An End-to-End Hybrid Document Intelligence System for Unstructured and Semi-Structured Data

Swapnaja Yadav[1], Soleha Tamboli[1], Shaivi Jaiswal[1], Yash Lulla[1], Shivam Angral[1] and Preeti Bailke[2]

[1] Department of IT, Vishwakarma Institute of Technology, Pune, India

swapnaja.yadav242@vit.edu

soleha.tamboli24@vit.edu

yash.lulla24@vit.edu

viraj.shaivi241@vit.edu

shivam.angral24@vit.edu

[2] Assistant Professor, Department of IT, Vishwakarma Institute of Technology, Pune, India
preeti.bailke@vit.edu

Abstract. Extracting essential information from unstructured and mixed types of documents is a tremendous challenge in many organizations. Documents like scanned receipts, invoices, and native PDFs often contain noise, uneven layouts, faded text, and formatting differences, which makes it difficult for traditional OCR and rule-based systems to read them accurately. In this work, we present a complete end-to-end Document Intelligence system that cleans and preprocesses documents, uses EasyOCR to extract text, groups tokens into proper lines and columns, and then applies rules to pull out key fields. For digital PDFs and Word files, the system extracts clean text and generates meaningful summaries using an LLM. The system also includes MongoDB for storing results, a FastAPI backend for processing, a user-friendly web interface, and a text-to-speech feature. Testing on real scanned receipts and native documents shows that the system extracts information more accurately, works better on low-quality inputs, and answers user questions reliably. Overall, this hybrid AI pipeline is expandable and useful for automating document understanding across different types of files.

Keywords: Document Intelligence, OCR, EasyOCR, Unstructured Data, LLM, Information Extraction, Receipt Digitization.

1 Introduction

Unstructured and semi-structured documents make up a huge part of current digital content. Documents such as receipts, bills, financial statements, contracts, forms, and reports are produced in massive numbers covering a wide range of industries like retail, banking, healthcare, and government as well. However, the task of extracting organized and machine-readable information from these documents remains laborious due to layout variations, scanning artifacts, faded thermal prints, handwritten notes, and the lack of standardization. Conventional rule-based systems are highly template-dependent and lack robustness in the presence of varying document layouts. Similarly, traditional Optical Character Recognition (OCR) systems are prone to difficulties in low-quality scans, intricate backgrounds, and irregular typography.

Current trends in document intelligence revolve around deep learning-based and multimodal models that simultaneously model the text and image modalities. LayoutLMv3, a transformer-variant model, integrates layout, image, as well as text embeddings to upgrade structured document analysis. At the same time, OCR-free models

R. Vasanth Kumar Mehta et al. (eds.), *Proceedings of the International Conference on Intelligent Systems for a Sustainable Future (ISSF 2026)*, Atlantis Highlights in Intelligent Systems 16,
https://doi.org/10.2991/978-94-6239-693-7_103

like Donut master document portrayals directly from vision-language transformers, making them independent of text detection pipelines. These models have exhibited state-of-the-art outcomes on accepted benchmarks, mostly on structured invoices and forms. However, these models are computationally high-priced, require huge amounts of annotated data, and request domain-specific fine-tuning, making them less adjustable to resource-constrained enterprise settings.

Concurrently, hybrid models that integrate neural extraction with symbolic reasoning have come out for their efficiency and understandability. Neuro-symbolic models and rule-enhanced pipelines aim to find a balance between robustness and usability for real-world applications. However, very little research has been conducted on integrating the entire process of OCR, spatial grouping, rule-based extraction, LLM-based reasoning, and scalable backend integration into a single, end-to-end system. Real-world receipts and low-quality scans are also unexplored compared to clean benchmark datasets.

To address these research gaps, this paper gives an absolute end-to-end Document Intelligence system that combines document pre-processing, EasyOCR-based text extraction, spatial token assembling, rule-based key-value pair extraction, and LLM-based summarization and question-answering. The system is planned with usability in mind, featuring a flexible FastAPI backend, MongoDB database, and user-friendly web interface for real-world usability. For scanned documents, image processing is used to increase OCR accuracy, and for native PDF and DOCX files, structured text extraction is integrated with semantic summarization.

