AUTOMATED GENERATION OF MARITIME DOCUMENTS USING ARTIFICIAL INTELLIGENCE AND OPTICAL CHARACTER RECOGNITION

G.R.Khanaghavalle, Sudarshan N S, Jainras Peter P
Assistant Professor, Student, Student
Department of Computer Science and Engineering
Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India

Abstract : Maritime freight operations process thousands of shipments daily, each of them possess a lot of documents which are crafted manually. This manual approach is time-intensive, highly complex due to cross-document data dependencies, and susceptible to human error. To address these problems, we propose an automated maritime document generation system that leverages a standardized JavaScript Object Notation (JSON) schema to define attribute mappings between input and output document structures. Text and data attributes are extracted from input documents — including PDFs and scanned images — using PDFMiner and Tesseract Optical Character Recognition (OCR) respectively. The acquired values are then mapped to the corresponding fields of target output documents using In-Context Learning with large language models (LLMs) based on the Generative Pre-trained Transformers (GPT) architecture, enabling the system to understand document context without the need for extensive retraining. The generated documents are validated against extraction logic and standardized maritime document formats to ensure accuracy and compliance. Experimental results of automated generation of House Bill of Lading (House B/L) from Master Bill of Lading (Master B/L) demonstrate that the proposed system significantly reduces document processing time and minimizes manual errors compared to traditional workflows, offering a scalable and reliable solution for automating maritime freight documentation.

IndexTerms - JavaScript Object Notation (JSON), Tesseract Optical Character Recognition (OCR), Large Language Models (LLMs), House Bill of Lading (House B/L), Master Bill of Lading (Master B/L)

I. INTRODUCTION

1.1 Overview of the Industry

The maritime freight industry involves four key stakeholders — the Shipper (also known as the Consignor), the Liner (also known as the Carrier), and the Freight Forwarder. The Shipper is the entity seeking to transport goods from one country to another. The Liner is responsible for physically transporting the shipment from the specified origin to its destination. To determine the rates for moving a shipment, the Shipper engages a Freight Forwarder, who acts as a broker by sourcing contract rates from Liners and spot rates from online platforms. Once the rate, route, and related logistics are finalized, the goods are consolidated as a shipment and loaded onto a vessel [1].

Upon loading, the Liner drafts the Master B/L, a critical document containing shipment details such as the mode of transport, origin, destination, parties involved, package information, and applicable charges, among other details. This document is handed to the Freight Forwarder, who must manually draft the House Bill of Lading House B/L and certain other documents [2] by consolidating information from the Master B/L and the Shipper's details. With respect to ocean freight, shipments are transported in two modes: Full-Container Load (FCL), where a single shipper's goods occupy an entire container, and Less-than-Container Load (LCL), where multiple shippers share a single container.

1.2 Current Challenges

Manually crafting the House B/L from the Master B/L and Shipper details is a time-intensive and complex process due to cross-document data dependencies and is highly susceptible to human error, especially in dynamic and uncertain situations of global disruptions such as COVID-19 pandemic [3]. In the case of LCL shipments, a separate House B/L must be generated for each individual Shipper within the same container, making the process highly inefficient and repetitive at scale.

1.3 Proposed Work

To address these challenges, we examined the attributes of both the Master B/L and House B/L to construct a structured JavaScript Object Notation (JSON) representation for each document type. This structured representation enables Large Language Models (LLMs) to accurately map raw text extracted from source documents to the relevant attributes, significantly improving the efficiency and reliability of the generation process. Unlike prior approaches that rely solely on rule-based extraction or template matching, the proposed system combines OCR-based extraction with in-context learning via LLMs, allowing the system to handle varied document layouts without extensive retraining.

The primary objective of this work is to extract information from maritime documents as raw text and map it to a predefined JSON structure using LLMs, followed by rule-based validation and structured document generation. To this end, we propose the following framework:

1. Extraction of information as raw text from images and PDFs using Optical Character Recognition (OCR) tools.

2. Mapping of the extracted raw text to the attributes of the JSON structures defined for the Master B/L and House B/L using Large Language Models (LLMs).

3. Validation of each mapped attribute value using simple rule-based logic to ensure correctness and completeness.

4. Binding of the validated structured representation with an HyperText Markup Language (HTML) template, which is subsequently converted to PDF, yielding the final output document.

