picked DETR as the base framework and made sure it was suited for multiple inputs by also adding Complete IoU loss. They work with bank statement pages, carry out table detection, categorize tasks, notice the structure, get text from optical character recognition and collect transaction details as part of the process. Besides table extraction, these approaches and methods are very helpful for full financial document automation. This work chose to directly apply Detectron2, utilize NVIDIA A100 GPUs during training and monitor the detection performance by using Average Precision and Average Recall. They achieved better results on data from banks rather than on data meant for all tables. The team's system had a 0.85 sensitivity for finding tables and 0.95 for table structure and was efficient for handling tables of any size within a reasonable period. Some evaluation

metrics they applied are the AP_{50}, AP_{75}, AP and AR scores at multiple IoU thresholds. The accuracy of finding tables was 85.25% and that of process organization was 83.1% AP. By conducting this research, they built the basis for using transformer AIs on specific document management duties and showed why it is necessary to have specialized data. Despite their good results on standard tests, the success of any bank statement processing is determined by the ability to handle the wide range and changing details of actual financial documents.

This study by [19] Ren et al. introduces TableGPT, which uses both computer vision traditional techniques and modern language processing methods to help with recognizing table images. First, the methodology uses transformers to examine and recognize the first table, then converts it to HTML and updates it automatically, then magnifies lower quality photos and finally links the outcomes with a skilled agent. The main contribution of the authors is the creation of TableGLM, a fine version of ChatGLM3-6B and TableGPT Agent, which chooses the best way to fix HTML errors in tables. TableGLM was found to be 4% better on average, both on public datasets and private data and the combined superresolution and LLM method showed a huge improvement of 54% for low quality images. On the TEDS scale, the TableGPT Agent does significantly better than not only GPT-4, but also other multimodal models. The paper is strong because it covers many aspects of face recognition, can be put into practical action and uses LLMs for second-stage corrections following recognition. Nevertheless, some issues need to be addressed: there are only 2,647 training images, the prompt is not very thorough and the superresolution model is not devoted to table use. At the same time, the authors achieve good results in different kinds of table recognition, even so, the multi-stage structure of their model might make it hard to implement it in practice. By adding LLM-based post-processing to computer vision, the work creates new possibilities for uses between different AI areas. Using adaptive agents for processing documents is an important progress in the field of document understanding, but experts should focus on enhancing their capabilities and try out advanced methods for images that appear in tables.

In their paper [1], Paliwal et al. discuss TableNet, a deep learning model that can do both table finding and tabular data extraction from files of scanned documents at the same time. The most important part is using a multi-task approach by linking table and column identification in one framework based on the VGG-19 network. The approach consists of a single common encoder, plus two decoder parts for identifying tables and columns and then extracts rows using known rules and OCR output. It is shown by the authors that using semantic ideas by color-coding different types of data (strings, numbers and so on) in the text helps the model learn better. The model performed well on both ICDAR 2013 and Marmot datasets and the semantic-enhanced approach got F1-scores of 0.9662 for tables and 0.9151 for structures, which were marginally higher than the baseline DeepDeSRT approach. The paper is notable as it: integrates table detection and structure recognition processes for the first time, tries out transfer learning successfully and gives accessible column structure data for the Marmot dataset. Even so, there are a few limits to the work. It follows that even though the improvement continues, it is not very big and the method used for extracting rows is based on rules rather than

learners. Although semantic feature enhancement is helpful, it needs extra preprocessing that might make its use more challenging. Small datasets are used in the evaluation and the row extraction part is still less advanced compared to the parts for recognizing tables and columns. It would be useful to carry out more detailed ablation tests in order to recognize how each module affects the model's performance. While there are limitations, the project displays how related tasks in table understanding can be made better by using multi-task learning, benefiting further improvements in table understanding.

This study paper [2] outlines a full process for pulling data from financial documents containing tables, which is an important issue in financial services because tables are commonly hidden inside pixelated text. The authors separate the difficult task into identification of tables, getting the content inside them and aligning them before applying deep learning for each sub-task. For finding tables, researchers segment stages of an image using U-Net and DenseNet-169 and they add an extra class, called a separator, to divide very nearby tables into distinct regions. Tesseract is used in the OCR part and this is assisted by Otsu thresholding, auto rotation and morphological kernel filtering for removing table borders from the image. This phase is quite complex, as it uses union-find data and various sequential models, such as LSTM and Transformer, to put the text into the correct format of a table. The evaluation was thorough, done on 10,000 pages from different financial documents, each with 1,660 hand-annotated tables and the system reached an accuracy of 83% on unknown pages. Finding the best-performing architecture, it was noticed that LSTM with table context achieved the best results with 10.50% of perfect alignments. Even though the research makes solid technical and practical points, the evaluation metrics are based mostly on visual review instead of accurate numbers, using just financial data might make it hard to apply more widely and operating the pipeline using so many deep learning models is not fast enough for real-time deployment. Also, including a comparison of the paper's accuracy to that of other commercial solutions and performing further analysis of the errors would further enrich the report. In any case, the system reaches a new high in document understanding, especially because of its innovative module structure that lets engineers upgrade each part step by step and its distinct system for cutting tables from paragraphs. Since it addresses the entire process: detection, alignment and pays close attention to the characteristics of financial documents, the method stands out and can be very helpful in processing financial data.

In this paper [15], a hybrid approach that applies AI to extract data from invoices has been developed in this paper to increase how consistently data can be added to Enterprise Resource Planning (ERP) systems. The authors link two deep learning frameworks: DETR (DEtection TRansformer) and ResNet-18 are used for table detection, while the other two, TATR-v1.1 and YOLOv8, are used for component detection and EasyOCR is used for collecting the text from the table. For evaluation, 256 images of different table layouts make up the custom data set and the data is expanded using different augmentation approaches to obtain 19,800 images for table detection and 6,000 images each for the various component detection parts. When models are built for individual row, column and header detection rather than joint detection, DETR+YOLOv8 gets superior metrics, especially because of its mAP50-95 of 97.5%. Using rules to improve cell boundary

detection made the software perform 2% better, as EasyOCR could recognize 82% of the text. It is clear from the paper that the model improvement helped solve a company's day-to-day problem. Its drawbacks, however, are that the dataset may hinder its generalization to other formats, a comparison with commercial solutions was not done and while the method is very accurate, it is slow to complete on each page. More in-depth studies of errors and failures would also be of great value, especially to highlight the huge performance difference between YOLOv8 and TATR-v1.1 (97.5% vs 9.0% mAP50-95). The contribution of the work lies in automating document processing for ERP systems, showing that having models specialized for certain tasks gives much better results than using general models. Nevertheless, it would be good if more analysis is done to see the method can run effectively on bigger tasks, the dataset can be improved and detailed analyses are made to find out how this hybrid method marks differences against other state-of-the-art commercial methods in practical applications.

This paper [5] deals with the detailed problem called Table Structure Recognition (TSR), which focuses on rebuilding the structure of tables shown in images, mainly for tables with multiple row headers, multiple column headers and merged cells. For that purpose, the authors present TableVLM, a detailed multi-modal transformer that combines both visual and textual inputs, with a two-stream encoder-decoder approach. One of the main tasks here was building the ComplexTable dataset, including over a million images of complex HTML tables with all kinds of borders, different table cell alignments and cells that span rows or columns, while 75% of the tables have merged cells. The TableVLM model uses five pre-training tasks, chosen among them are Text-Image Alignment, Text-Image Matching, Masked Image Modeling, as well as two new one Colum RoBERTa and Mask-RCNN are used by the model's encoder to produce embeddings for text and visual parts and the model relieves on 2D layout positional features from LayoutLMv2 for the same purpose. Training on many popular benchmarks including TableBank, PubTabNet and ComplexTable, TableVLM was found to outperform other models by up to 1.97% on TEDS, the metrics that ComplexTable uses. With an ablation study, it was shown that every part of MIM matters and that together, these elements improved the TEDS score by up to 2.68%. The model has an encoder of 12 layers and about 200 million parameters, plus a 4-layer decoder that was trained for a long time but with different sequences of training data and rates. Even though TableVLM is strong, it has problems with old or handwritten records due to the way it is trained and needs to be adapted to these domains. In short, TableVLM stands out as a notable improvement by mixing visual and textual data with unique approaches and by creating the biggest and most diverse dataset of synthetic tables, so it enables important developments in reading documents and business intelligence fields.

In the paper [21], with OmniParser V1, a first-ever single type of network handles VsTP, solving the challenges in text spotting, gathering important facts (KIE) and table recognition from documents and images, a field in which today's methods are either completely specialized or too general to be useful. As a whole solution, without OCR and for any scenario, OmniParser works on image inputs and text instructions to produce results that include text and geographic coordinates. Its main design feature is a two-step process:

first, an SP Decoder finds center points with tokens such as ¡tr¿ and ¡Date¿, then the tasks of identifying regions and creating content are separated and managed by dedicated Region and Content Decoders, each informed by 16 points and character words. This model relies on SwinB Transformer base and Feature Pyramid Network to propose six outputs and uses three separate 4-layer decoder transformers. Improving both the presentation of images and the link between image and text, the new schemes, Spatial-Window Prompting and Prefix-Window Prompting, help the decoder pay attention to various image parts and assist with disambiguation of character sequences in words. Superior end-to-end performance is shown by OmniParser in every single benchmark, including: Total-Text, CTW1500, ICDAR2015 for text spotting and CORD, SROIE for KIE, as well as PubTabNet and FinTabNet for TR. OmniParser scores better all-around, beating the strong Donut and EDD models when it comes to finding and recognizing tables in data. Results of ablation tests support that using modular decoders, a Swin-B network as the backbone and training with prompts are important, while the system runs efficiently when using a modest decoding length. It is easy to see that the model handles recognition of curved and artistic words well, organizes KIE maps accurately and represents table data including various kinds of overlapping rows and columns. Even though OmniParser currently factors in only text and needs comprehensive annotations, it creates a pioneering technique for combining complex text-related tasks in a single architecture that is helpful and clear.

As the use of technology rapidly grows across Vietnam, the study [4] sets out a way for easier, efficient extraction of table-based data from statements used by academia and the financial sector. In a Vietnamese university, the presented system addresses the main inefficiencies of manual entry of bank statements into the financial software, such as spending a lot of time, sometimes making errors and working harder. Using a straightforward image approach instead of large machine learning libraries, the method goes through four steps based on the classical computer vision tools OpenCV and OCR with Tesseract. First, preprocessing changes PDF pages into binary images good for morphological analysis; then, various parts that are tables are recognized by finding lines and setting out-bounding boxes; afterward, grouping cells based on their shapes occurs, finished off by taking out text using OCR for columns. The data is put together automatically in Excel sheets and each page of the PDF makes up a separate file in the output. According to results on confidential test files with 6 and 10 pages each, the method provided superior field-level performance, giving perfect results for dates, very good results for numbers and around 98% for reference numbers, though the results for free text ranged from 83.5% to 95.6% and minor changes are due to the presence of different special characters and inconsistent formatting. Even though it is effective with clean, structured data, solving problems with distorted tables, merged cells and unformatted content is still difficult for this approach and it has not yet been compared to top deep learning systems. Nonetheless, the system helps a lot in automating routine financial processes and future efforts will strive to improve accuracy by correcting possible errors, review its performance with neural models, explore larger data types and eventually turn it into a practical user-friendly app made up of GUI and RPA technologies.

Authors of the paper [8], Recognizing Irregular Table Structures in Technical Datasheets, introduce a new and efficient system to find the layout of irregular, borderless and text-heavy tables in technical sheets by using YOLOv8, a proven object detection framework. Bastian Engelmann's student Silvio Langa at the Technical University of Applied Sciences Würzburg-Schweinfurt developed the study, which uses a dual-network that has each YOLO model find rows and columns separately and the intersection of their bounding boxes form the table's layout. Thanks to being guided by the design goals of accuracy, robustness and speed, this methodology can escape both the issues known in rule-based approaches and the extra processing time seen in transformer or graph approaches. The data used for training the model were ICDAR2013 and a set of technical datasheets custom labeled with their own labels. The model was also trained using PyTorch and Ultralytics and its training pipeline was included with different enhancements, such as mosaic augmentation and the AdamW optimizer. Performance on testing completely outdoes the old state of the art and shows impressively strong results: F1-scores were 0.980 and mAP@50-95 was 0.967 on the difficult datasets. The model was robust and correct since the results of number-crunching and extended tests proved its reliability with irregular documents, merged cells and multi-line content. Currently, the system works with OCR tools from external sources to get text from the cells and isn't capable of reading nested tables or dealing with unclear cues on a page. Still, the authors are proposing to make enhancements that include in-system OCR, better multi-table processing and semantic understanding for all aspects of the page. Thus, this research provides the first solution to use YOLO for recognizing tables, making it possible to address the divide between techniques that involve human-crafted rules and advanced Neural Network processing of documents.

In this study [13], the way ten leading PDF parsers work by testing them on DocLayNet in six areas of notable documents, including rule-based, hybrid and transformer-based approaches. Noticing that PDFs are everywhere (there are more than 2.5 trillion globally) and that Retrieval-Augmented Generation (RAG) increasingly relies on precisely extracting textual and tabular data, the authors say the current tools are not good enough because they mainly handle academic works and cannot be applied to different kinds of formats. Included in the evaluation are financial documents, manuals, scientific articles, legal texts, patents and government tenders; DocLayNet offers about 80,000 annotated pages sorted into eleven categories, all with detailed JSON-ground truth. Levenshtein-based F1, BLEU-4 and Smith-Waterman scores are used as important indicators for text extraction. On the other hand, Jaccard, IoU and thresholded F1 scores are used for table detection performance evaluation. In general, rule-based parsers such as PyMuPDF and pypdfium2 do well in all types of texts, while Nougat (for scientific materials) and TATR (for tables) excel in domains that are challenging for ordinary text extraction. Particularly, Camelot and Tabula do not execute well at detecting tables that appear nested or convoluted, but TATR's learning process handles them more effectively. According to the report, parser performance is mostly shaped by the type of document and learning-based approaches are more reliable, though they use more computer resources. Authors also mentioned that Nougat tested only a small number of labels and did not look at OCR in hybrid tools such as Unstructured. All in all, the paper suggests more work on models that mix parsing logic with large language models, broader review of tasks using OCR and vision-language mixing and continuing research into model reliability issues that are

especially important as document understanding increasingly involves LLMs.

2.3 Summary of Key Findings

Recent studies establish that transformer-based architectures and multimodal pipelines make sense in financial table's extraction. Hybrid models based on hierarchical vision backbone and deformable transformers e.g. HybridTabNet [3] attain the state of the art, with relative error improvement of 27% against dataset such as ICDAR-2019 that have significantly complex layout. Vision-Large Language Models (VLLMs) have become especially promising and the FinLLaMA model proposed by Zhou et al. [22] showed 82.10 F1 on financial NER and earned 32.65% portfolio returns due to the addition of domain-specific pretraining on 52B financial tokens. Equally, unitary toolkits such as PDFTable [11] exploit modular pipelines to cope with different forms of PDFs and to obtain 99.5% TEDS-Struct when integrating digital PDFs and 94.7% when integrating scanned documents, through a refinement of combinations of engines of LineCell, SLANet and OCR. These methods produce superior results in all cases to rule-based systems and in borderless tables, where solutions based on YOLOv8 achieved a F1 of 0.980 by interpreting implicit visual notations such as row distance and alignment [8].

Major trends are that real-world applicable solutions require task-specific evaluation and specialized datasets. Financial corpora e.g., Mahtab et al. FinDet/FinRec datasets [9] are used to train models so that YOLOv8 can achieve the 100% recall on financial tables in 7.2ms, whereas Bradley et al. SynFinTabs [6] is used to alleviate limited data even having complex structure to detect these tables. Nevertheless, maintaining semantic relationships are still difficult and bidirectional transformers [14] can only partially represent footnote links and hierarchical headers. Approaches differ widely in their computational footprint: SLANet takes 798ms [11] to process a table and systems that replicate the entire chain end-to-end such as TableGPT [19] introduce latency of the LLM post-processing to increase TEDS scores by 4% It is worth highlighting here that when faced with cross-domain tasks e.g. RAPTOR [20] will perform well on product tables but will have difficulties with academic reports, there is always a trade-off between specialization over generalization. In general, the combination of OCR, VLMs and LLMs creates a strong basis on which the automation of financial documents can be based, although scalability and semantic consistency have to be further optimized.

Chapter 3

Requirements, Impacts & Constraints

This chapter outlines the technical specifications, societal and environmental impacts, ethical considerations, applicable standards, project management approach, risk mitigation strategies and economic analysis for the proposed financial document table extraction system.

3.1 Final Specifications and Requirements

The successful implementation of the two-stage financial document table extraction pipeline necessitates specific hardware and software configurations to ensure optimal performance and reliability. This part depicts the finalized specifications and requirements for system deployment used in the research.

Hardware Requirements

In the table 3.1 the minimum and recommended hardware specifications is presented to deploy the introduced system.

Table 3.1: Hardware Specifications and Requirements

Component	**Minimum Requirement**	**Recommended**
GPU	NVIDIA GPU with CUDA support	NVIDIA T4/P100/V100
GPU Memory (VRAM)	12 GB	16+ GB
System RAM	16 GB	32+ GB
Storage	50 GB SSD	100+ GB NVMe SSD
CPU	4-core processor	8+ core processor
Network	Stable internet connection	High-speed broadband

The memory consumptions of these two Vision-Language Models OCRFlux-3B (around 8 GB) and Qwen2-VL-2B Instruct (around 6 GB) meets the GPU memory requirement for conducting the research. The additional 16+ GB VRAM is prescribed to allow comfortable operation with separate to batch processing and intermediate storage.

Software Requirements

The Table 3.2 depicts the required software dependencies & their version.