This research provides the given key contributions:
- A hybrid document intelligence system that combines OCR, spatial grouping, rule-based extraction, and LLM reasoning.
- Evaluation of EasyOCR versus Tesseract on noisy retail receipts
- A structured extraction strategy that uses positional information and token clustering to hold unorganized receipt layouts.
- LLM-assisted summarization and answer extraction to lessen hallucinations with controlled prompting.
- A deployable backend-frontend infrastructure using FastAPI and MongoDB for real-world scalability.
On both scanned and native receipts, the system is shown to be more demanding on low-quality inputs, highly correct on key financial values, and sound in reasoning. In summary, it comes up with a real-world implementation framework and observed analysis of hybrid AI techniques for document intelligence.

2 Related works

Information extraction on semi-structured documents has shifted from traditional rule-governed systems to the current learning-driven systems. In their survey on information extraction on PDFs, Jayaram and Sangeeta[1] discuss the evolution of information extraction methods on PDFs and classify these methods into three categories: statistical approaches, NLP Techniques, and Machine Learning Models. However, Skalicky et al.[2] and his team point out that even in information extraction, standardization is still absent. According to Skalicky and his team, the current evaluation metric fails to address the requirements in practical business-to-business communications. Therefore, they proposed two novel problem definitions for information extraction—Key Information Localization and Extraction (KILE) and Line Item Recognition (LIR). In addition, they proposed the need for bigger synthetic datasets that support privacy and that resemble true document characteristics.

By tackling complex document layouts more efficiently, current research relies on deep learning models that focus on the visual aspect of documents, rather than merely their texts. Nguyen Dang and Nguyen Thanh[3] propose the use of a “Multi-Stage Attentional U-Net” for document processing based on their 2D character-grid representation. Their method involves an encoder-decoder architecture that employs self-attention and box convolutions and outperforms traditional U-Nets at modeling long-range visual dependencies by 40% fewer

parameters. Meanwhile, research by Avcı et al. [4] contrasts the practicability of various deep learning models for performing invoice extraction. They match up Graph Convolutional Networks (GCNs) opposed to transformer models such as LayoutLMv1 and LayoutLMv3 and show that, although the former can be successful without pre-training, the second exhibits the best precision rate (0.95) by integrating both text and image information.

There is also an attempt to design end-to-end information extraction systems that are suitable for an enterprise environment. Vishwanath et al.[5] introduce "Deep Reader," an end-to-end system that utilizes deep vision networks to identify document entities like text regions and tables and convert them into a relational form. The most notable aspect of it is its language interface that allows people without technical knowledge to analyze data from a document through conversational SQL. Massarenti and Lazzarinetti[6] design an RPA-specific pipeline for industry automation. They combine Image Denoising with GAN networks and a Siamese Neural Network for template match detection from a single image. They claim a success rate of approximately 90% accuracy on real-world scanned images.

Researchers are also exploring approaches that integrate neural networks with symbolic logic for better data efficiency and reasoning. Rastogi et al. [7] suggested a system that uses Formal Concept Analysis (FCA) for detecting document templates and a Knowledge Graph (KG) for learning rules. Their method uses a lattice-based similarity measure to handle structural variations and can learn extraction rules from just a single annotated example. Similarly, Sunder et al. [8] describe a "neuro-deductive" method where pre-trained neural networks extract basic facts from documents, and a meta-interpretive learner constructs executable logic programs. This approach shows excellent generalization from very few training samples, sometimes only one or two making it ideal for scenarios with limited annotated data.

3 Methodology

This proposed EDI system is designed as an end-to-end pipeline meant to combine different modules, such as document pre-processing, OCR, information extraction, and LLM-driven reasoning with a functional backend–frontend configuration. Experiments are done in a controlled environment: developing with Kaggle Notebooks, processing with a FastAPI backend, storing extracted data in MongoDB, and setting up a lightweight web interface for user interaction. This approach is divided into several clear phases of work; each of them added an important part to the system and helped build a full and complete document-intelligence solution.