II. RELATED WORK

Recent advances in document intelligence have significantly improved structured information extraction from visually rich and semi-structured documents. Existing studies mainly focus on layout-aware extraction, OCR optimization, and LLM-based structured understanding. However, freight documents introduce additional challenges such as logistics-specific terminology, mixed tabular structures, and operational dependencies that require not only extraction but also downstream transformation.

2.1 Layout-Aware Document Understanding

Document understanding has evolved from rule-based extraction to layout-aware neural architectures that integrate textual and spatial relationships, as a model. A lightweight multigranular graph representation capable of capturing document structure at word, region, and page levels while maintaining low computational complexity, was proposed by LMGDoc [4]. Similarly, the graph neural representations were integrated with pretrained language models to enhance the process of information extraction by modeling the directional and positional relationships between the elements of the document [5].

These approaches demonstrate strong strides of development in understanding visually rich documents but they are primarily designed for extraction of key attributes rather than document transformation workflows.

2.2 OCR and Region Extraction Techniques

Robust text extraction before semantic interpretation is crucial for reliable document automation. Segmentation-less extraction methods improve OCR efficiency by separating text and non-text regions through partial decompression strategies, reducing unnecessary computational overhead [6].

Page-level OCR services for camera-captured and scanned documents further improve text recovery under varying document quality conditions, in the case of practical OCR deployment [7]. These techniques are essential as freight documents often contain signatures, stamps, tabular cargo sections, and irregular layouts.

2.3 Large Language Model-Based Structured Extraction

Large language models have recently improved structured attribute extraction from complex domain documents. A legal-document extraction study demonstrates that schema-guided LLM prompting technique allows fine-grained attribute generation from unstructured text using weak supervision and few-shot learning [8].

Broader document intelligence research shows that modern document systems increasingly combine OCR, layout analysis, schema prompting, and retrieval-based reasoning to generate structured outputs from heterogeneous documents [9].
Recent end-to-end document extraction methods involve direct mapping of document content into structured outputs, which reduces dependencies on manually separated pipeline stages [10], while multimodal large language models extend this capability to complex archival and multi-layout documents [11].

2.4 Automation in Logistics and Freight Documents

Although document extraction research has advanced significantly, logistics document automation remains comparatively less explored. Robotic process automation has been applied in logistics service workflows to reduce repetitive manual processing, demonstrating efficiency gains in operational document handling [12].

Freight forwarding studies further emphasize that document consistency, process accuracy, and service quality directly influence logistics efficiency [1] [2]. However, these studies focus mainly on operational process analysis rather than AI-driven document transformation.

2.5 Research Gap

Existing literature demonstrates strong progress in OCR, layout-aware extraction, and LLM-based structured generation. However, limited work integrates these capabilities into a freight-specific automation pipeline that supports extraction, transformation, validation, and target document generation.
The proposed system addresses this gap by integrating OCR-based text acquisition, schema-guided AI extraction, freight field mapping, validation logic, and PDF generation in a unified freight automation workflow.

III. SYSTEM ARCHITECTURE

The proposed architecture is designed as a generalized freight document automation platform capable of processing multiple logistics documents through a consolidated extraction and transformation pipeline. The current implementation validates the architecture using Master Bill of Lading (Master B/L) to House Bill of Lading (House B/L) conversion as the primary operational use case. The architecture integrates modular services that separately manage document ingestion, extraction, transformation, validation, storage, and output generation, enabling future scalability for multiple freight document categories.

3.1 Frontend Interface and User Interaction

The frontend interface acts as the entry point allowing freight forwarders and other agents of the industry to select the input and output documents along with the preferred AI model. Users upload a freight document, which is considered as the input document, through a web-based interface, where supported formats include scanned images and PDF documents. In addition to document upload, the frontend supports document preview, editing, and final download operations. This layer communicates with the backend through RESTful APIs to initiate extraction and generation workflows.

3.2 Backend and Application Programmable Interface

The backend Application Programmable Interface (API) layer coordinates communication between all internal services. After receiving uploaded files from the frontend, the backend performs request validation, authentication handling, and routing to the document extraction service. This orchestration layer also manages freight entity mapping, shipment data association, and final document generation requests. By separating orchestration logic from extraction and rendering modules, the system maintains modularity and service independence.