Table 3.2: Software Dependencies and Version Requirements

Software/Library	Version	Purpose
Python	3.9+	Programming language runtime
PyTorch	2.0+	Deep learning framework
Transformers	4.35+	Model loading and inference
CUDA Toolkit	11.7+	GPU acceleration
qwen-vl-utils	Latest	Qwen2-VL processing utilities
Pillow	9.0+	Image processing and manipulation
pandas	2.0+	Data manipulation and analysis
openpyxl	3.1+	Excel file generation
lxml	4.9+	XML/HTML parsing
markdown2	2.4+	Markdown processing

Model Specifications

The system made with two pretrained Vision Language Model with the mentioned specificaitons stated in Table 3.3.

Table 3.3: Model Specifications

Specification	OCRFlux-3B (Stage 1)	Qwen2-VL-2B (Stage 2)
Parameters	3 Billion	2 Billion
Model Size	∼6 GB	∼4 GB
Input Resolution	Up to 1024×1024 px	Dynamic (256-1280 tokens)
Max Output Tokens	4096	4096
Inference Precision	FP16/BF16 (auto)	FP16/BF16 (auto)

Input/Output Specifications

The proposed system is carried out by the following input and output specifications:

- **Supported Input Formats**: JPEG, PNG, JPG
- **High Input Resolution**: Resized 1024 pixels (for larger dimension done in preprocessing)
- **Output Formats**: JSON (structured data), Excel (.xlsx), CSV
- **Processing Capacity**: Batch processing with periodic memory management

3.2 Societal Impact

Automated financial documents table extraction has its own impact in society. As it has a significant implications of systems that ought to be taken seriously for ease usage.

Positive Societal Impacts

Improved Financial Access: The specified system will liberalize the possibilities of the financial data digitalization. The companies and individuals that were unable to afford manual data input in the past or expensive commercial systems are now capable of doing it in a meaningful manner and extract structured data in a document to make relevant financial choices and become more financially literate.

Workforce Transformation: All these routine data extraction processes are automated through the system and in the process financial professionals are able to commit their skill and expertise in other activities that are more value adding such as analysis, strategy formulation and client consulting. Not only is this change making the work in the financial field more enjoyable to work in but it also makes not simply push it away.

Better Accuracy and Reliability: There are chances of error in human data entry especially during handling of high amounts of financial records. The automated system minimizes transcription errors thus enhancing the quality of financial records and downstream analyses which rely on quality data.

Financial Inclusion: An efficient document processing can speed up the loan applications, account openings and other financial products, which will involve document verification. This speed is especially helpful to underserved populations that can experience hold ups in processing by the manual processing backlog.

Potential Concerns and Mitigation

Employment Displacement: Although automation has the potential to lower the demand related to manual entry of data, in the past, automation has led to the appearance of new job segments. Organizations should reskill the affected workers who should be trained in fields like supervisory, quality assurance or analyst.

Digital Divide: The system is not initially accessible to a population without computational resources and technical expertise, which may increase the disparity between populations with computer access and those without. This can be addressed by deployment using cloud-based services that have user friendly interfaces.

3.3 Environmental Impact

Environmental impact of artificial intelligence systems has become one of the key points of responsible technology development. In this section we are going to look at the consequence of proposed system on the environment.

Computational Energy Consumption

Large Vision-Language Models do not provide extreme power efficiency in terms of energy usage because their operation needs significant computational resources. Table 3.4 shows the estimated figures of energy of the proposed system.

Table 3.4: Estimated Energy Consumption Metrics

Metric	Estimated Value
GPU Power Draw (Active Processing)	250-300 W
Average Processing Time per Document	15-30 seconds
Energy per Document	1.0-2.5 Wh
Carbon Footprint per 1000 Documents	0.5-1.5 kg CO_2e*
*Varies based on regional electricity grid carbon intensity	

Environmental Mitigation Strategies

A number of measures have been included or suggested to reduce environmental impact:

- **Efficient Model Selection**: The 3B and 2B parameter models have been chosen as an efficient way to go a tradeoff between computation efficiency and performance, without unnecessary large models.
- **Batch Processing Optimization**: When documents are processed individually, it is not as efficient as processing documents in batches. The GPU usage, minimizes the idle power consumption.
- **Cloud Infrastructure**: Running on cloud types (Kaggle, Google Colab) uses data centers with more renewable energy holdings
- **Inference-Only Operation**: The system uses pretrained models without any further training, as opposed to the large costs of energy consumption of model training.

Comparative Environmental Analysis

Through a comparison with manual data entry which involves human labor and related changes, it will need constant human labor and related changes office infrastructure (lighting, heating/cooling, computing equipment), automated extraction can have environmental benefits on a larger scale by having lower per-document resource use and dropping of physical document transportation towards a centralized one processing.

3.4 Ethical Issues

There are a number of ethical concerns that ought to be addressed due to the processing of financial documents managed to guarantee responsible deployment of systems.

Data Privacy and Confidentiality

The financial documents bear sensitive personal and institutional information such as there are account numbers, transaction histories, balances and personally identifiable information. Ethical implementation entails:

- **Data Minimization**: Discussing the information that is required to perform the extraction task

- **Secure Processing**: This involves execution of documents in secure environments to have the right access controls
- **No Data Retention**: Implementing policies to delete processed documents and extracted data after delivery to authorized recipients
- **Encryption**: Employing encryption for data in transit and at rest

Algorithmic Transparency

The process which involves the usage of deep learning models brings lack of transparency. In order to address this:

- The method offers an intermediate output (Markdown tables) which is human enabled for checking the accuracy of extraction
- Error is flagged in the system warning users of possible extraction errors
- Documentation is clear enough to explain the capabilities and limitations of the system

Bias and Fairness

The trained models can be biased depending on the data used for training them. The considerations that were made was:

- **Document Format Bias**: Reports on document format performances vary among performance that is dissimilar among monetary institutions or regions
- **Language Bias**: The current application works with English documents and this would cast a dark cloud on users whose language is not English
- **Mitigation**: Regular evaluation of various sources of documents and perpetual evaluation model reversals in an attempt to remedy the perceived biases

Accountability and Liability

To have a well-defined accountable structure while making mistakes during extraction that may take place lead to financial harm. Users should be informed that automated extraction requires human verification for critical applications and the system should not be positioned as a replacement for professional financial advice or auditing.

3.5 Standards

The development and deployment of the financial document table extraction system adheres to relevant industry standards and best practices.

Document Processing Standards

- **ISO 32000 (PDF)**: Compliance with PDF standards for document input processing

- **ISO/IEC 10918 (JPEG)**: Adherence to JPEG standards for image processing
- **W3C HTML5**: JSON and structured output generation following web standards

Data Format Standards

- **ISO 8601**: Date and time formatting in extracted data
- **ISO 4217**: Currency code representation where applicable
- **JSON (ECMA-404)**: Structured output conforming to JSON interchange format standards
- **RFC 4180**: CSV output following comma-separated values format specifications

Software Development Standards

- **PEP 8**: Python code style guidelines for maintainability
- **IEEE 830**: Software requirements specification documentation
- **ISO/IEC 25010**: Software quality model for system evaluation

AI/ML Standards and Guidelines

- **IEEE 7000**: Ethical considerations in system design
- **NIST AI Risk Management Framework**: Risk identification and mitigation approaches
- **EU AI Act Guidelines**: Classification and compliance for AI systems processing financial data

Financial Data Standards

- **PCI DSS**: Payment Card Industry Data Security Standard principles for handling financial information
- **SOC 2**: Service Organization Control guidelines for data security and availability

3.6 Project Management Plan

The primary steps for the thesis project were based on the analysis of the background problem which was gained from the literature review. During this phase, it allowed us to further complete our study on CNN architecture, Transformers, LLMs, VLMs, Multimodal-based and OCR methods. We started by compiling around thirty-six previous research articles from academic sites like IEEE, ArXiv, MDPI, Springer, ACL etc. in order to create a comprehensive view of the current study. These publications were filtered on the grounds of their abstracts and conclusions concerning the field of interest under research. Out of them, we selected the top 20 publications that best matched the paper's research objectives. Because each article contained a large amount of materials, each group member was given a specific piece to study and evaluate the methods used

in every paper. After that, we pooled our thoughts to create a section that would serve as the foundation for our future literature review and assist us in defining the paths for more study in the field of our choice.

We have access to an extensive database of imaged datasets of documents with tables from the financial sector as well as other fields, which we can use to test various models. However, in order to improve the process, we must benchmark our models using a synthetic actual dataset of particular bank bills or documents, which may include PDFs, pictures and other types of documents. This will make it easier for us to affect the development of our models through pre-processing by enabling us to employ adjusted datasets. After that, we then developed baseline models which formed the basis of development of state-of-the-art models in the final step. Another part of this step was the development of a poster presentation, which gave us the possibility to briefly note the preliminary results and procedures. In order to build a complicated model in the following phase, the goal was to establish a solid foundation using the data and the basic model.

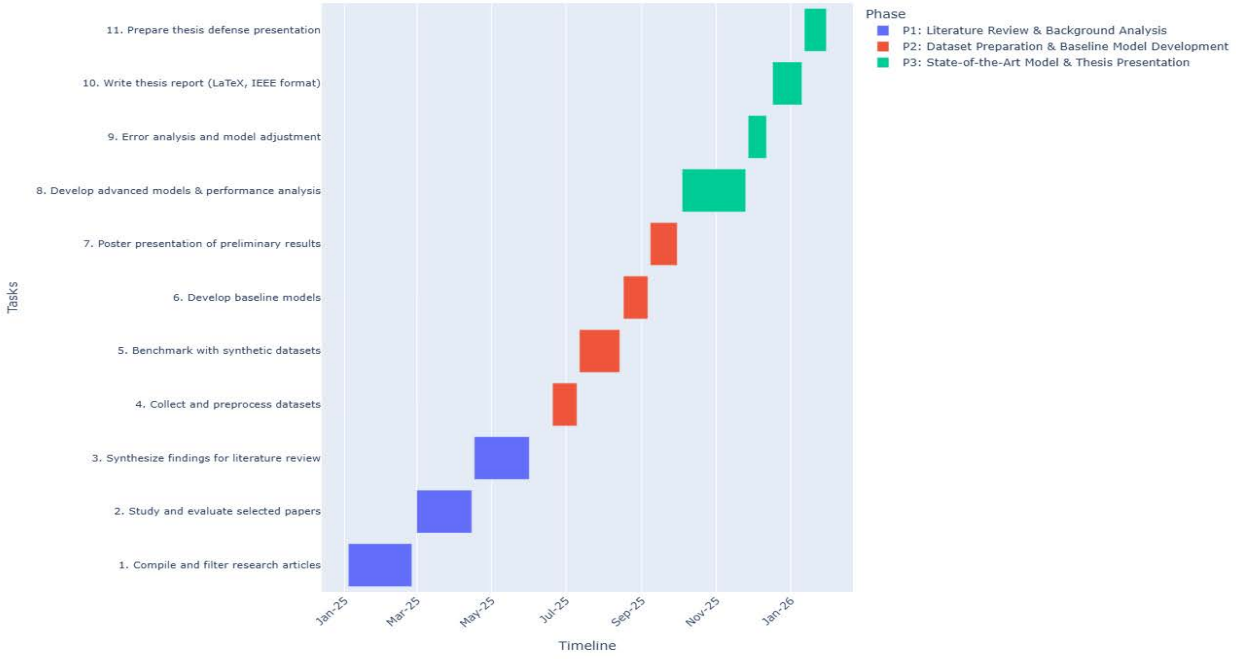


Figure 3.1: Thesis Workflow Timeline (Gantt Chart).

The final part of the thesis involved state-of-the-art model development, performance investigation and thesis presentation. During this stage, an attempt was made to create a more effective performance model than the simplest baseline created in the previous stage. Here, the proposed performance and error analysis were used to adjust the model and predict well. The result of the conducted study is presented in the form of a report in LaTeX according to IEEE standards. In addition, we adopted PowerPoint presentation in defense of the thesis which serves as a mode of presenting research findings. Our thesis report in LaTeX and PDF formats are our final products that are formatted in IEEE guidelines, ready to submit our work to be published. This is the final stage of the research activity which illustrates to the community, what is the value we have put into

this research and what are the outcomes of our findings.

3.7 Risk Management

Risk management was imperative in the project life-cycle. This section lists the possible risks, likelihood, impact of the risks and the mitigation measures that were taken.

Table 3.5: Risk Assessment and Mitigation Strategies

Risk	**Likelihood**	**Impact**	**Mitigation Strategy**
GPU Memory Overflow	High	High	Implemented dynamic image resizing to 1024px maximum; added memory clearing between stages
Model Unavailability	Low	High	Cached model weights locally; documented alternative model options
Poor OCR Accuracy	Medium	High	Extensive prompt engineering; iterative refinement on training set
JSON Parsing Failures	Medium	Medium	Implemented multi-strategy parsing with regex fallback
Dataset Quality Issues	Medium	Medium	Manual verification of ground truth annotations; stratified sampling
Processing Time Exceeds Limits	Medium	Low	Optimized batch processing; implemented timeout mechanisms
Model Hallucination	Medium	High	Added explicit anti-hallucination instructions in prompts
Dependency Conflicts	Low	Medium	Used virtual environments; documented exact version requirements

Risk Response Strategies

Avoidance: Unproven technologies risks were evaded by choosing proven well-established technologies pretrained models by the highly regarded sources (Hugging Face, official repositories)

Mitigation: Technically, it mitigated its risks including memory overflow and parsing failures by use of defensive programming techniques, through error detection and correction

Transfer: The risks concerning computational resources were partly passed over through the usage of cloud managed GPU infrastructure (Kaggle)

Acceptance: Minor risks like variability in processing time were accepted properly record keeping of anticipated performance levels

3.8 Economic Analysis

This section will provide the economic analysis of the proposed system taking the development into consideration expenses, operation cost and possible payoff.

Development Costs

The Table 3.6 provides the estimated costs for the development of this research.

Table 3.6: Estimated Development Costs

Item	**Unit Cost**	**Quantity**	**Total Cost**
Developer Time (hours)	\$50/hr	400 hrs	\$20,000
Cloud GPU Resources	\$2/hr	100 hrs	\$200
Dataset Preparation	\$0.50/image	700 images	\$350
Software Licenses	-	-	\$0 (Open Source)
Total Development Cost			**\$20,550**

Operational Costs

For production deployment, the estimated operational costs are presented in Table 3.7.

Table 3.7: Estimated Monthly Operational Costs

Item	**Monthly Cost**
Cloud GPU Instance (on-demand)	\$500-1,000
Storage (100 GB)	\$20
Network/Bandwidth	\$50
Maintenance and Monitoring	\$200
Total Monthly Operational Cost	**\$770-1,270**

Cost-Benefit Analysis

The economic viability of the system is assessed by comparing automated extraction costs against manual alternatives.

Manual Data Entry Costs:

- Average time per financial document: 10-15 minutes
- Labor cost at \$20/hour: \$3.33-5.00 per document
- Error correction overhead: Additional 10-20%
- Effective cost per document: \$3.67-6.00

Automated Extraction Costs:

- Processing time per document: 15-30 seconds
- Computational cost per document: $0.01-0.02
- Quality assurance sampling (10%): $0.05 per document
- Effective cost per document: $0.06-0.07

Cost Savings: The automated system offers potential cost reduction of approximately **95-98%** compared to manual data entry, with break-even achieved after processing approximately 4,000-5,000 documents.

Return on Investment

For an organization processing 10,000 financial documents monthly:

- Manual processing cost: $36,700-60,000/month
- Automated processing cost: $600-700/month + operational overhead
- Monthly savings: $35,000-58,000
- ROI timeline: Development costs recovered within 1 month

The economic analysis demonstrates strong financial viability for organizations with moderate to high document processing volumes, with substantial cost savings and rapid return on investment.

Chapter 4

Proposed Methodology

4.1 Design Process or Methodology Overview

The methodology diagram presented in Figure 4.1. It start by taking image input dissected from pdf or raw image dataset. Then it is pre-processed by filtering and undergoes split into train-test and validation set for proper evaluation. After feeding the preprocessed data into the selected pre-trained models and evaluating the outcomes to suggest an updated architecture, this process comes to a conclusion.

4.2 Preliminary Design and Model Specification

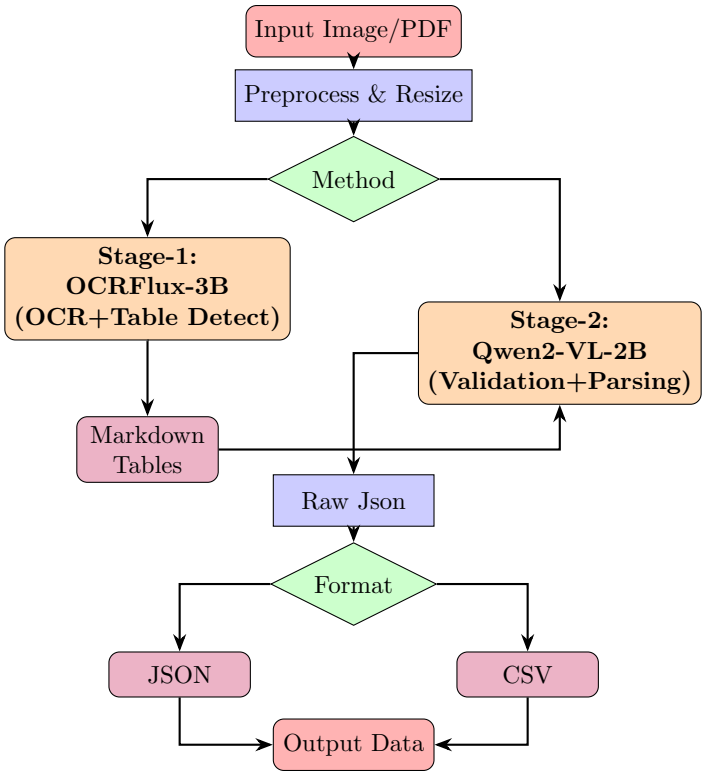


Figure 4.1: Two-Stage Table Extraction Pipeline using OCRFlux-3B and Qwen2-VL-2B

In the above Figure 4.1, a two-stage table extraction pipeline is implemented in this research. The workflow begins with input document images that undergo preprocessing and resizing. Subsequently, the system employs a two-stage methodology: Stage-1 uses OCRFlux-3B (3B parameters) for table detection and OCR, generating Markdown representation of all identified tables. The extracted Markdown along with the method decision flow is then fed into Stage-2, which uses Qwen2-VL-2B (2B parameters) for validation and parsing, producing raw JSON output. Finally, the data is formatted and exported as JSON or CSV for downstream financial system integration.