3.1 Pre-processing and OCR pipeline for Scanned PDFs

For the receipts and PDFs which were scanned, PyMuPDF converted each page to an image. The system then performed a sequence of preprocessing operations designed to enhance text clarity: grayscale conversion to reduce the computational load while preserving most of the character details; Gaussian blur, which reduces speckle noise and thermal-print distortions; followed by adaptive thresholding, which emphasizes faint or unevenly illuminated text. Morphological operations such as dilation and erosion were applied to strengthen thin strokes and fix broken characters. A 2× resolution upscaling improved the recognition of small fonts seen in retail receipts. While these steps greatly improved clarity, extremely faded or poorly lit receipts remained difficult to read. The OCR extraction was performed using EasyOCR, whose CRAFT-based detection and deep-learning recognition pipeline generates bounding boxes, confidence values, and text tokens for every page.

These greatly enhanced contrast and clarity but could not resolve distortions in the varied conditions of ink fade and lighting.

Fig 1: WholeFoods Preprocessed receipt **Fig 2**: EasyOCR json output for receipt

3.1.1 OCR Method Comparison

The system benchmarked Tesseract and EasyOCR on low-quality, real-world receipts to identify the most reliable OCR engine. Tesseract frequently failed on faded thermal prints, broken characters, symbol misinterpretations, and cluttered backgrounds. The efficiency was low because it required pre-cropped text regions for good detection. EasyOCR showed far stronger token accuracy, better robustness to uncontrolled lighting, and more reliable character boundary detection by its built-in CRAFT detector. More importantly, EasyOCR tended to handle stylized and handwritten text more elegantly and embraced GPU acceleration for faster processing. Since EasyOCR outperforms in terms of token accuracy, robustness, and qualitative check, it has been selected as the main OCR engine.

Table 1: OCR Performance Comparison on Retail Receipt

Metric	Tesseract Output	EasyOCR Output
Number of correct tokens	43/105	87/105
Percentage accuracy (token-level)	40.95%	82.85%
Speed	Faster on CPU	Faster on GPU
Language/Font Training	Custom training possible but complex	Custom training easier and well-documented
Text Detection	Weak; requires pre-cropped text regions, fails on cluttered backgrounds	Built-in CRAFT-based text detection; robust on street photos, surfaces, and noisy documents

Handwritten Text	Very weak	Better performance, though not perfect

3.2 Token Cleaning, Line Grouping, and Column Structuring

The raw OCR output had duplicate tokens, overlapping bounding boxes, split characters, and low-confidence segments. A custom cleaning module removed duplicate coordinates, discarded very low-confidence tokens, normalized inconsistencies, and merged fragmented characters. After cleaning, the system grouped tokens into lines based on their vertical alignment, since a fixed tolerance allows tokens belonging to the same horizontal region to be merged into coherent text lines. In order to distinguish product descriptions on the left from price values on the right, column grouping was applied using x-coordinate clustering. This step was necessary because receipt layouts depend heavily on left–right alignment. The final grouped lines were serialized for later extraction tasks.

3.3 Information Extraction from Grouped OCR Lines

Once this text was segmented into meaningful lines and columns, a rule-based extraction layer turned these lines into structured fields 1. Regular expressions, keyword patterns, and positional heuristics extracted store name, purchase date, taxes, subtotal amount, total amount, and complete item lists. Identification of merchants was further informed by filename cues, detection of uppercase patterns, and title-case analysis. Product-line extraction included the logic to skip totals or summary rows mistaken for an item price, so only those valid product entries would be captured. These structural cues combined with rule-based reasoning enabled the system to reliably extract from irregular receipt layouts as well.

3.4 Identification of OCR Quality Issues and Pipeline Refinements

During experimentation, some common OCR issues that came up were misrecognition of symbols such as “$”, “%”, “/”, and “:”, confusion of similar-looking characters such as O/0 and 1/I, and missing product lines due to low contrast. Narrow bounding boxes sometimes result in unnatural word splits. Addressing these issues, refinements to the pipeline involved enhancing rendering DPI to 300, adjusting contrast, improving thresholding parameters, eliminating very low-confidence OCR outputs, and tuning EasyOCR detection thresholds. These refinements together improved token accuracy while reducing line-grouping errors across a variety of receipt formats.

3.5 Backend Development Using FastAPI

For the proposed project, a FastAPI-based backend was built in order to support scalable document processing. For the */upload_scanned* endpoint, the entire workflow for the scanned documents is implemented. The workflow includes the PDF page rendering and the image preprocessing processes. Password generation from the documents is followed by password cleaning. Then the line formation and the rule extraction of the merchant information, the totals, and the items are done. The temporary storage of the entire output is done in the memory. Each document is given a unique identifier called *doc_id* for future reference.