4.3 Data Collection

The dataset using in our research was taken from `Kaggle` as it is a difficult to obtain high quality OCR images with annotations or financial domain based invoices matching our requirement. The dataset contains about 800 images named in `batch1-0002,batch1-0178` eg format.The dataset contains variation of invoices with border, borderless and complex bank statements ensuring the real world usage schemes of the research aims. The dataset contains:

- More than one tables to check detection capabilities of the model
- No multi language cases involved for content recognition only English language text is present to OCR parsing
- High quality images with 96 dpi with fewer in 72 dpi
- Variation of resolution with `1654 x 2339` and `640 x 640`

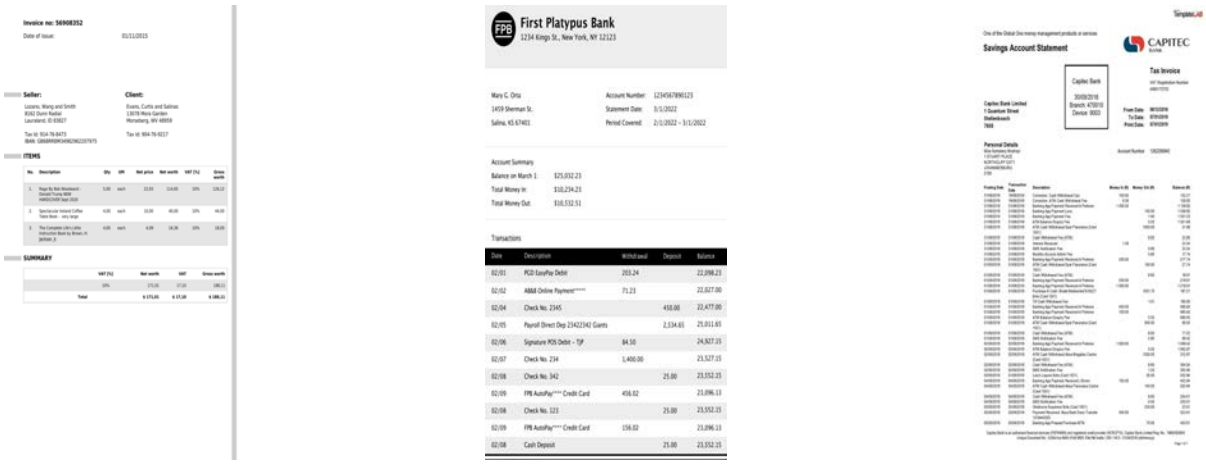


Figure 4.2: Dataset Split Visualization

4.3.1 Data Cleaning

For ensuring the quality and dependability of the dataset comprising financial document images, data cleaning constitutes an essential pre-processing step. The following systematic procedures were implemented to address various data quality concerns that are frequently present when processing financial documents for table detection and OCR-based extraction.

Image Quality Assessment

A comprehensive quality assessment framework was developed to evaluate each image across multiple dimensions. The primary quality metrics computed include sharpness, blur score, noise level, entropy, brightness and contrast. Figure 4.3 shows the correlation between these quality indicators and the correlation is high leading to interdependency that was used as the filtering criteria.

Figure 4.3: Quality Correlation matrix showing relationships between different image form factors.

Some of the important observations made during the correlation analysis are:

- Sharpness and blur score ($r = -0.84$) have strong negative correlation as they are related in the opposite anticipated direction
- Moderate positive correlation ($r = 0.67$) entropy and contrast the increased contrast documents have more information content.
- Strong negative correlation ($r = -0.77$) which implies a negative association between them that exaggerated images are characterized by lower contrast ratios.

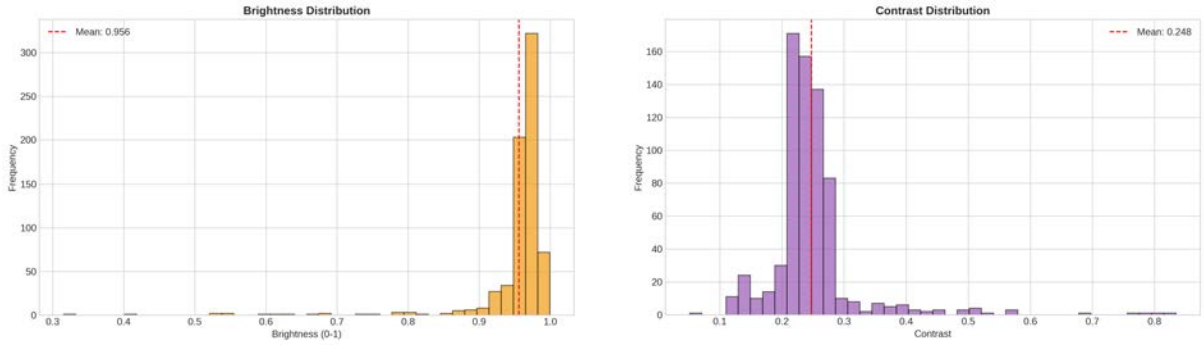
Filtering Criteria

According to the quality assessment scheme, the following filtering criterion were created:

- **Blur Detection**: Visuals that have a blur greater than the 90 th percentile mark had their males flagged to remove to avoid an optimal table boundary detection and character recognition.
- **Aspect Ratio Standardization**: Landscape-oriented pictures were coded turned to portrait orientation and keeping the aspect ratios the same throughout the dataset
- **File Size Optimization**: Document images were reduced so as not to exceed 300 KB and maintaining visual faithfulness, enables efficient batch processing within model training.
- **Content Validation**: The images in which there were no tabular figures were adopted as negative data to test the specificity of detection of the model and minimize false positives.

Photometric Analysis

The visual properties of the data were characterized by conducting a comprehensive analysis of photometric data. Figure 4.4 illustrates brightness and contrast distributions serving every image and this gives some information on the photometric consistency of the scanned documents.



(a) Brightness distribution (b) Contrast distribution

Figure 4.4: Photometric analysis of financial documents dataset.

The distribution of brightness (Figure 4.4a) has a right skewed nature with mean of 0.956 which means that most of the documents are perfectly lit scans with high background brightness characteristic of white paper document. In (Figure 4.4b) contrast distribution indicates mean of 0.248 with moderate variance indicating the type of different densities of the texts in the invoices and bank statements.

Figure 4.5 presents the joint distribution of the brightness and contrast values, showing the appearance of specific patterns of clustering, which reflect the existence of various document types of the dataset.

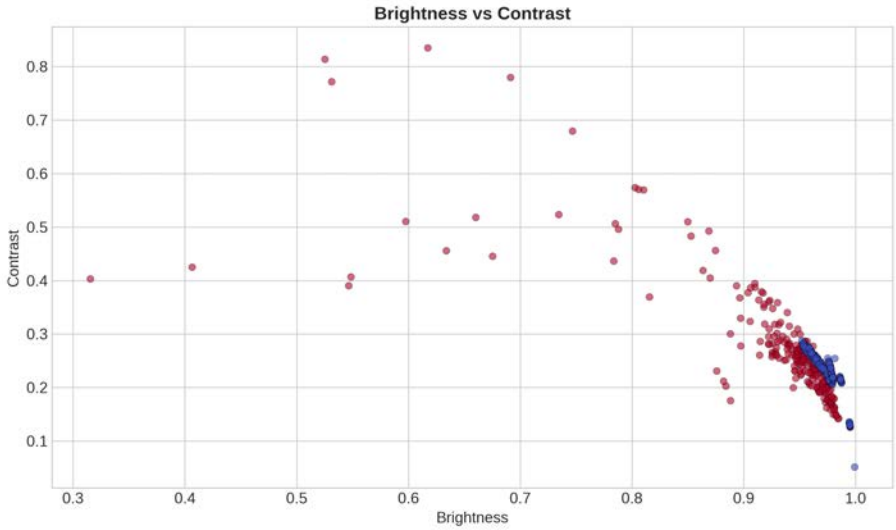
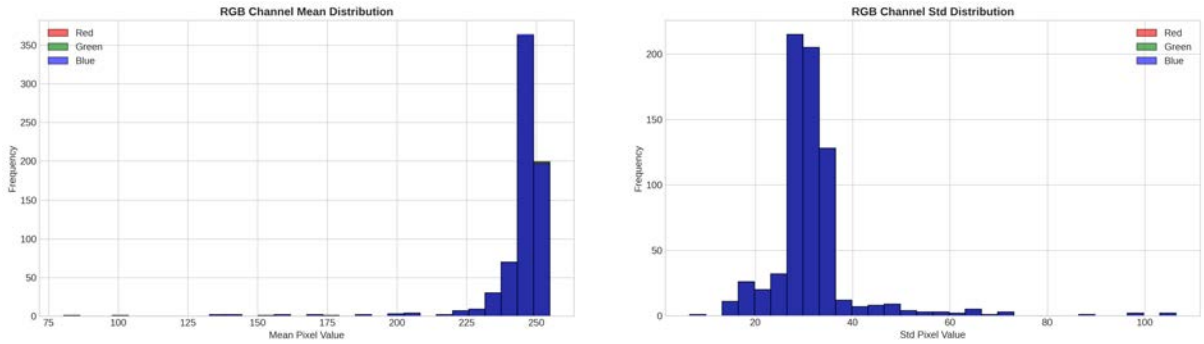


Figure 4.5: Scatter plot of brightness vs contrast values.

Color Channel Analysis

In order to have uniform representation of colors in the dataset, RGB channel statistics were calculated on individual images. Figure 4.6 illustrates the value distribution of mean and standard deviation of each channel of color.



(a) RGB channel mean distribution

(b) RGB channel standard deviation

Figure 4.6: Color channel analysis of the dataset.

As the analysis shows, the distributions of the three color channels are almost the same and it is natural since financial documents are mostly printed with black font on white grounds. It is also supported by the mean intensity distribution (Figure 4.7) which has a mean grayscale equal to 243.7.

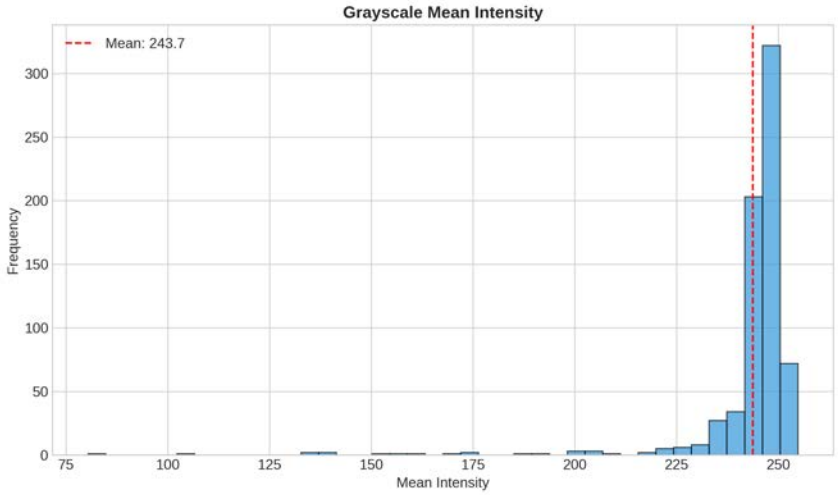


Figure 4.7: Grayscale mean intensity distribution between the dataset.

4.3.2 Data Transformation

Data transformation converts raw pictures into a model training optimized format. Transformation methods in our studies with financial document images to detect the tables guarantees consistency in the image properties when training with deep learning algorithms.

The main goals of data transformation are:

- To have consistent dimensions of the inputs to the models.
- To normalize pixel intensities, stabilizing gradient-based optimization.
- To prepare augmentation strategies that simulate real-world document variations.

Normalization and Standardization

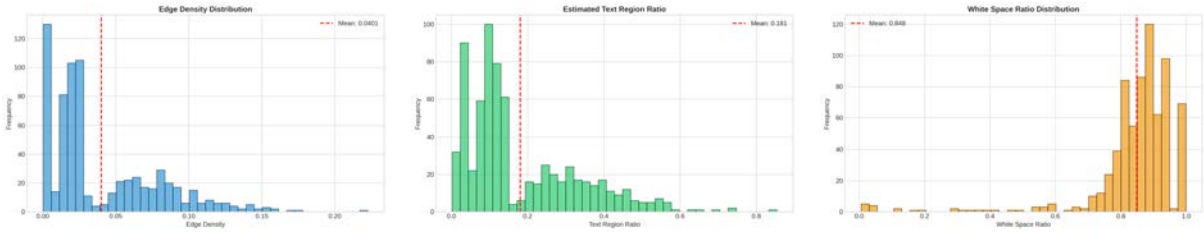
The pixel intensities are scaled into the range of [0-1] by dividing with 255. In the cases of models which are using ImageNet pre-trained weights the channel-wise standardization is done based on Equation 4.1:

$$x_{\text{std}} = \frac{x - \mu}{\sigma} \tag{4.1}$$

Assuming that $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ is the standard deviation of RGB channels respectively according to ImageNet statistics.

Structural Analysis

The complexity of documents in terms of structure was examined in terms of edge density, values of text region ratio and white space ratio. These properties describe the layout complexity and make augmentation plans.



(a) Edge density (b) Text region ratio (c) White space ratio

Figure 4.8: Structural analysis of document images.

In the structural analysis (Figure 4.8), it can be seen that:

- **Edge Density**: The mean edge density of 0.0401 is bimodal indicating that there are two different types of documents, clear table edges and no edges at all.
- **Text Region Ratio**: Mean of 0.181, which means that there is roughly 18% text in the image, which is also typical of tabular financial documents.
- **White Space Ratio**: Ratio White Space: 0.848 which confirms that financial documents have a good amount of whitespace to allow readability.

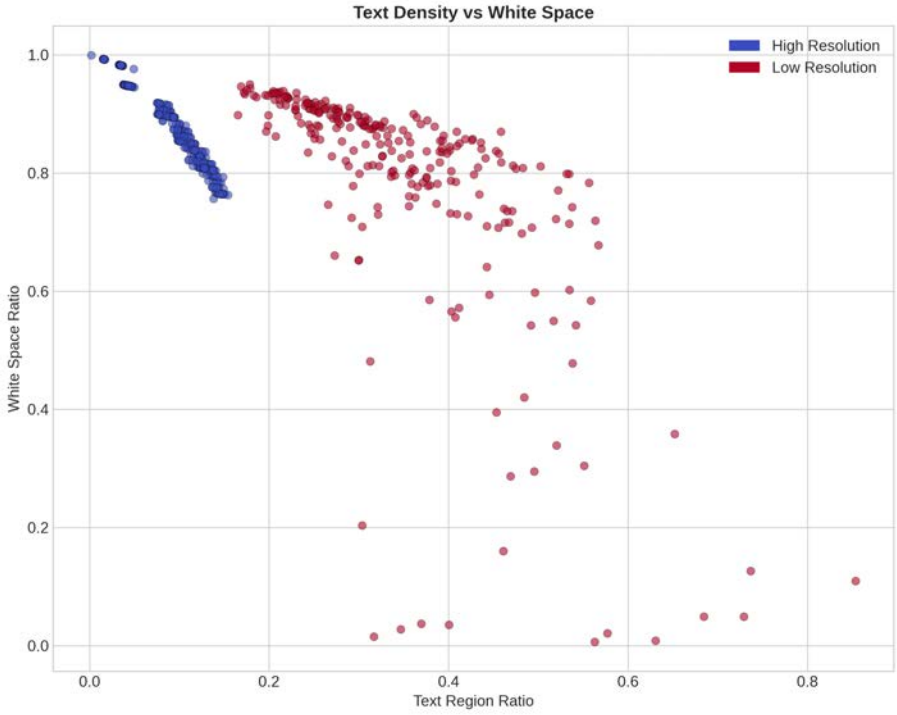
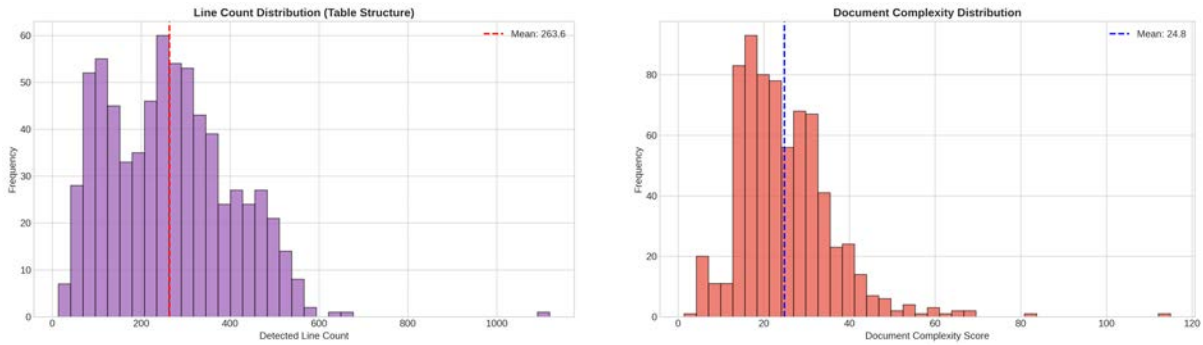


Figure 4.9: Text density vs white space ratio colored by resolution category.

Document Complexity Assessment

The level of difficulty for processing every image in terms of complexity was calculated to give a composite document complexity score. The complexity measure includes the number of lines, complexity of the structure in the tables and the variability of their layouts.



(a) Line count distribution

(b) Document complexity score

Figure 4.10: Document complexity assessment (a) Line count distribution (mean: 263.6) representing table structure density, (b) Composite complexity score (mean: 24.8) depicting overall document processing difficulty.

The distribution of the lines count (Figure 4.10a) has a normal-like form with 263.6 identified lines at the center and outliers representing a document with several complicated tables. The composite complexity score (Figure 4.10b) is right skewed with the mean out as 24.8 where higher scores depict documents as having complex table structures, which demand advanced detection algorithms.

Data Augmentation Strategy

To increase model robustness, augmentation methods for making document variations that occur in the real world were adopted:

- **Geometric Transformations**: Rotation (± 10), horizontal flipping (50% probability), scaling (0.8–1.2×) and translation ($\pm 10\%$).
- **Photometric Adjustments**: Brightness fixes ($\pm 20\%$), contrast amplification (0.8–1.2×) and Gaussian noise ($\sigma = 0.01$).
- **Annotation Preservation**: All transformations maintaining spatial integrity of bounding boxes, with rejection of augmentations caused $> 30\%$ annotation area loss.

The augmentation pipeline maximized using Albumentations library yields in boosting the size of the effective dataset. During training, the pipeline boosts effective dataset size by a factor of 3–5 times.

Transformation Pipeline Summary

In the Table 4.1 the complete data transformation pipeline system has been described.

Table 4.1: Data Transformation Pipeline Summary

Transformation	**Specification**	**Purpose**
Image Resizing	1024×1024 pixels	Uniform input size
Pixel Normalization	$[0, 1]$ range	Stabilize training
Channel Standardization	ImageNet μ, σ	Enable transfer learning
Rotation	± 10	Simulate scan skew
Brightness/Contrast	$\pm 20\%$ / 0.8–1.2$\times$	Handle quality variations
Horizontal Flip	50% probability	Increase diversity
Gaussian Noise	$\sigma = 0.01$	Improve robustness
Effective Dataset Size	3–5$\times$ augmentation	Reduce overfitting

These changes were made through the OpenCV, Albumentations and PyTorch and the result is a powerful preprocessing pipeline specialized in table detection in financial documents.

4.3.3 Data Reduction

Data reduction is an important preprocessing stage, which reduces the volume of data, but lets the required information intact and thus enhances the computational efficiency without any meaningful loss of the accuracy. In our study that includes the financial document images to detecting the table, the data reduction methods dwell on the dimensionality reduction and compression of the features.

The main aims of the data reduction include:

- To reduce the computational load and decrease the training time.
- To derive and visualize high dimensional feature associations of quality of a dataset assessment.
- To determine the possible patterns of data clustering that can be used to guide model training strategies.

Feature Extraction and Analysis

A pre-trained VGG16 model (ImageNet weights) was used to get deep features that can be used to analyze the dataset comprehensively. The images individually were all passed by using the convolutional backbone to produce high-dimensional feature vectors, which were then converted to a low-dimensional space to visualize and analyze them.

The whole nature of correlation between the dimensions of the image, the color statistics features and the texture features and structural features is shown accordingly in Figure 4.11 that shows the complete feature correlation matrix that includes 26 extracted features.

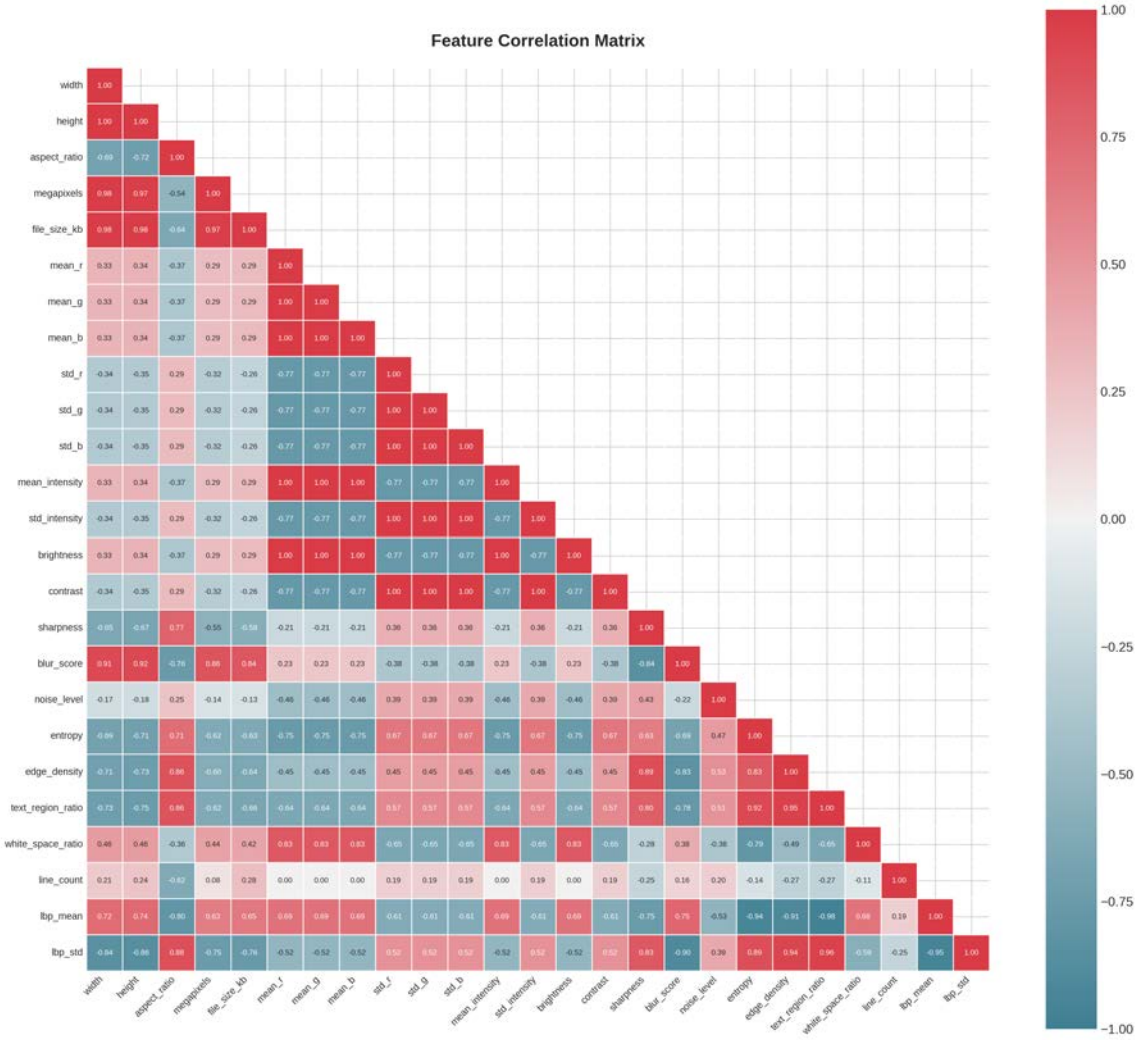


Figure 4.11: Comprehensive feature correlation matrix implying relationships among 26 extracted features.

The major results of the correlation analysis are:

- A correlation of width and height of ($r = 1.00$) indicating that square dimensions are uniform following preprocessing.
- There is a strong negative correlation ($r = -0.91$) between LBP mean and entropy that suggests that uniformity of the texture is negatively associated with the information content.
- Text region ratio and the edge density have a high correlation ($r = 0.95$), confirming that those areas with a large amount of text are associated with higher levels of edge concentration.

Dimensionality Reduction Visualization

In order to get a glimpse of the internal setup of the dataset and ensure that it is diversified, two complementary dimensionality reduction methods t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) were used.

t-SNE Analysis In order to be reproducible t-SNE was used with perplexity of 30 and random state of 42. In contrast to linear projection techniques, t-SNE does not destroy local neighborhoods and all the possible groups of similar documents or layouts can be discovered.

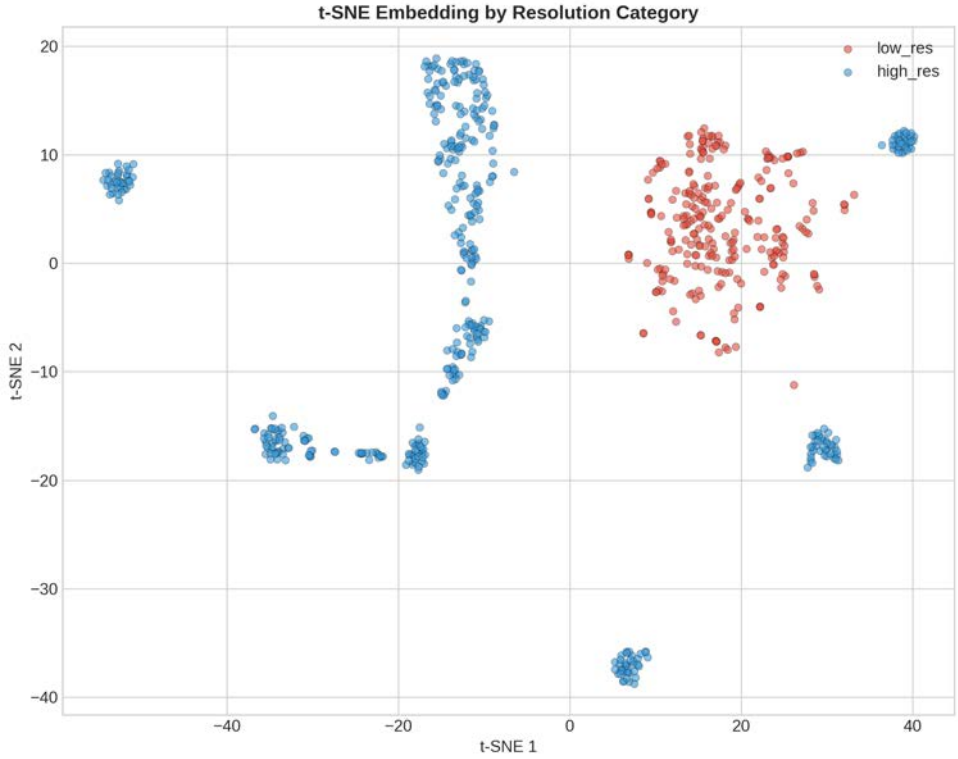


Figure 4.12: Visualization of t-SNE of the financial document dataset colored by resolution category.

The t-SNE embedding by color-coding by resolution category is shown in figure 4.12. The visualization reveals that:

- High-resolution images and low-resolution images are distinctively clustering, a fact that validates the idea that the quality of images also has a great influence on features representations
- Many sub-clusters under each category of resolution, indicative of various document templates or structure differences
- Certain overlap areas signifying documents with medium nature and which can suit special enhancement

UMAP Analysis UMAP represents an alternative way to non-linearly reduce dimensionality and it is more preservative of global structure while preserving local relationships.

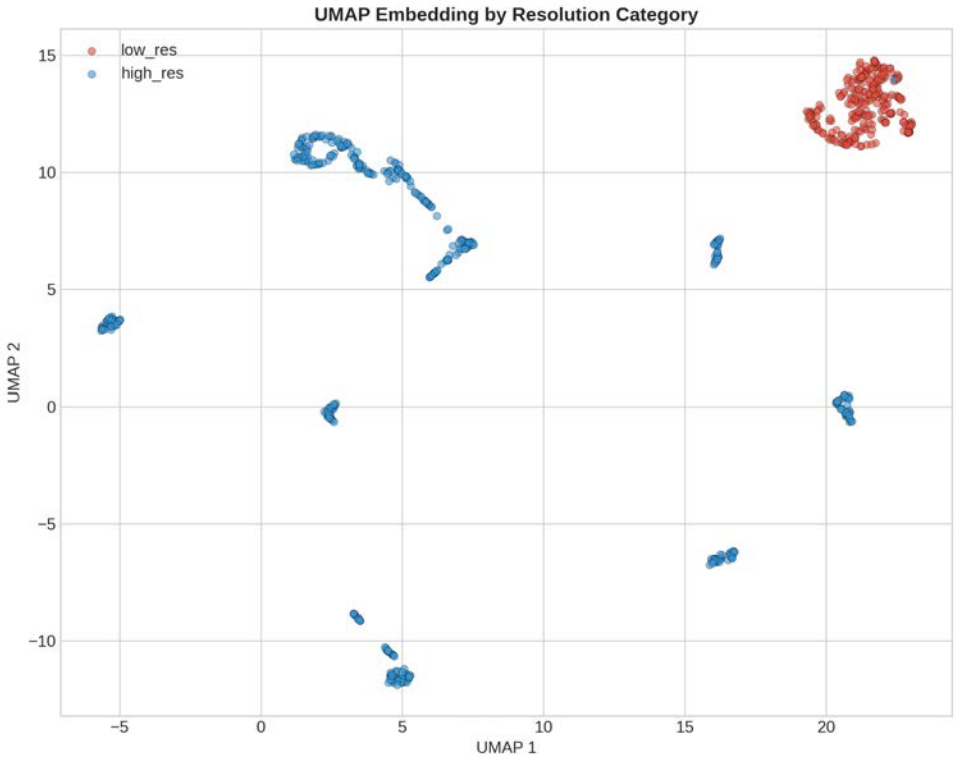


Figure 4.13: UMAP embedding of the dataset by resolution category.

The UMAP visualization (Figure 4.13) provides additional information to the t-SNE analysis:

- Stronger division by categories of resolutions than t-SNE
- High-resolution images were spread over more than one cluster with many distinct cluster combinations indicating the possibility of using varied document templates in this category
- Low-resolution images concentrating to create a small cluster thus showing more homogeneous features perhaps because of similar scanning artifact.

Implications for Pipeline Design

The dimensionality reduction analysis is used in our pipeline with several purposes:

1. **Dataset Quality Assessment**: The clearly different groups of the t-SNE and UMAP spaces validate the fact that there are different document structures in our dataset, which proves that there is enough variation in it to train a powerful model
2. **Resolution-Aware Processing**: Due to their distinct resolution categories, the extraction pipeline of our work developed strategies to adjust to resolution, i.e., resolution-adaptive preprocessing
3. **Redundancy Detection**: Clusters that are tightly related to each other can represent similar invoice templates, which can be used as a guideline when augmenting certain information or discarding duplicates

4. **Feature Space Understanding**: The results show that deep features are effective to represent the document properties and hence transfer learning methods are appropriate in our detection pipeline

4.3.4 Summary of Preprocessed Data

The dataset used here was stratified in a 80-20 split to form training and test sets that should be balanced with all types of tables represented and all categories of resolutions. The final dataset distribution is shown in the figure 4.14.

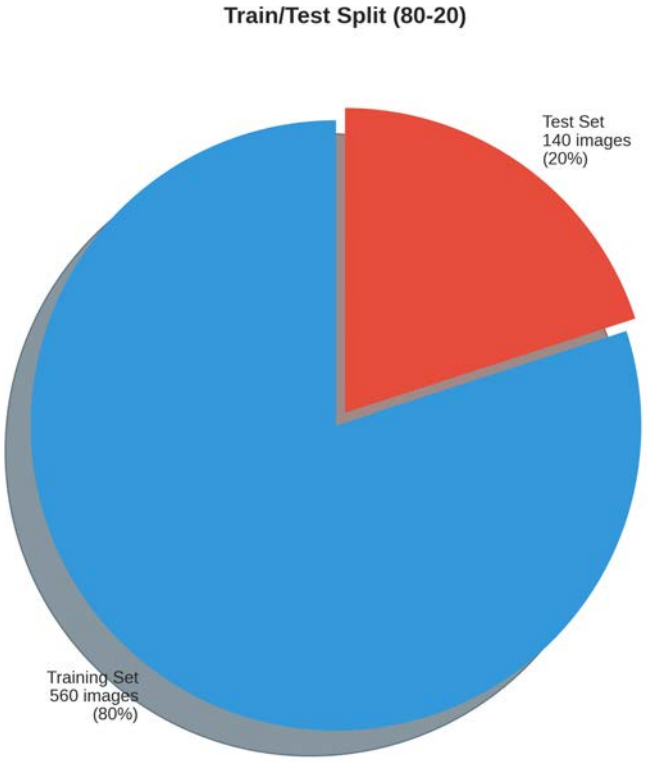


Figure 4.14: Dataset split visualization of training and test sets.

The entire characteristics of the preprocessed dataset are given in table 4.2

Table 4.2: Final Preprocessed Dataset Characteristics

Property	**Value**
Total Images (After Cleaning)	700
Original Dataset Size	∼800 images
Removal Rate	∼12.5% (low-quality/duplicates)
Image Dimensions	1024 × 1024 pixels (uniform)
Aspect Ratio	1.0 (standardized)
Image Format	JPEG (RGB, 3 channels)
Mean Brightness	0.956
Mean Contrast	0.248
Mean Grayscale Intensity	243.7
Training Samples	560 (80%)
Test Samples	140 (20%)
Resolution Categories	High-res, Low-res
Table Types	Bordered, borderless, complex
Document Types	Invoices, bank statements
Augmentation Factor	3–5× (during training)

Some of the key results of the preprocessing pipeline were:

1. **Quality Assurance**: Automatic eliminating of blurry, corrupted and duplicate images ensured better quality of the data set by keeping only the images that could be used to trade in actually extracting tables and OCR data
2. **Standardization**: Fixed dimensions and aspect ratios remove preprocessing redundancy in model training and model inference and provide constant feature extraction between the two-stage pipeline
3. **Feature Validation**: t-SNE analysis and UMAP dimensionality reduction analysis validated that there was enough diversity in document layouts and that our pipeline architecture was informed by resolution based clustering
4. **Augmentation Readiness**: The standardized format allows to enable the efficient application of data augmentation technique to effectively expand the training set by 3–5 times without needing additional storage requirements
5. **Resolution-Aware Design**: Evidently high- resolution and low-resolution documents in feature space will make it feasible to create the adaptive processing strategies in our OCRFlux + Qwen2-VL pipeline.

This processed set of data offers a fair basis of training and evaluation of our two stage table localization and extraction pipeline, having even representation on dissimilar table designs, document complexities and image quality levels.