3.6 Native PDF and DOCX Text Extraction

Digitally generated documents, on the other hand, are processed through a different pipeline, which helps maintain structure. The “*/upload_native*” endpoint uses PyMuPDF for text extraction from native PDF documents, while text extraction from DOCX documents uses the “python-docx” library. These extracted text

paragraphs are cleaned, merged, and then passed through a Large Language Model (LLM) mechanism, which assists in generating summaries, bullet points, and semantic interpretations. Raw text as well as LLM outputs are all stored for further analysis.

3.7 LLM-Based Question–Answering for Scanned Receipts

The question answering function is facilitated via the */ask* path to allow for dynamic querying based on processed receipts. With reference to a related *doc_id*, details on OCR patterns, key-value pairs, and a list of products obtained from receipts are generated. The well-structured question encourages an LLM to answer questions based only on the related patterns obtained from receipts. This results in a reduction in hallucination and helps to answer questions at an inference level.

3.8 Summarization Module for Native Documents

For native pdf and docx files, an LLM summary module is utilized to generate summarized versions as well as a segment-wise analysis to extract the relevant pieces of information and remove any unnecessary information present in the document. Thus, the semantic structure of a document containing ample information can be effectively analyzed.

3.9 Final API Architecture

Lastly, the complete backend architecture is made up of four major endpoints, including */upload_scanned* for the OCR process, */upload_native* for native documents, */ask* for the question-answering task, and */structured* for viewing the intermediate result collections. This kind of modular architecture allows for easy extensibility and enables a clear division of tasks of processing.

3.10 MongoDB Integration for Persistent Storage

MongoDB was integrated for storing all the processed documents. The DB, *"avansaber_information_extraction,"* includes all the documents, both scanned documents and original documents, identified through a distinct "*doc_id*" along with other details, "OCR results," "key-value pair" fields, "key-value pair" extractions, and "LLM summaries" generated. The *"/mongo_docs*" endpoint has been designed specifically for fetching "document history".

3.11 Frontend Integration

A light web interface has been developed using HTML, CSS, and JavaScript to create an interactive web platform. The frontend contains the functionality for uploading documents and displaying the results for merchant name extraction and the totals and lists in an image. It has a history page where it calls the previously stored documents in the MongoDB.

3.12 Text-to-Speech (TTS) Module for Audio Output

For enhanced accessibility, a text-to-speech module has been incorporated using the Google Text-to-Speech (gTTS) library. By using the */audio* endpoint, users can make an audio output of either the text obtained by OCR or the text summary produced by the LLM module. The module produces audio in the format of an MP3 file that can be directly played on the frontend interface.

3.13 System Evaluation and Testing

For evaluation, the system was tested with over twenty scanned receipt images and various types of natural documents, including legal papers and formal texts. Accuracy of extraction of merchants, dates, and amounts, robustness of line item extraction, coherence of summaries, and correctness of answers to answers were among the criteria for evaluation. Overall, through various modules, the system performed well, proving its effectiveness in its end-to-end solution.

4 Result and Discussion

A mixed dataset of scanned retail receipts, native PDFs, DOCX files, and handwritten entries were used to test the system. The assessments were made regarding the OCR accuracy, correctness of extracted fields, line–column grouping method stability, reasoning quality of the LLM, and overall speed of the entire system.

A. OCR Performance: EasyOCR and Tesseract were compared based low-quality text. On the other hand, Tesseract scored a mere 40.95% token-level accuracy and showed frequent symbol errors with heavy fragmentation on faded thermal receipts. EasyOCR performed much better, reaching 82.86% token-level accuracy with far fewer missing tokens and more reliable detection because of its CRAFT-based text-processing pipeline. Because of these reasons, EasyOCR was selected for all further stages.

B. Key–Value Extraction Accuracy: On 20 real-world receipts, this system achieved high accuracy across all major fields. The system could detect merchant names correctly in 90% of cases, dates in 85%, total amounts in 95%, and product lists in 80%. Most mistakes occurred on receipts that were extremely faded or with unusual layouts that would make grouping worse.

C. Native Document Summarization: In the generated document, the summarization ability was tested for its effectiveness in understanding and representing the meaning of the document. The summaries generated by LLM were found to retain 85-90% of the original content, reduced the document size by 65-75%, and maintained the logical flow of the document's content division.