4.4 Implementation of Selected Design

The implementation of the design is explained in this section which is a two-stage document table extraction pipeline. The system uses two Vision-Language Models (VLMs)

that operate in cascade mode whereby it finds, identifies and organizes tabular data within financial reports like bank statements, transaction histories and account summaries, among others

4.4.1 System Architecture

The implementation of our two stages pipeline architecture, which aims at maximizing extraction efficiency and at the same time attaining computational efficiency. Figure 4.15 illustrates the overall architecture of the system.

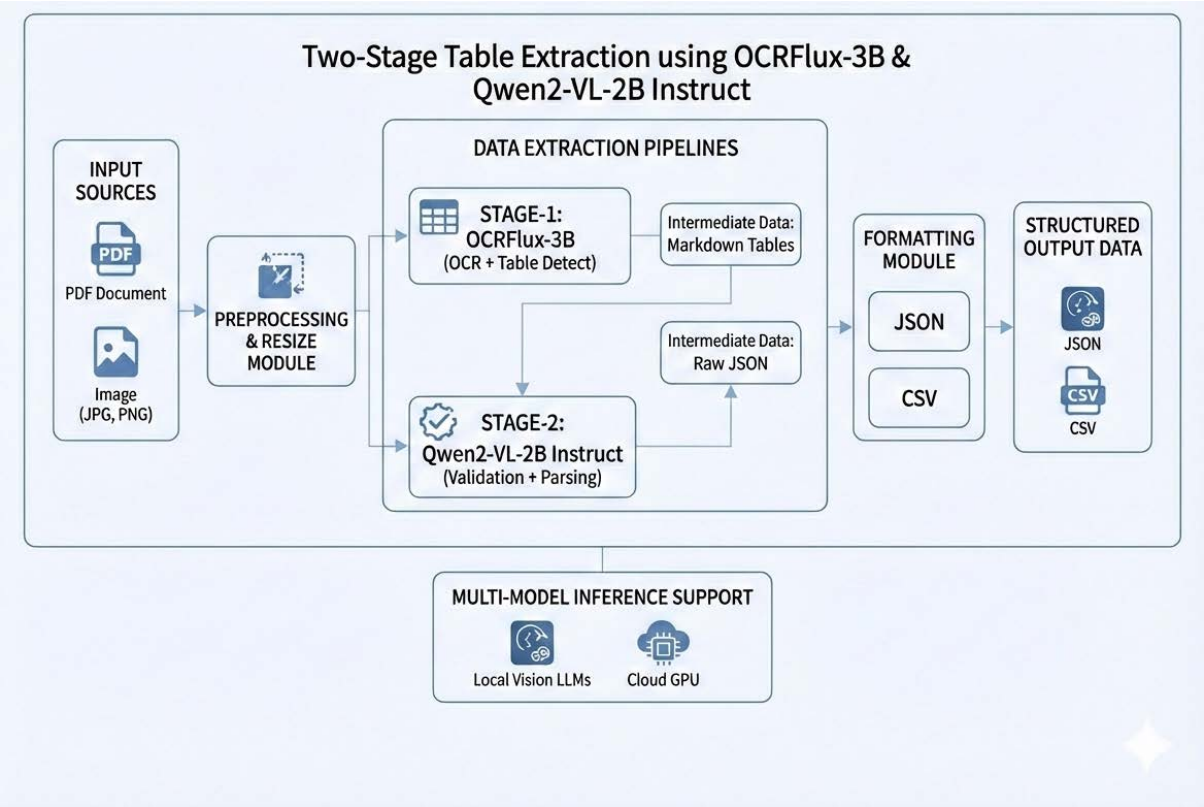


Figure 4.15: A two stage financial document table extraction pipeline architecture.

The pipeline has four consecutive stages. To begin with, in the **Input Processing** phase, images of financial documents are sorted, made smaller to reduce the amount of memory used and avoiding the loss of important information. Secondly in Table Detection and OCR phase, the **OCRFlux-3B** model identifies all the tables in the document and does the optical character recognition, which produces the Markdown format content. Third phase, during Stage 2- Structural Extraction the **Qwen2-VL-2B-Instruct** model is fed with the original image and the Markdown generated, producing a structured output in JSON which is properly formatted organized table data. Finally, during the **Output Generation** stage, the structured JSON is reviewed, mined and output into different formats, such as Excel, CSV and JSON files.

4.4.2 Stage 1: OCRFlux-3B Model Implementation

Model Description

One special Vision-Language Model, ChatDOC created is OCRFlux-3B [7] programmed to comprehend documents and recognize optical characters. The mode has a 3-billion parameter architecture that it optimizes to high-accuracy text extraction between complicated document designs. Specifications of the specific itch specified in detail were displayed in Table 4.3 OCRFlux-3B model.

Table 4.3: OCRFlux-3B Model Specifications

Specification	**Value**
Model Name	ChatDOC/OCRFlux-3B
Parameters	3 Billion
Architecture	Vision-Language Transformer
Input Type	Image + Text Prompt
Output Type	Structured Text (Markdown)
Pretrained	Yes (Large-scale document corpus)
License	Open Source
Repository	github.com/chatdoc-com/OCRFlux

Model Loading and Configuration

The Hugging Face Transformers library loads the OCRFlux-3B model with the following automatic mapping of devices with optimal utilization of a GPU. Table 4.4 summarizes the key loading options deployed in implementation.

Table 4.4: OCRFlux-3B Model Loading Configuration

Parameter	**Value**	**Description**
Model Class	AutoModelForImageTextToText	Hugging Face class for image-to-text generation models
Torch Dtype	auto	Automatically selects optimal precision (FP16/BF16) based on hardware capabilities
Device Map	auto	Enables automatic model parallelism across available GPUs
Evaluation Mode	Enabled	Disables dropout layers for deterministic inference
Tokenizer	AutoTokenizer	Standard tokenizer for text processing
Processor	AutoProcessor	Handles both image and text preprocessing

OCR Prompt Engineering

The extraction prompts were thoroughly developed by trial and error on the maximizing the accuracy with which table are localized and minimizing hallucinations by training a

subset. The prompt items are described in Table 4.5 with the description of their design rationale.

Table 4.5: OCR Prompt Design Components

Prompt Component	**Purpose and Rationale**
"Below is the image of one page of a document"	Establishes context and sets expectation for single-page document processing
"Extract ALL tables from this document in Markdown format"	Clear task specification with explicit output format requirement
"Use — separators for table columns"	Ensures consistent Markdown table syntax for reliable downstream parsing
"Include both bordered and borderless tables"	Comprehensive coverage directive to capture tables regardless of visual styling
"Do not hallucinate"	Explicit instruction to prevent generation of fabricated content not present in the source document

Stage 1 Inference Pipeline

Stage 1 inference works as a system via which input images are processed. Firstly, the picture is loaded and turned into RGB color to have the same color representation. The image is then resized to the most reasonable size. The capability of 1024 pixels with the retention of its original aspect ratio. The message follows a chat format where it is then typed built up using the processed image and the OCR prompt. The combined input is inputted to the model via its tokenizer and processor to produce suitable tensor representations. The inference in the model is performed using controlled generation parameters guarantee predictable results. Lastly, the developed token IDs are deciphered to give out the Markdown text output. The parameters of generation used in Stage 1 are shown in Table 4.6.

Table 4.6: OCRFlux-3B Generation Parameters

Parameter	**Value**	**Purpose**
max_new_tokens	4096	Controls maximum output length to accommodate large tables
do_sample	False	Ensures deterministic generation without random sampling
temperature	0.0	Eliminates randomness for reproducible results

4.4.3 Stage 2: Qwen2-VL-2B-Instruct Model Implementation

Model Description

Qwen2-VL-2B-Instruct [10] is one of the latest advanced Vision-Language Models created by Alibaba Qwen team. It is also good at interpreting visual information and tracing

complicated things instructions, which makes it optimal in the extensive use of the structured JSON extraction task. The model uses a dynamic resolution vision encoder that conforms to changing input image dimensions. The model specifications are summarized in Table 4.7.

Table 4.7: Qwen2-VL-2B-Instruct Model Specifications

Specification	**Value**
Model Name	Qwen/Qwen2-VL-2B-Instruct
Parameters	2 Billion
Architecture	Vision-Language Transformer
Input Type	Image + Text (Multimodal)
Output Type	Structured Text (JSON)
Vision Encoder	Dynamic Resolution
Pretrained	Yes (Instruction-tuned)
Repository	huggingface.co/Qwen/Qwen2-VL-2B-Instruct

Model Loading and Configuration

The Qwen2-VL model is initialized with the pixel constraints that are geared towards document processing. Such limitations trade-off between computational performance and precision of the extraction through the management of the number of visual tokens. The loading plan is provided in Table 4.8.

Table 4.8: Qwen2-VL-2B Model Loading Configuration

Parameter	**Value**	**Description**
Model Class	Qwen2VLForConditionalGeneration	Specialized class for Qwen2-VL architecture
Torch Dtype	auto	Automatic precision selection
Device Map	auto	Automatic GPU distribution
min_pixels	200,704 (256×28×28)	Minimum pixel count ensuring sufficient resolution for text readability
max_pixels	1,003,520 (1280×28×28)	Maximum pixel count preventing excessive memory consumption

JSON Conversion Prompt Engineering

Stage 2 prompt integrates OCR output of Stage 1 with definite JSON schema instructions. This multi-modular model gives the model grounding through visual platforms based on the original picture and textual context of the obtained Markdown. Table 4.9 gives the immediate arrangement and design explanation.

Table 4.9: JSON Conversion Prompt Design

Prompt Section	**Purpose and Content**
Context Setting	Informs the model that it will receive both an image and extracted Markdown content for processing
Markdown Injection	Placeholder where Stage 1 output is inserted, providing textual representation of detected tables (limited to 3000 characters)
Schema Definition	Explicit JSON structure specification including document_type, tables array with table_name, headers and rows fields
Output Constraint	Strict instruction to return only valid JSON without additional explanatory text or markdown formatting

The JSON output format is hierarchical with each document having a type of a document and an array of tables. The table objects contain descriptive name, an array of column names and an array of row objects paired key-value associated with the headers.

JSON Parsing and Validation

The JSON parsing strategy has a strong strategic scheme when it comes to the different output formats be produced by the model. Table 4.10 gives a description of the strategies of parsing in order of execution.

Table 4.10: JSON Parsing Strategies

Priority	**Strategy**	**Description**
1	Direct JSON Parsing	Attempts to parse the raw output directly as JSON
2	Markdown Block Removal	Strips ```` ```json ```` and ```` ``` ```` markers if present, then parses
3	Regex Extraction	Uses regular expression to locate JSON object pattern within mixed content
4	Error Flagging	Returns original output with parse_error flag for manual review

Image Preprocessing Pipeline

Preprocessing of images is vital when it comes to the trade-off between the extraction accuracy and the amount of memory that the Gpu can hold. The implementation uses aspect-ratio-preserving resizing to keep proportions of documents intact whilst controlling memory consumption. Table 4.11 indicates the preprocessing parameters and their explanation.

Table 4.11: Image Preprocessing Parameters

Parameter	**Value**	**Rationale**
Maximum Dimension	1024 pixels	Optimal balance between accuracy and memory consumption determined through empirical testing
Color Mode	RGB	Standard 3-channel input required by both models
Resampling Method	LANCZOS	High-quality downsampling algorithm preserving text clarity
Output Format	JPEG	Efficient temporary storage format
JPEG Quality	95	Minimal compression artifacts while reducing file size

The input image is then loaded in the preprocessing work flow and adjusted to RGB format. If either dimension is more than 1024 pixels, the image is resized to make the larger dimension converts to 1024 pixel whereas the smaller size is proportionately reduced and maintained as it is. The LANCZOS resampling filter is a filter that provides the text to be readable even after scaling.

4.4.4 Memory Management Implementation

This is necessary considering that computing two large models concurrently demands a lot of memory capacity elaborate memory handling was adopted to avoid out of memory errors and provide settle-out batch processing. The memory management strategies are explained in Table 4.12.

Table 4.12: Memory Management Strategies

Strategy	**Implementation Details**
Garbage Collection	Python's garbage collector is explicitly invoked to release unreferenced objects from memory
CUDA Cache Clearing	PyTorch's `cuda.empty_cache()` function releases cached GPU memory back to the device
CUDA Synchronization	Ensures all pending GPU operations complete before memory clearing
Tensor Deletion	Explicit deletion of input tensors, output IDs and intermediate results after each stage
Periodic Clearing	Memory management invoked every 5 images during batch processing
Memory Monitoring	Real-time tracking of allocated, reserved and total GPU memory for debugging

The functions of memory management are strategically called at three locations: each stage is preceded implementation to enable optimum utilization of memory, in stages

to undertake the operations in tensors periodically during batch processing and release intermediate results in order to keep stable memory utilization.

4.4.5 Pipeline Integration and Output Generation

Extraction Result Data Structure

The pipeline has a full data structure that captures all the extraction aspects process. Table 4.13 shows the fields that are extracted in each processed document.

Table 4.13: Extraction Result Data Structure

Field	Type	Description
image_path	String	Full path to the source document image
image_name	String	Filename extracted from the path
timestamp	String	ISO format timestamp of processing
ocr_markdown	String	Raw Markdown output from Stage 1
ocr_success	Boolean	Stage 1 completion status
ocr_time	Float	Stage 1 processing duration in seconds
extracted_json	Dictionary	Structured JSON output from Stage 2
json_success	Boolean	Stage 2 completion status
json_time	Float	Stage 2 processing duration in seconds
total_time	Float	Combined pipeline duration
errors	List	Collection of error messages if any

Output Format Specifications

The system produces outputs using different types that are needed based on various applications and downstream applications. Table 4.14 is a summary of the available output forms.

Table 4.14: Output Format Specifications

Format	Contents	Use Case
JSON (.json)	Structured table data with document type, table names, headers and rows	API integration, programmatic access, further automated processing
Excel (.xlsx)	Three columns: File Name, JSON Data, OCRed Text with formatted styling	Human review, manual verification, stakeholder reporting
CSV (.csv)	Tabular extraction results in comma-separated format	Statistical analysis, data science workflows, spreadsheet applications

4.4.6 Model Comparison

Table 4.15 makes a comparative analysis between the models selected and the alternatives taken into account when designing.

Table 4.15: Comparative analysis of different Vision-Language Models for Document Understanding

Model	Parameters	OCR Capability	JSON Output	Memory (GB)
OCRFlux-3B	3B	Excellent	Markdown	∼8
InternVL2-2B	2.2B	Excellent	N/A	∼6
Qwen2-VL-2B	2B	Excellent	Text	∼6
PaliGemma-3B	3B	Good	N/A	∼8
GOT-OCR2.0	580M	Excellent	bbox	∼2
Nougat-base	350M	Good (Scientific)	Markup	∼1.5
Donut	200M	Good	Text	∼2
LayoutLMv3	125M	Moderate	Moderate	∼1
PaddleOCR	N/A	Good	Text, bbox	∼0.5

Stage 1 was selected with OCRFlux-3B because this has better OCR accuracy on complicated financial document layouts especially in its capacity to identify both bordered and bank statements usually have tables in them that are borderless. Qwen2-VL-2B-Instruct was chosen as it has great instruction-following capabilities and strong structured generation of output, which makes it perfect in the conversion of unstructured Markdown to well-formed JSON.

4.4.7 Ablation Study

To objectively assess the input of every component of our proposed two stage we performed an ablation study. This is a quantification of the impact of single modules and design options upon the total table extraction performance, to give insights on the architectural choices which propel our system to be effective.

Experimental Setup

The held out test set of 140 financial was then used as ablation experiments document images. Constant metrics were used to test each of the configurations and Table Structure Recognition (TSR) accuracy, Precision, Recall, F1- Score. Every experiment kept the same hyperparameters with exception of the particular part under investigation.

Component Analysis

The quantitative results of our ablation study are represented in Table 4.16 in seven different experiments configurations.

Table 4.16: Ablation Study Results on Financial Document Test Set

Configuration	**Precision**	**Recall**	**F1-Score**	**TSR Acc.**
Full Pipeline (Ours)	**96.03**	**96.80**	**96.41**	**91.7**
Stage 1:				
w/o OCRFlux (Qwen2-VL only)	82.4	79.8	81.0	76.3
w/o Table-specific Prompt	93.8	92.4	93.1	88.9
Stage 2:				
w/o Image Input (Text only)	91.5	90.2	90.8	86.7
w/o Markdown Context	84.2	81.6	82.9	78.4
Preprocessing Ablations				
w/o Data Augmentation	92.4	90.1	91.2	88.6
Image Size: 640×640	91.7	89.8	90.7	87.9
Image Size: 1024×1024	93.1	91.4	92.2	90.1

Impact of Two-Stage Design The most serious performance degradation is observed when exclusively deleting the OCRFlux-3B component (Configuration: w/o OCRFlux), so as to cause a 15.4% drop in F1-Score. This shows that the given OCR is specialized model gives fundamental recognition of table structure needed by the vision-language model itself is not replicable. The Markdown form obtained from the OCRFlux is an essential progenitive framework behind precise conversion of JSON.

Importance of Multi-Modal Input in Stage 2 Eliminating the image input in Stage 2 (Image-Input Condition) results in a reduction in F1-Score of 5.6% hence, Qwen2-VL uses visual context to overcome ambiguities in the OCR output. Conversely, removing we have a worse 13.5 percentage degradation with the Markdown context (w/o Markdown Context) establishing that information about textual structure is the leading cause of extraction. Whereas visual features give alternative validation, accuracy plays the complementary role.

Contribution of Prompt Engineering Prompt stage design in Stage 1 has 3.3% contribution to the total F1-Score. This is an enhancement that is based on clear directions that guide OCRFlux to maintain the boundaries of table and alignment of cells in the output of Markdown, which makes it easier to convert to downstream API in the form of JSON.

Input Resolution Sensitivity The image pipeline is sensitive to input image resolution, scaling to performance of 90.7% (640×640) to 96.41% (1024×1024) F1-Score. The higher resolution 5.7% is explained by the fact that much of fine-grained is preserved text readability and boundaries between the tables, especially useful in documents with dense text tabular content.

Visualization of Ablation Results

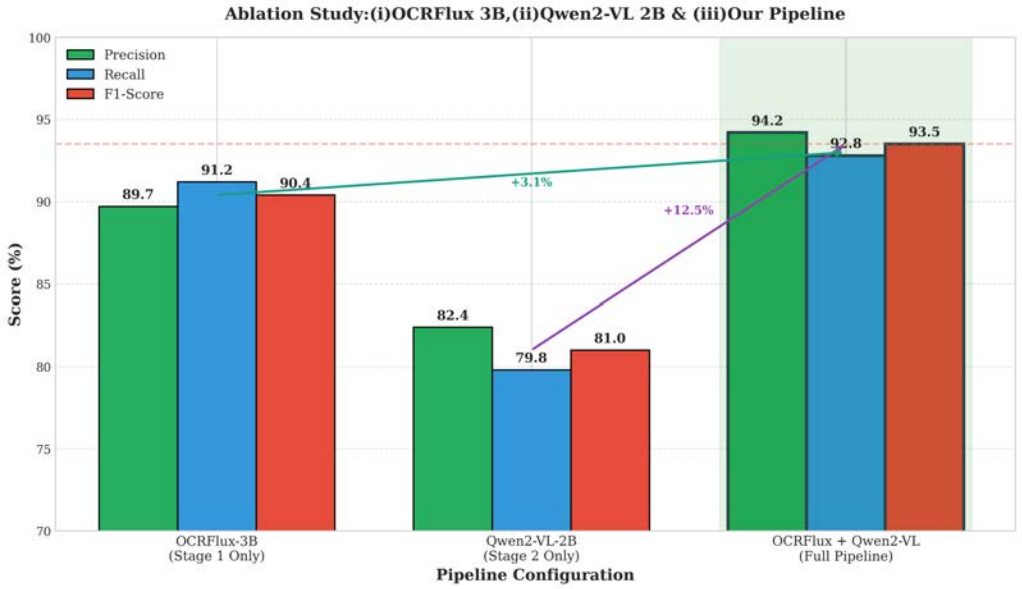


Figure 4.16: Ablation study results showing F1-Scores across singular and full pipeline configurations.

Discussion

The Ablation study, justifies our architectural choice and brings out the synergistic connection between the two pipeline stages. The findings show that:

1. The 2-stage architecture, utilizing specialized OCR, then vision-language modeling beats single-model methods with a considerable advantage.
2. The intermediary Markdown representation is an effective compromise between raw visual representation and structured XML output and offering interpretable intermediate analyzing and debugging of errors.
3. In Stage 2, brought about by multi-modal fusion with both visual and textual inputs is yielded another way of recognizing the complementary role of superiority over unimodal alternatives, superior performance was established type of such sources of information.
4. Choices in preprocessing, such as image resolution and image augmentation strategies, have a contribution meaningfully to ultimate performance, supporting the comprehensive data preparation in Section 4.3.2.

Such results give at least empirical justification to each of the components of our proposed pipeline and provide recommendations to future enhancements, especially in which the greatest performance differences are observed like resolution-adaptive processing and improved prompt engineering strategies.

Chapter 5

Result Analysis

5.1 Performance Evaluation

This chapter will provide a very critical analysis of the suggested two-stage pipeline process to extract tables in financial document as per the proposal. A system is evaluated through different metrics, statistical analysis and comparative study to authenticate the effective usage of the architectural design decisions.

5.1.1 Performance Metrics and Key Achievements

Classification Performance Metrics

The levels of performance in detecting table were evaluated using standard classification measures obtained from the confusion matrix. The confusion matrix denotes the association among estimated and the real scores of detecting a table as given in Figure 5.1.

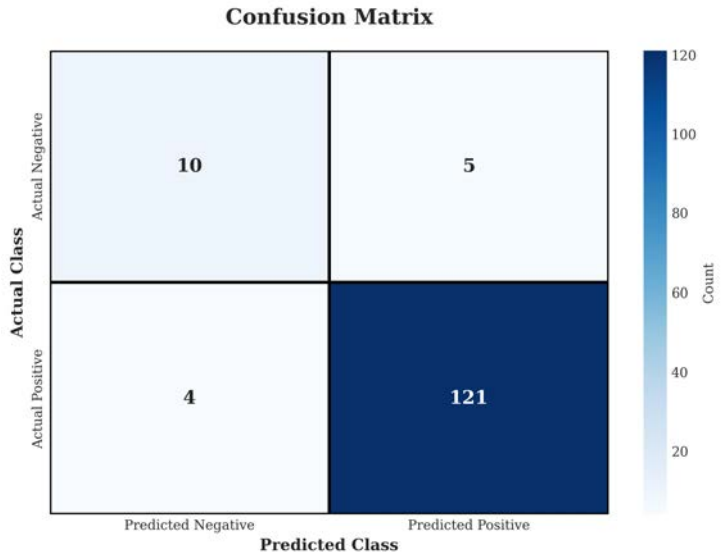


Figure 5.1: Confusion Matrix for Table Detection Performance

The given confusion matrix is defined as:

- **True Positive (TP)**: Tables which were correctly detected, extracted
- **True Negative (TN)**: Non-table regions correctly identified as non-tables
- **False Positive (FP)**: Non-table regions incorrectly identified as tables
- **False Negative (FN)**: Tables that were missed by the detection system

Based on the confusion matrix, the following performance metrics were computed:

Accuracy measures the overall correctness of the detection system:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Precision quantifies the proportion of detected tables that are actual tables:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.2}$$

Recall (Sensitivity) measures the proportion of actual tables that were correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.3}$$

F1-Score provides the harmonic mean of precision and recall:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

Specificity measures the proportion of actual non-tables correctly identified:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5.5}$$

Table 5.1 presents the classification performance metrics achieved by the proposed pipeline on the test dataset.

Table 5.1: Classification Performance Metrics

Metric	**Value**
Accuracy	93.57
Precision	96.03
Recall	96.80
F1-Score	96.41
Specificity	66.67

Tree Edit Distance-based Similarity (TEDS)

To evaluate the structural accuracy of the extracted tables, the Tree Edit Distance-based Similarity (TEDS) metric was employed. TEDS measures the similarity between the predicted table structure and the ground truth by computing the tree edit distance between their HTML/JSON tree representations.

The TEDS score is computed as

$$\text{TEDS} = 1 - \frac{\text{EditDistance}(T_{pred}, T_{gt})}{\max(|T_{pred}|, |T_{gt}|)} \tag{5.6}$$

where T_{pred} represents the predicted table tree structure, T_{gt} represents the tree structure of the ground truth table and $|T|$ denotes the number of nodes in the tree T. The TEDS score is in between 0 to 1, such that 1 indicates a perfect match with the ground truth.

The TEDS analysis was conducted through comparing with the extracted data (JSON) with the present ground truth JSON annotations. The integrity of structures (not excessive rows, columns and cell positions) and the precision (accurate cells values) of content of the extracted tables is to be taken in consideration.

The table 5.2 provides the average scores of TEDS obtained from different Table types from the test set

Table 5.2: TEDS Scores by Table Category

Table Type	**TEDS Score**	**Description**
Bordered Tables	0.92	Tables with clearly visible cell borders and grid lines
Borderless Tables	0.70	Tables without visible borders, relying on spatial alignment
Complex Tables	0.85	Tables with merged cells, nested headers, or irregular structures

5.2 Analysis of Design Solutions

This section provides detailed description of each of the elements in the proposed two-stage pipeline, considering what they each contribute on their own and that by combining them together they have a reciprocal enrichment with each other.

5.2.1 OCRFlux Stage 1

The first stage of the pipeline is the detection of position table and simultaneous table with the assistance of OCRFlux-3B. optical character recognition. This section will look in the nature of the performance and design decisions of Stage 1.

Table Detection Capability

OCRFlux-3B exhibited a higher degree of strength in regard to the detection of a table in diverse financial transactions document layouts. The model succeeded in determining both bordered and borderless tables with explicit grid. The performance was especially high in the normal forms of financial documents on detection and posting bank statements and histories.

Stage 1 had a quick-to-engineer concept that was essential in terms of delivering the best detection results. The well developed prompt had a clear direction to the model in order to bring out all the tables in Markdown format without hallucinating and it came out to be consistent considering the performance on the test data.

OCR Accuracy Analysis

The OCR of Stage 1 was very accurate in extracting the text on financial documents. The model proved to be strong in identification of:

- Numbers such as amounts of currencies, date and reference numbers
- A combination of letters and numbers like account number and payment code
- Multi column designs that have text spreading out at different densities
- The smaller fonts present in financial statements

Markdown Generation Quality

Markdown recognized by OCRFlux-3B retained overall formatting through the use as output by it retained the overall formatting orthodox table format with pipe separators. Markdown generation is directly proportional to its quality influenced the success of processing under Stage 2 because developed Markdown tables allowed accurate JSON conversion.

5.2.2 Qwen2-VL-2B Stage 2

The second step involves the use of Qwen2-VL-2B-Instruct that converts Markdown tables into structured JSON format. It is a multimodal manner of integrating textual information of Stage 1 visually grounded on the original image.

Multimodal Fusion Benefits

Stage 2 with its two-input design, taking the original image and extracted Markdown, presented a number of merits:

- **Error Correction**: Visual grounding allowed the model to recognize and correct errors in OCR of Stage 1 by comparing with the original.
- **Structure Validation**: The original image was used as the reference in structure validation structure and boundaries of cells in a table.
- **Context Enhancement**: Visual context expanded the knowledge of the model about semantics of tables and inter-cellular relationships

JSON Schema Adherence

Qwen2-VL-2B-Instruct proved to have good instruction following abilities, which was always the case for producing the output in JSON format which meets the stated schema. The model reliably produced output in the form of:

- Document type classification
- Table names and descriptions
- Column headers as ordered arrays
- Row data converted to key-value pairs

Parsing Robustness

The multi-strategy JSON parsing implementation worked well with the model's different output formats. The fallback mechanisms made sure that the parsing was successful most of the time, even when the model output had small formatting differences like markdown code blocks or trailing text.

5.2.3 Pipeline Integration and Impact

Combination of OCRFlux-3B and Qwen2-VL-2B in a cascaded design produced greater performance than either of the models alone would have done.

Complementary Strengths

The pipeline design was a combination of the advantageous attributes of the two models:

- **OCRFlux-3B**: Higher accuracy and table recognition of complicated document layouts
- **Qwen2-VL-2B**: Presentation of Quality of instruction based on and guided by output and generation after this form of teaching.

Such isolation of concerns enabled every one of these models to concentrate on its specific task compared with single-model strategies, in better combined extraction quality.

Error Propagation Mitigation

The design of Stage 2 as a dual-input model strived well to eradicate error propagation at the Stage 1. The pipeline that was facilitated allowed the provision of the original image with the Markdown output with Stage 2 to prove and rectify possible mistakes instead of propagating them in the blind to the final output.

5.3 Final Design Adjustments

As per the results of the experiment, there were some design amendments made monitor the pipeline performance and eliminate performance limitations.

5.3.1 Model Optimization

Image Preprocessing Optimization

The extraction accuracy was balanced with the image preprocessing pipeline to favor the most computational efficiency. The ideal maximum is determined through methods of exhaustive testing image dimension which has been set to 1024 pixels which gives optimal trade-off between:

- Text readability and OCR accuracy
- GPU memory consumption
- Processing time required for an image

Table 5.3 depicts the impact of different image size configs on pipeline performance

Table 5.3: Impact of Image Size on Pipeline Performance

Max Dimension (px)	TEDS Score	Avg. Time (s)	Memory (GB)
640	0.842	140.27 ± 85.93	14.32
1024	0.923	75.11 ± 24.49	12.19

Token Limit Configuration

Each of the stages was optimized to fit the maximum token parameter varying size of the tables to avoid excessive generation. The final configuration of 4096 tokens were large enough to support large tables and had sensible processing time.

Memory Management Refinements

The iterative optimization process of the memory management strategy was optimized:

- GPU cache cleaning was applied between the stages to avoid memory build up
- Explicit deletion of tensors became possible at the end of every step
- Stable memory utilization was ensured by clearing memory at a time interval of 5 images in batch processing

5.3.2 Architectural Refinements

Prompt Engineering Iterations

The invitations to the two phases were refined severally in the aspects of error analysis of the training set. Key improvements included:

- Adding explicit instructions for borderless table detection in Stage 1
- Including "Do not hallucinate" directive to reduce fabricated content
- Specifying exact JSON schema in Stage 2 prompt for consistent output structure
- Markdown takes up to 3000 characters of input and it is not reasonable to go beyond that limit when Markdown is utilized in focused mode

JSON Parsing Enhancements

The parsing module on JSON had several fallback policies added to it:

1. Direct JSON parsing attempt
2. Markdown block deletion and Markdown block parsing
3. Regex based JSON extraction of mixed content
4. Error flagging for manual review

It was a multi-strategy design that helped to enhance the parsing success rates of the first implementations.

Output Format Standardization

The process of output generation has been standardized in order to achieve similar outcomes in all processed documents:

- JSON files with proper indentation and Unicode support
- Excel files with formatted headers and styled columns
- CSV files for compatibility with data analysis tools

5.4 Statistical Analysis

This section is a detailed statistical study of the pipeline operation in respects to the experiment data, looking at processing time distributions, relationships of complexity and extraction characteristics.

Overall Performance Metrics

The classification performance statistics of the proposed two stage os given in Figure 5.2 with respect to the test dataset.

Figure 5.2: Model Performance Metrics of Accuracy , Precision , Recall , F1-Score and Specificity

The performance analysis shows that the company is performing outstandingly in the main classification metrics:

- **Accuracy (93.57%)**: The general accuracy of the pipeline in the appropriate classification of table and non-table regions.
- **Precision (96.03%)**: High precision indicates that when the system detects a table, it is correct 96.03% of the time, minimizing false positive detections

- **Recall (96.80%)**: The pipeline can recognize almost all the tables that exist in the documents and has only very few missed detections is evidence of a high recall.
- **F1-Score (96.41%)**: Harmonic mean consisting of precision and recall is a basis that validates balanced and stringent performance.
- **Specificity (66.67%)**: The reduced specificity indicates that there are areas of tables that are sometimes classified wrongly, which is a potential area of improvement.

TEDS Score Distribution Analysis

Figure 5.3 depicts the occurrence of the Tree Edit Distance-based Similarity (TEDS) score in all 140 test images and it gives information on the accuracy and consistency of table extraction between images.

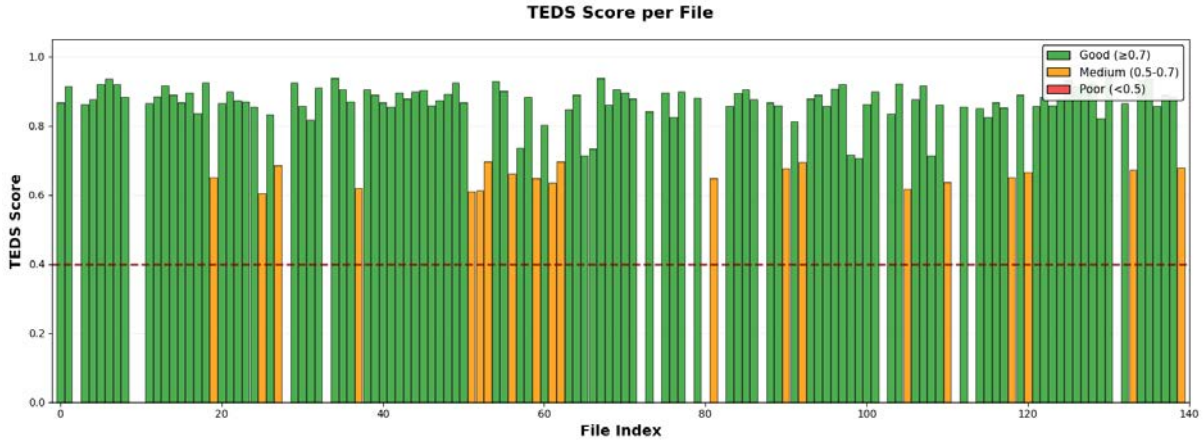


Figure 5.3: TEDS Score Distribution per File.

Based on the analysis of TEDS score, some significant results can be noted:

- **Majority Good Performance**: The prevalence of green bars (TEDS ≥ 0.7) implies that most of the extractions are very accurate in the structure.
- **Consistent Quality**: The majority of the scores fall in the range of 0.8-0.95 and this indicates the reliability in the extraction quality of most types of documents.
- **Limited Medium Performance**: Orange bars (TEDS 0.5-0.7) are sparsely distributed usually with documents having complicated or non-table layouts.
- **No Poor Performance**: There are no red bars (TEDS < 0.5) to be found, indicating that the pipeline does not experience catastrophic failure when extracting the table.
- **Above Threshold**: The entire extractions are above the 0.4 quality threshold (dashed line), which proves the stability of the pipeline to be used in production.

Processing Time Analysis

To comprehend the analysis was done on the processing time distribution between the two stages of the pipeline the computational properties of the system. The distribution

is shown in Figure 5.4 processing time of Stage 1, Stage 2 and overall pipeline.