D. Question and Answering Consistency: The */ask* component, which relied on the LLM for answering, was subjected to a test of over twenty questions per document. The component answered all binary, fact-based questions like "List all the products purchased", "How many items were bought?", "What is the price of [specific product]?" etc. with a correctness of 100%, recorded 95% correctness for questions concerning counting, and 90% correctness in questions that needed inferences, like the identification of store features or location cues.

E. System Performance: Overall, the system offered efficient processing capabilities for fast real-time applications. The time required for the extraction process for the scanning of receipts was 4.2 seconds, 1.1 seconds for native PDFs, 1.8 seconds for generating LLM summaries, and 0.8 seconds for question-answering queries. All these metrics ascertain the feasibility of this entire process for real-time functionality.

5 Challenges and Limitation

A. Limitations of EasyOCR on Extremely Low-Quality Receipts by EasyOCR. Extremely faded receipts, torn edges and motion blur reduced OCR accuracy. The text was either incomplete or missing. EasyOCR's performance was affected in the CRAFT-based detection process even for characters that are just visible.

B. Ambiguity in Product Line Detection : Irregular layout patterns of receipts sometimes resulted in anomalous token formation or a failure to map items to their respective prices based on a multi-line product name.

C. Variability in Handwritten Contents: The handwriting contents like writing, signatures, and scribbles have highly varying patterns resulting in fluctuating levels of recognition accuracy, thereby impacting the overall extraction.

D. Dependence on the OCR Accuracy of LLM Reasoning : The summarization module and the question-answering module using the LLM are dependent on the output from the OCR module entirely. This means that the accuracy of the modules depends on the accuracy of the OCR module output.

E. Need for Domain-Specific Fine-Tuning :The system uses general-purpose OCR and LLM models. Some other domains such as finance, healthcare, and legal documents may find the use of customized fine-tuning to be helpful and could be considered in future studies.

6 Conclusion

In this research, an extensive end-to-end solution for Document Intelligence has been provided. This solution can handle scanned PDFs, standard documentation, as well as unstructured text. For this, the solution applies state-of-the-art OCR, rule-based text extracting, and LLM-based inference. It performs well in practical applications such as digitizing receipts, summarizing text, and business automation. The solution also provides an end-to-end pipeline related to document intelligence, which can be used in research and industry purposes.

The future would include improvements in accuracy using domain-specific fine-tuning of OCR, incorporation of vision transformers for text detection, and extension for multilingual documents. Further improvements would include creating a larger dataset for evaluation, allowing customizations for LLM with few-shot learning, and utilizing advanced layout-aware models like LayoutLMv3 for complex document layout. Such additional ideas could further improve the scalability, robustness, and usability of the proposed system.

References

1. K. Jayaram and K. Sangeeta, "A Review: Information Extraction Techniques From Research Papers," *International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2017.
2. M. Skalický, Š. Šimsa, M. Uřičář, and M. Šulc, "Business Document Information Extraction: Towards Practical Benchmarks," *arXiv:2206.11229*, 2022.
3. T. A. Nguyen Dang and D. Nguyen Thanh, "End-to-End Information Extraction by Character-Level Embedding and Multi-Stage Attentional U-Net," *arXiv:2106.00952*, 2021.
4. U. İ. Avcı, B. Deveci, D. Goularas, and E. E. Korkmaz, "Information Extraction from Scanned Invoice Documents Using Deep Learning Methods," *2024 IEEE 13th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2024.
5. D. Vishwanath et al., "Deep Reader: Information extraction from Document images via relation extraction and Natural Language," *arXiv:1812.04377*, 2018.
6. N. Massarenti and G. Lazzarinetti, "A Deep Learning Based Methodology for Information Extraction from Documents in Robotic Process Automation," *CEUR Workshop Proceedings*, 2021.
7. M. Rastogi et al., "Information Extraction from Document Images via FCA based Template Detection and Knowledge Graph Rule Induction," *CVPR Workshops*, 2020.
8. V. Sunder, A. Srinivasan, L. Vig, G. Shroff, and R. Rahul, "One-shot Information Extraction from Document Images using Neuro-Deductive Program Synthesis," *arXiv:1906.02427*, 2019.