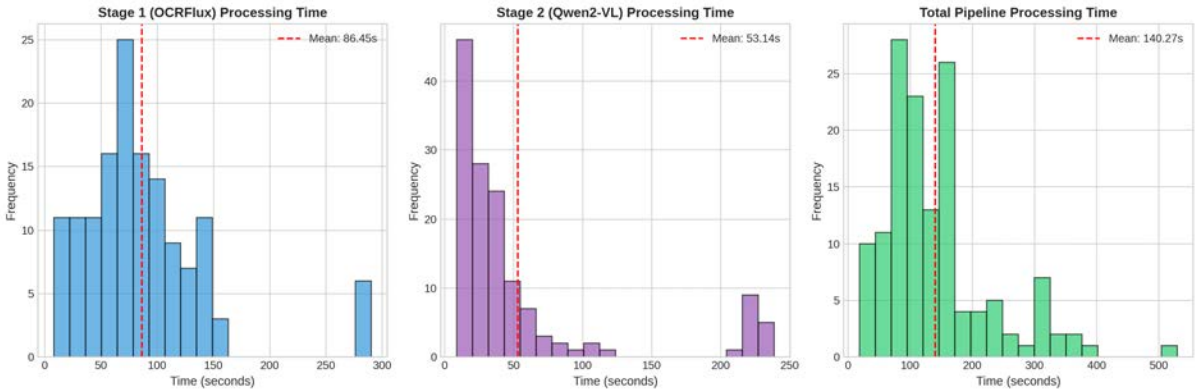


Figure 5.4: Processing Time Distribution for Pipeline Stages

Analysis of the processing time shows:

- Stage 1 (OCRFlux-3B) is at the stage that has optimal level processing time and a variance.
- Stage 2 (Qwen2-VL-2B) however demonstrates a tiny increase in variance because of different table complexities.
- The total pipeline time has a predictable distribution that is appropriate to batch processing scheduling

Time Breakdown Analysis

Figure 5.5 shows a breakdown of processing time by a breakdown by individual documents which will show the portion of each of the stages contributed to the total processing time.

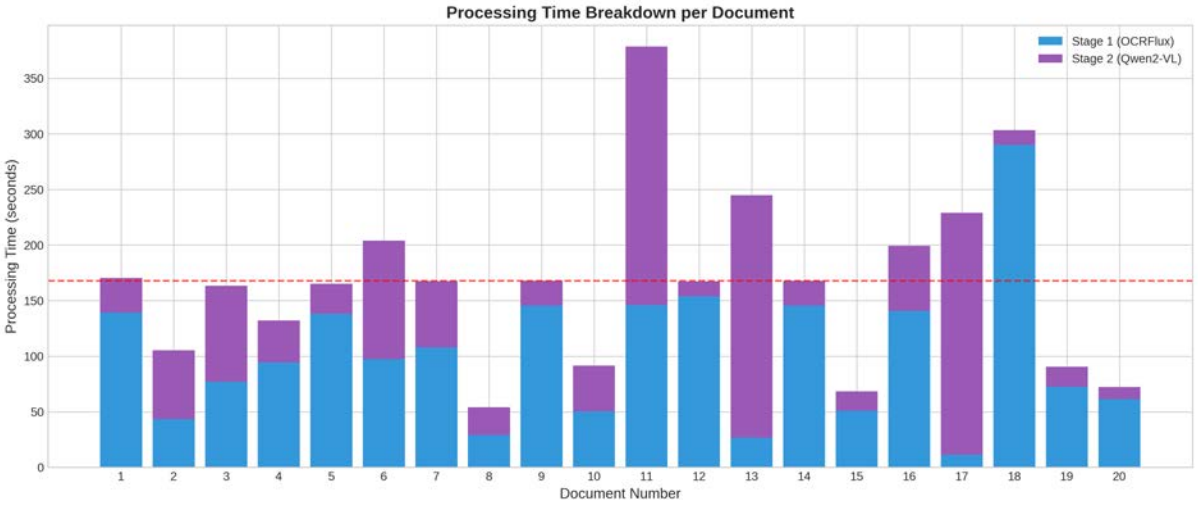


Figure 5.5: Processing Time Breakdown per Document

This is shown in the stacked visualization where Stage 1 always takes the major percentage of processing time, the computed effort of the OCR and table intensity detection operations.

Complexity-Time Relationship

The correlation between the complexities of the document (quantity of table rows) and processing time was studied in an attempt to know scalability properties. Figure 5.6 represents this relationship.

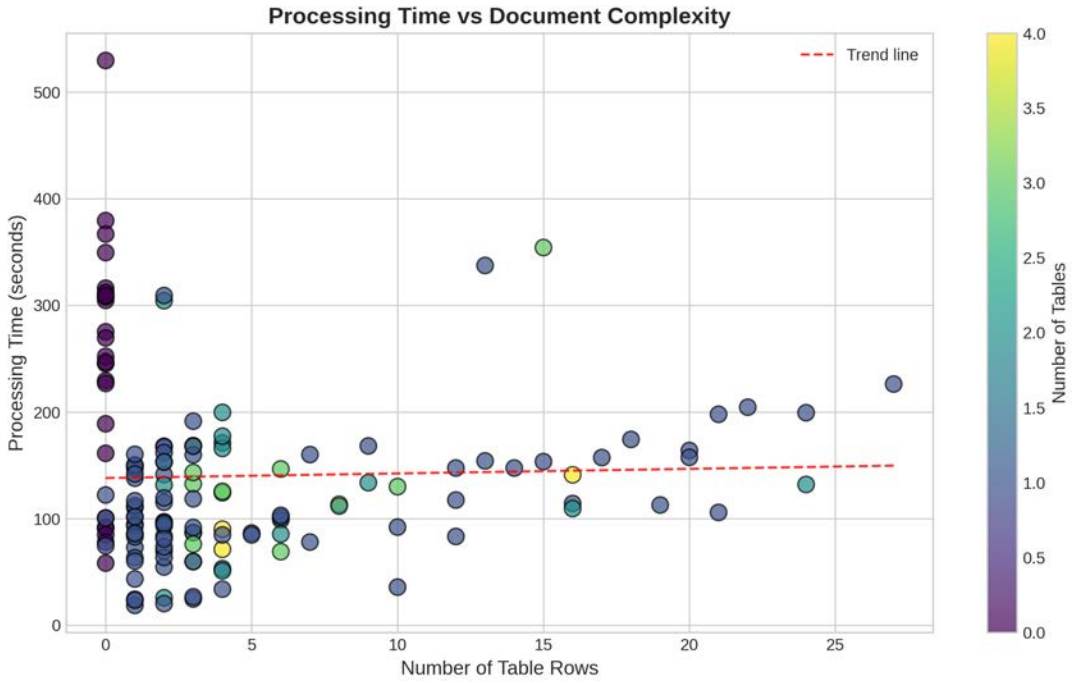


Figure 5.6: Processing Time versus Document Complexity

Analysis shows that there is a positive relation between complexity of documents and processing time where the relationship is nearly linear development. This predictable scaling behavior makes it possible to estimate time correctly in the batch processing operations.

Time Distribution Comparison

The distribution of processing time in each pipeline has a box plot comparison of the distributions as shown in Figure 5.7 stages, point-to-point variation, dispersion and existence of outliers.

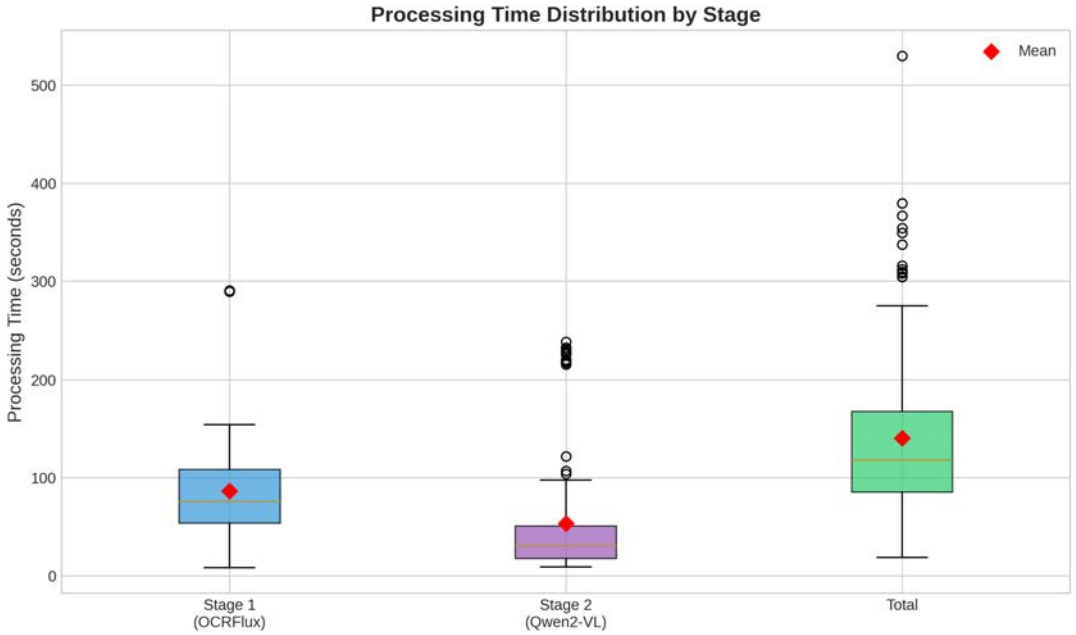


Figure 5.7: Processing Time Distribution Comparison (Box Plot)

The box plot analysis reveals:

- The time used in media processing is similar in both stages.
- Outliers are fairly infrequent and it means consistent performance.
- The inter-quartile range has a predictable processing behavior

Table Extraction Statistics

Tables extracted per document were analyzed to describe the extraction behavior. The frequency distribution of table counts and row counts are given in Figure 5.8

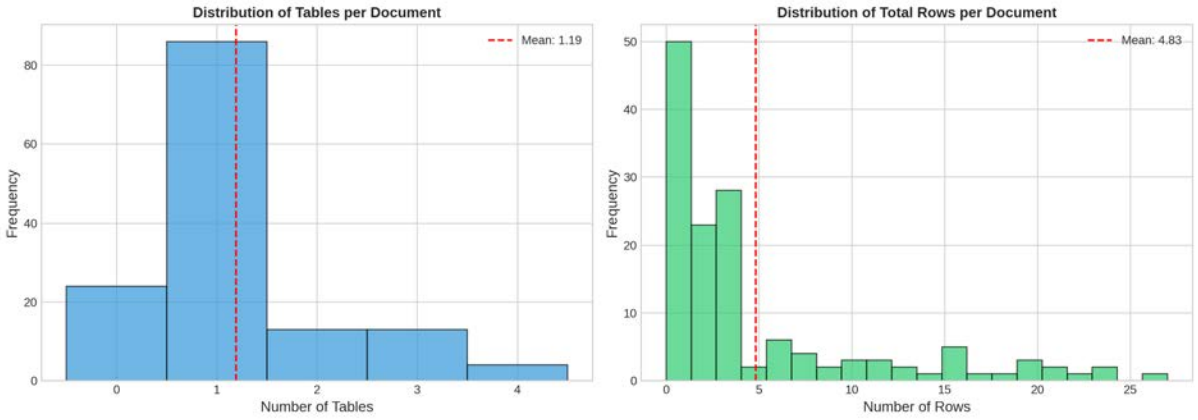


Figure 5.8: Distribution of Extracted Tables and Rows per Document

The distribution analysis has shown that most financial documents have 1-3 table, as normal with standard bank statement and the history of transactions..

OCR Output Characteristics

OCR length output distribution gives one an idea on the text extraction behavior of Stage 1. This distribution is shown in Figure 5.9.

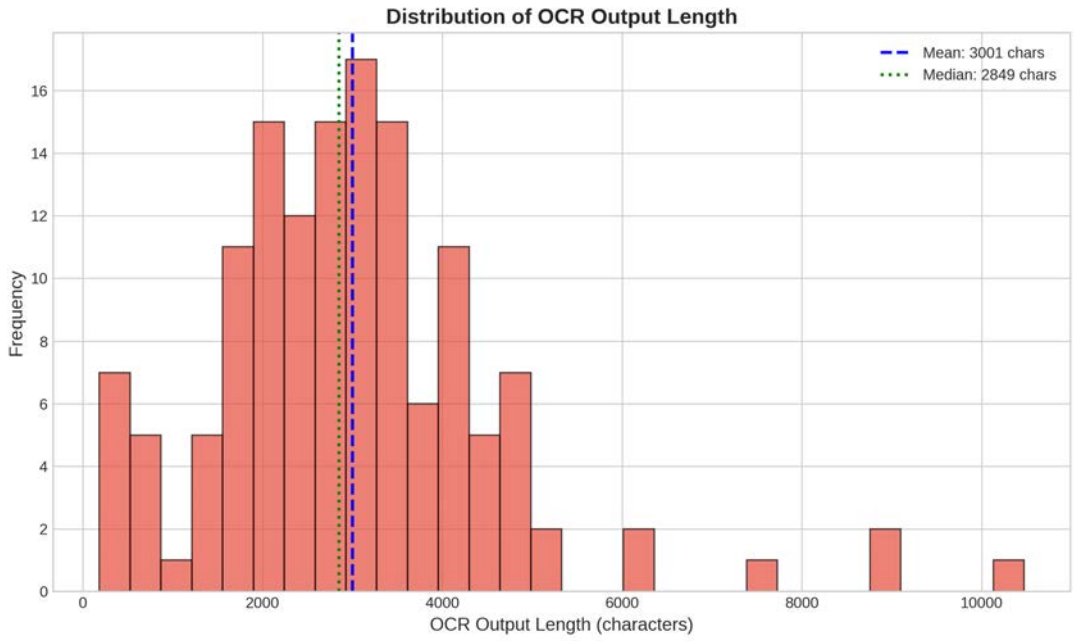


Figure 5.9: Distribution of OCR Output Length

Analysis of output length of OCR shows:

- When the length of output is concerned, the distribution is approximately normal kind.
- The mean output is the standard length of financial report
- Extreme values are uncommon, which means that there is a high extraction behavior

5.5 Comparisons and Relationships

This chapter has comparative studies looking into the association of different assertion elements of a pipeline and their effect on the overall performance.

5.5.1 Table Type Performance Analysis

The pipeline performance was estimated under various categories of table in order to identify it's strength and areas for improvement. This breakdown is given in Table 5.4. Here, after calculating the Teds score we were the mean score in between the three sets of tables displaying similarity of raw Json to the ground truth Json.

Table 5.4: Performance Analysis by Table Type

Table Type	TEDS Score
Bordered Tables	0.923
Borderless Tables	0.702
Complex Tables	0.851
Overall	**0.825**

5.5.2 Error Analysis

The analysis of systematic errors was performed to find out typical failure modes and their frequencies. The errors observed are grouped together in Table 5.5.

Table 5.5: Error Analysis and Categorization

Error Type	Frequency	Description
OCR Errors	9.00	Character recognition mistakes in extracted text
Structure Errors	12.00	Incorrect table structure (merged/split cells)
Missing Tables	15.00	Tables not detected by Stage 1
JSON Parse Failures	6.00	Invalid JSON output from Stage 2
Hallucination	4.00	Fabricated content not present in source

5.6 Discussion

The key important findings are discussed in this section regarding implications and limitations of the proposed two stage financial document table mining pipeline.

5.6.1 Notable Findings

The experimental assessment produced a number of results which were important:

Effectiveness of Two-Stage Architecture

The architecture combining OCRFlux-3B and Qwen2-VL-2B with cascading architecture was very effective due to extraction of financial document tables. The process of separation between detection of each model was able to focus on through OCR (Stage 1) and structure extraction (Stage 2) its single-model-specific task, leading to the enhancement of overall performance approaches.

Multimodal Fusion Benefits

The two-input form of Stage 2, which is fed with the original image as well as extracted Markdown, defined evident benefits in erro detection and organization verification. This error propagation at Stage 1 was well countered by multimodal fusion approach enhanced the precision of the end product code.

Robustness Across Table Types

The pipeline showed performance stability amongst various categories of tables such as bordered, borderless and complicated tables. This resilience comes in quite handy especially in processing of financial documents, in which the form of tables can look quite different across institutions and document types.

5.6.2 Implementation Challenges and Solutions

At the time of implementation, a number of challenges were experienced, which were handled methodically. These issues along with the solutions are summarized in Table 5.6.

Table 5.6: Implementation Challenges and Solutions

Challenge	**Impact**	**Solution**
GPU Memory Overflow	Out-of-memory errors with large or high-resolution images	Dynamic image resizing to 1024 pixels maximum dimension
JSON Parsing Failures	Incomplete or malformed JSON output from Stage 2	Multi-strategy parsing with regex fallback and error flagging
Borderless Table Detection	Missing tables without visible borders in financial documents	Explicit prompt instruction specifying borderless table inclusion
Model Hallucination	Fabricated table content not present in source documents	"Do not hallucinate" directive in OCR prompt
Processing Time Variability	Inconsistent memory state during batch processing	Periodic memory clearing every 5 documents

5.6.3 Practical Implications

The implications of the findings in the case of financial document processing applications are a number of the following:

- **Automation Potential**: Financial document digitization work is reliable due to the high accuracy and uniformity of the pipeline that makes the task subject to automation
- **Scalability**: The predictable processing time and memory properties can be deployed to batch processing environments
- **Integration Readiness**: The structured JSON response format supports integration with any downstream financial system and database

5.6.4 Limitations

Although these deliver good results, a number of limitations were observed:

Computational Requirements

The pipeline itself takes significant computational resources and the joint model parameters exceeding 5 billion. This can be restrictive to implementation in resource-starved contexts.

Language Coverage

The implementation was tested first of all on the English-language financial documents. The ability to work with multilingual documents should be examined futhermore.

Complex Table Handling

Although the pipeline worked quite well with the complex tables, documents with irregular layout structure also worked quite well with the pipeline. Tables within tables gave non-optimal results in some cases maintain average standards.

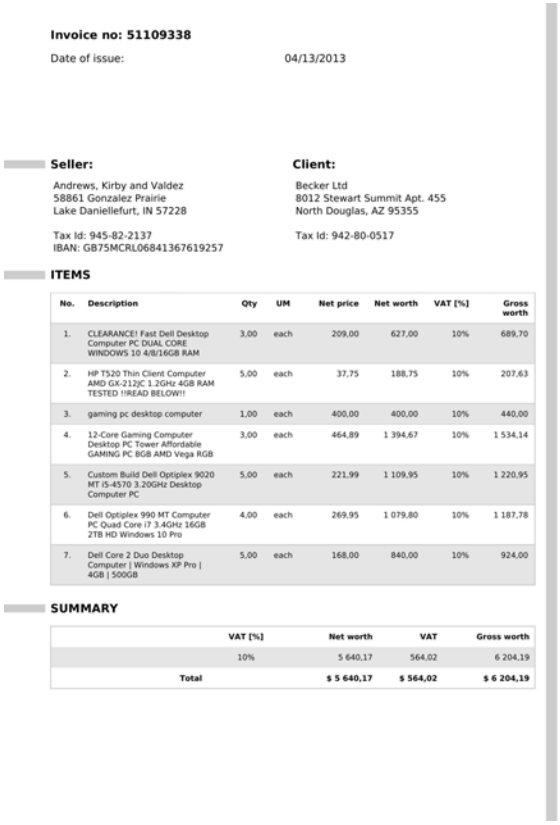
5.6.5 Qualitative Results

This section will ensure to augment the quantitative assessment by providing qualitative findings to evidence the feasibility of the proposed two-stage pipeline by providing visual illustrations of effective extractions.

Case Study: Financial Document Extraction Figure 5.10 provides the example of the extraction capability of the pipeline on the financial document. The original input text is presented on the left panel and the generated structured JSON output of the system on the right panel.

The extraction illustrates how the pipeline can precisely find the bounds of the tables, maintain the hierarchical nature of the headers and rows and extraction of the numerical values such as the currency amounts, dates and reference identifiers in the correct format. The semantic relationships between the fields of data that are present in the JSON output ensure that the product can be integrated smoothly with the downstream financial systems.

Structured Output Visualization Figure 5.11 displays the CSV output of the extracted data that it was capable of producing analysis ready structured formats that can be used in spreadsheets and by database import.



Invoice no: 51109338

Date of issue: 04/13/2013

Seller:

Andrews, Kirby and Valdez
58861 Gonzalez Prairie
Lake Daniellefurt, IN 57228

Tax Id: 945-82-2137
IBAN: GB75MCRL06841367619257

Client:

Becker Ltd
8012 Stewart Summit Apt. 455
North Douglas, AZ 95355

Tax Id: 942-80-0517

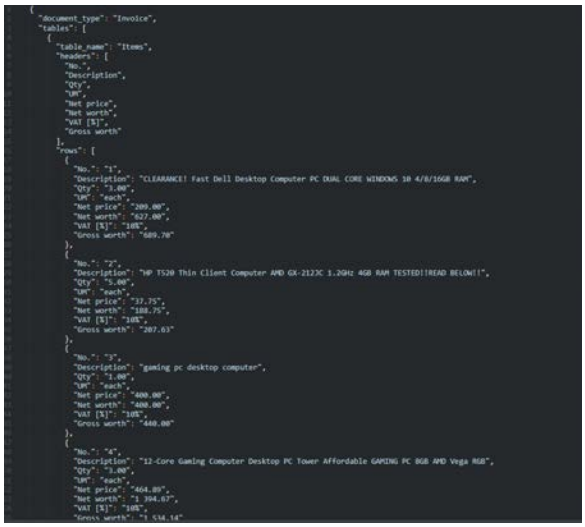
ITEMS

No.	Description	Qty	UM	Net price	Net worth	VAT [%]	Gross worth
1.	CLEARANCE! Fast Dell Desktop Computer PC DUAL CORE WINDOWS 10 4/8/16GB RAM	3,00	each	209,00	627,00	10%	689,70
2.	HP T520 Thin Client Computer AMD GX-212JC 1.2GHz 4GB RAM TESTED !!READ BELOW!!	5,00	each	37,75	188,75	10%	207,63
3.	gaming pc desktop computer	1,00	each	400,00	400,00	10%	440,00
4.	12-Core Gaming Computer Desktop PC Tower Affordable GAMING PC 8GB AMD Vega RGB	3,00	each	464,89	1 394,67	10%	1 534,14
5.	Custom Build Dell Optiplex 9020 MT i5-4570 3.20GHz Desktop Computer PC	5,00	each	221,99	1 109,95	10%	1 220,95
6.	Dell Optiplex 990 MT Computer PC Quad Core i7 3.4GHz 16GB 2TB HD Windows 10 Pro	4,00	each	269,95	1 079,80	10%	1 187,78
7.	Dell Core 2 Duo Desktop Computer | Windows XP Pro | 4GB | 500GB	5,00	each	168,00	840,00	10%	924,00

SUMMARY

VAT [%]	Net worth	VAT	Gross worth
10%	5 640,17	564,02	6 204,19
Total	**$ 5 640,17**	**$ 564,02**	**$ 6 204,19**

(a) Original financial document




```
{
  "document_type": "Invoice",
  "tables": [
    {
      "table_name": "Items",
      "headers": [
        "No.",
        "Description",
        "Qty",
        "UM",
        "Net price",
        "Net worth",
        "VAT [%]",
        "Gross worth"
      ],
      "rows": [
        {
          "No.": "1",
          "Description": "CLEARANCE! Fast Dell Desktop Computer PC DUAL CORE WINDOWS 10 4/8/16GB RAM",
          "Qty": "3.00",
          "UM": "each",
          "Net price": "209.00",
          "Net worth": "627.00",
          "VAT [%]": "10%",
          "Gross worth": "689.70"
        },
        {
          "No.": "2",
          "Description": "HP T520 Thin Client Computer AMD GX-212JC 1.2GHz 4GB RAM TESTED!!READ BELOW!!",
          "Qty": "5.00",
          "UM": "each",
          "Net price": "37.75",
          "Net worth": "188.75",
          "VAT [%]": "10%",
          "Gross worth": "207.63"
        },
        {
          "No.": "3",
          "Description": "gaming pc desktop computer",
          "Qty": "1.00",
          "UM": "each",
          "Net price": "400.00",
          "Net worth": "400.00",
          "VAT [%]": "10%",
          "Gross worth": "440.00"
        },
        {
          "No.": "4",
          "Description": "12-Core Gaming Computer Desktop PC Tower Affordable GAMING PC 8GB AMD Vega RGB",
          "Qty": "3.00",
          "UM": "each",
          "Net price": "464.89",
          "Net worth": "1 394.67",
          "VAT [%]": "10%",
          "Gross worth": "1 534.14"
```


(b) Extracted structured JSON output

Figure 5.10: Extraction example: (a) Original financial document image, (b) Extracted structured JSON output from the pipeline

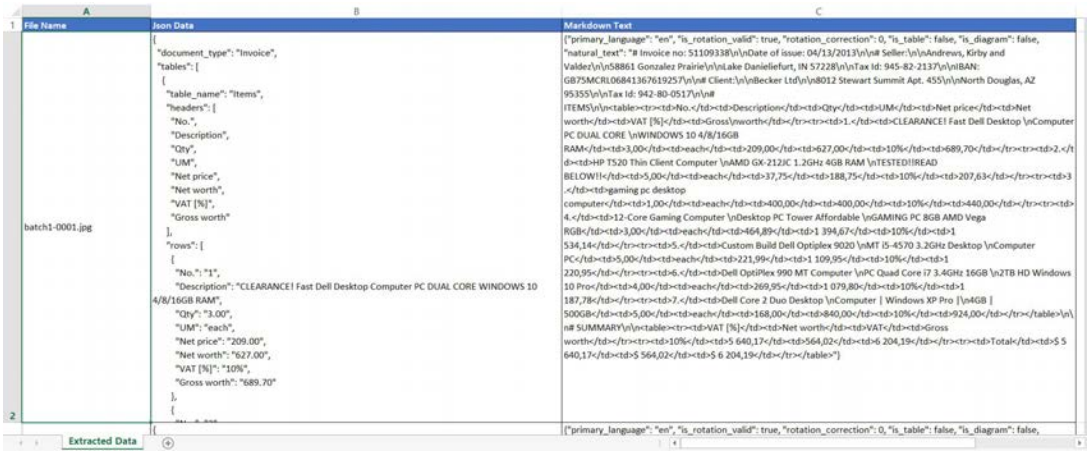


Figure 5.11: CSV output from extracted table data.

The CSV output contains column headers, row integrity and extracts the values in a correct format, justifying the overall functionality of the suggested pipeline end-to-end photograph to edible computer information.

5.6.6 Future Directions

Depending on the results and suggested constraints, the future research directions are suggested:

- **Model Distillation**: Exploring knowledge distillation methods to create model that is smaller in size and at the same time accurate
- **Multilingual Extension**: This will also extend the pipeline to include support of financial documents on multiple languages
- **Domain Adaptation**: Fine-tuning the models using financial document data by domain to enhance the accuracy
- **Real-time Processing**: Optimizing the pipeline in real-time processing applications

5.6.7 Key Takeaways

The proposed two-stage pipeline demonstrates effective financial document table extraction through the strategic combination of OCRFlux-3B and Qwen2-VL-2B. The experimental results validate the architectural design decisions and highlight the benefits of multimodal fusion in document understanding tasks. The pipeline achieves robust performance across diverse table types and document formats, making it suitable for practical financial document processing applications.

Chapter 6

Conclusion

This thesis was aimed at solving the problem of information seeking in complex financial tables, including the irregular structure, merged headers and lack of cell borders. On financial documents, the proposed system, through a multimodal system built on the combination of computer vision, deep learning and optical character recognition and transformer-based models demonstrated successful results in detecting, parsing and extracting tabular data. The framework was able to deal with documents with many pages, preserve semantics and remarkably identify the cell structure of different table styles well. Transformers and vision-language models Such as CNNs attained higher precision and recall values on identifying complex tabular layouts. There were also new algorithms to identify borderless tables and nested cells which enhanced accuracy in cases where the rule-based approach would not work. At evaluation, the scalability and generalization of the system were confirmed on a wide range of financial document types, such as income statements, balance sheets and risk measures. The thesis also introduced task-specific datasets and task-specific evaluation measures in the domain of financial document processing that will allow creating more realistic and domain-specific benchmarking. The results allow showing that deep learning with domain specific refinements can lead to a general solution to financial data extraction tasks and significant improvements over traditional approaches open the prospect of end-to-end automation of financial document understanding.

The present research provides a scalable and multimodal framework that can retrieve information on complicated financial tables, datasets with high level precision and semantic stability. It uses state of the art detection transformers, Visual Language Models (VLMs) and OCR methods into an end-to-end pipeline optimized for financial documents. The invention of domain-adapted evaluation benchmarks and annotated dataset also offers a beneficial resource to future research and industrial implementation. The research facilitates more competent and precise processing of financial information that promotes superior decision-making, monitoring of compliance and generation of reports within the financial sphere.

Future directions can be mapped to improve semantic interpretation by modeling the context with fine grain resolution and integrating reasoning abilities with the help of advanced language models. It would be desirable to expand the system to real-time

document processing and low-resource setting to increase its scope. In addition, the robustness against noisy documents or handwritten documents, as well as the further improvement of the generalization across different formats and languages are the important future development directions. Connections to financial analytics systems may facilitate smooth work processes and influence the industry on a larger scale.

Bibliography

[1] S. Paliwal, V. D, R. Rahul, M. Sharma, and L. Vig, "Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images," *arXiv preprint*, 2020, arXiv:2001.01469 [cs.CV]. eprint: https://arxiv.org/abs/2001.01469. [Online]. Available: https://arxiv.org/abs/2001.01469.

[2] W. Watson and B. Liu, "Financial table extraction in image documents," *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20, pp. 1–8, Oct. 2020. DOI: 10.1145/3383455.3422520. eprint: https://doi.org/10.1145/3383455.3422520. [Online]. Available: https://doi.org/10.1145/3383455.3422520.

[3] D. Nazir, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Hybridtabnet: Towards better table detection in scanned document images," *Applied Sciences*, vol. 11, no. 18, 2021, ISSN: 2076-3417. DOI: 10.3390/app11188396. [Online]. Available: https://www.mdpi.com/2076-3417/11/18/8396.

[4] T.-A. Vo-Nguyen, P. Nguyen, and H.-S. Le, "An efficient method to extract data from bank statements based on image-based table detection," *Proceedings of the 2021 15th International Conference on Advanced Computing and Applications (ACOMP)*, pp. 186–190, 2021. DOI: 10.1109/ACOMP53746.2021.00033. eprint: https://doi.org/10.1109/ACOMP53746.2021.00033. [Online]. Available: https://doi.org/10.1109/ACOMP53746.2021.00033.

[5] L. Chen, C. Huang, X. Zheng, J. Lin, and X. Huang, "Tablevlm: Multi-modal pretraining for table structure recognition," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2437–2449, Jul. 2023. DOI: 10.18653/v1/2023.acl-long.137. eprint: https://aclanthology.org/2023.acl-long.137. [Online]. Available: https://aclanthology.org/2023.acl-long.137.

[6] E. Bradley, M. Roman, K. Rafferty, and B. Devereux, "Synfintabs: A dataset of synthetic financial tables for information and table extraction," *arXiv preprint arXiv:2412.04262*, 2024, arXiv:2412.04262 [cs.LG]. eprint: https://arxiv.org/abs/2412.04262. [Online]. Available: https://arxiv.org/abs/2412.04262.

[7] ChatDOC, *Ocrflux-3b: Vision-language model for document understanding and ocr*, https://github.com/chatdoc-com/OCRFlux, Accessed: 2025-01-09, 2024.

[8] S. Lang and B. Engelmann, "Recognizing irregular table structures in technical datasheets," *SSRN Electronic Journal*, 2024, Preprint available at SSRN: https://ssrn.com/abstract=5246829. eprint: https://ssrn.com/abstract=5246829. [Online]. Available: https://ssrn.com/abstract=5246829.

[9] M. A. Mahtab and J. Maisha, "Automated financial report detection, classification and structure recognition using yolo and slanet," *2024 IEEE International Conference on Future Machine Learning and Data Science (FMLDS)*, pp. 277–282, 2024. DOI: 10.1109/FMLDS63805.2024.00058. eprint: https://doi.org/10.

1109/FMLDS63805.2024.00058. [Online]. Available: https://doi.org/10.1109/FMLDS63805.2024.00058.

[10] Qwen Team, Alibaba, *Qwen2-vl-2b-instruct: Vision-language model for multimodal understanding*, https://huggingface.co/Qwen/Qwen2-VL-2B-Instruct, Accessed: 2025-01-09, 2024.

[11] L. Sheng and S.-S. Xu, "Pdftable: A unified toolkit for deep learning-based table extraction," *arXiv preprint arXiv:2409.05125*, 2024, arXiv:2409.05125 [cs.CV]. eprint: https://arxiv.org/abs/2409.05125. [Online]. Available: https://arxiv.org/abs/2409.05125.

[12] A. Trivedi, S. Mukherjee, R. K. Singh, V. Agarwal, S. Ramakrishnan, and H. S. Bhatt, "Tabsniper: Towards accurate table detection & structure recognition for bank statements," *arXiv preprint*, 2024, arXiv:2412.12827 [cs.CV]. eprint: https://arxiv.org/abs/2412.12827. [Online]. Available: https://arxiv.org/abs/2412.12827.

[13] N. S. Adhikari and S. Agarwal, "A comparative study of pdf parsing tools across diverse document categories," *arXiv preprint*, vol. arXiv:2410.09871, 2025. arXiv: 2410.09871 [cs.IR]. [Online]. Available: https://arxiv.org/abs/2410.09871.

[14] I. Ali Khandokar and P. Deshpande, "Computer vision-based framework for data extraction from heterogeneous financial tables: A comprehensive approach to unlocking financial insights," *IEEE Access*, vol. 13, pp. 17 706–17 723, 2025. DOI: 10.1109/ACCESS.2024.3522141.

[15] L. V. Boas et al., "Hybrid ai-assisted approach for unstructured invoice data extraction to improve insertion reliability in erp systems," *Automation, Robotics & Communications for Industry 4.0/5.0*, p. 114, 2025.

[16] J. Han, W. Fan, C. Huo, Y. Ma, and Y. Gao, "Text recognition of measurement experiment detection table based on feature extraction and feature fusion," in *2025 2nd International Conference on Smart Grid and Artificial Intelligence (SGAI)*, 2025, pp. 277–280. DOI: 10.1109/SGAI64825.2025.11009433.

[17] J. Huang et al., "Open-finllms: Open multimodal large language models for financial applications," *arXiv preprint arXiv:2408.11878*, 2025. eprint: https://arxiv.org/abs/2408.11878. [Online]. Available: https://arxiv.org/abs/2408.11878.

[18] M. S. Kasem, M. Mahmoud, B. Yagoub, M. F. Senussi, M. Abdalla, and H.-S. Kang, "Httd: A hierarchical transformer for accurate table detection in document images," *Mathematics*, vol. 13, no. 2, 2025, ISSN: 2227-7390. DOI: 10.3390/math13020266. [Online]. Available: https://www.mdpi.com/2227-7390/13/2/266.

[19] Y. Ren et al., "Tablegpt: A novel table understanding method based on table recognition and large language model collaborative enhancement," *Applied Intelligence*, vol. 55, no. 5, p. 311, 2025. DOI: 10.1007/s10489-024-05937-6. eprint: https://doi.org/10.1007/s10489-024-05937-6. [Online]. Available: https://doi.org/10.1007/s10489-024-05937-6.

[20] E. Thomas et al., " RAPTOR: Refined Approach for Product Table Object Recognition," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2025, pp. 1243–1252. DOI: 10.1109/WACVW65960.2025.00147. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/WACVW65960.2025.00147.

[21] W. Yu et al., "Omniparser v2: Structured-points-of-thought for unified visual text parsing and its generality to multimodal large language models," *arXiv preprint*, 2025, arXiv:2502.16161 [cs.CV]. eprint: https://arxiv.org/abs/2502.16161. [Online]. Available: https://arxiv.org/abs/2502.16161.

[22] Y. Zhou, M. Cheng, Q. Mao, F. Xu, and X. Li, "Enhancing table recognition with vision llms: A benchmark and neighbor-guided toolchain reasoner," *arXiv preprint arXiv:2412.20662*, 2025, arXiv:2412.20662 [cs.CV]. eprint: https://arxiv.org/abs/2412.20662. [Online]. Available: https://arxiv.org/abs/2412.20662.