3.3 Document Extraction Layer

The document extraction service is responsible for converting freight documents into machine-readable structured data. Initially, uploaded files are analyzed to determine whether they are text-native PDFs or scanned image documents.

3.3.1 OCR Processing

For scanned freight documents, image preprocessing and PDF-to-image conversion are performed before applying optical character recognition. Tesseract OCR is used to extract textual content from image-based freight documents.

3.3.2 Schema-based Extraction using Artificial Intelligence

Once raw text is obtained, schema-guided language model extraction is performed. A predefined (JavaScript Object Notation) JSON schema representing freight document fields is combined with extracted text, enabling structured key-value generation. This process converts unstructured freight content into machine-readable JSON output suitable for downstream processing.

3.4 Transformation and Validation Layer

The transformation engine converts extracted freight entities into normalized target document structures using predefined business rules. In the current implementation, extracted Master B/L fields are mapped into House B/L document structures.

The validation layer then verifies generated fields through mandatory field checks, consistency verification, and business-rule compliance. This stage prevents incomplete or invalid freight data from propagating into final document generation.

3.5 Data Persistence and Document Generation

Validated freight data is stored in the database layer for persistence and future retrieval. Structured records are maintained to support preview updates, editing workflows, and repeated generation requests.

3.5.1 Template Binding:

The validated structured data is binded with the respective HyperText Markup Language (HTML) Template with styling provided by Tailwind Cascading Style Sheets (CSS).

3.5.2 PDF Conversion:

The template binded with the values from each attribute is converted into PDF using Playwright library which is used to launch a headless browser powered by Chromium Engine.

3.5.3 Download Process:

The PDF document generated can be downloaded as an individual file or as a .zip file in the case of multiple documents.

3.6 Extensibility of Architecture

Although the current implementation demonstrates Master B/L to House B/L conversion, the architecture is intentionally designed for future freight document expansion. By reusing schema definitions, extraction logic, and transformation modules, the same pipeline can support invoices, packing lists, delivery orders, shipping instructions, and additional logistics documents without major architectural modification.

IV. METHODOLOGY

The proposed methodology follows a modular freight document processing workflow designed to convert semi-structured logistics documents into validated machine-readable outputs and generate target freight documents through schema-driven transformation. The workflow consists of document acquisition, extraction, structured field generation, validation, model comparison, and output generation stages. The current implementation validates the methodology where Master B/L is the source document and the House B/L is the target document.

4.1 Source Document Acquisition and Dataset Preparation

The experimental dataset consists of freight document samples collected in PDF format, including both text-native and scanned freight documents. These documents contain logistics entities such as shipper details, consignee information, vessel details, port information, container identifiers, and cargo descriptions.

To evaluate extraction robustness, the dataset includes:

- Digitally generated freight documents as PDFs
- Scanned freight documents as images
- Mixed-layout form of Freight documents

Documents were selected to represent practical freight forwarding document variability, including differences in text density, alignment, and field placement. Since freight documents often contain domain-specific terminology, preprocessing is necessary before structured extraction.

4.2 Extraction Workflow

The extraction workflow is designed to process heterogeneous freight documents through a sequential pipeline that adapts dynamically according to document format and content characteristics. Since freight documents may originate as digitally generated PDFs, scanned copies, or mixed-layout images, an initial classification stage determines the most suitable extraction path before structured interpretation begins.

At the time of upload, the backend first performs file-type verification and metadata inspection to identify whether the document contains embedded machine-readable text or rasterized page images. Based on this classification, the system routes the document either to direct text extraction or OCR-based extraction.

4.2.1 PDF Text Extraction

For digitally generated freight PDFs, direct text parsing is applied to extract embedded textual content without introducing OCR overhead. This approach preserves the original textual fidelity because character positions, spacing, and document ordering remain available in machine-readable form. The extraction module processes page-level content by reading each PDF object and aggregating text blocks into a unified text stream while preserving logical reading order. Since freight documents frequently contain multiple aligned fields such as consignee details, cargo descriptions, and vessel information, text normalization is applied after extraction to remove formatting inconsistencies, extra line breaks, and duplicated spaces. Direct extraction improves processing speed and minimizes character recognition errors that typically arise in OCR pipelines. It is particularly effective when source freight documents are generated from digital freight systems or exported from shipping software platforms.

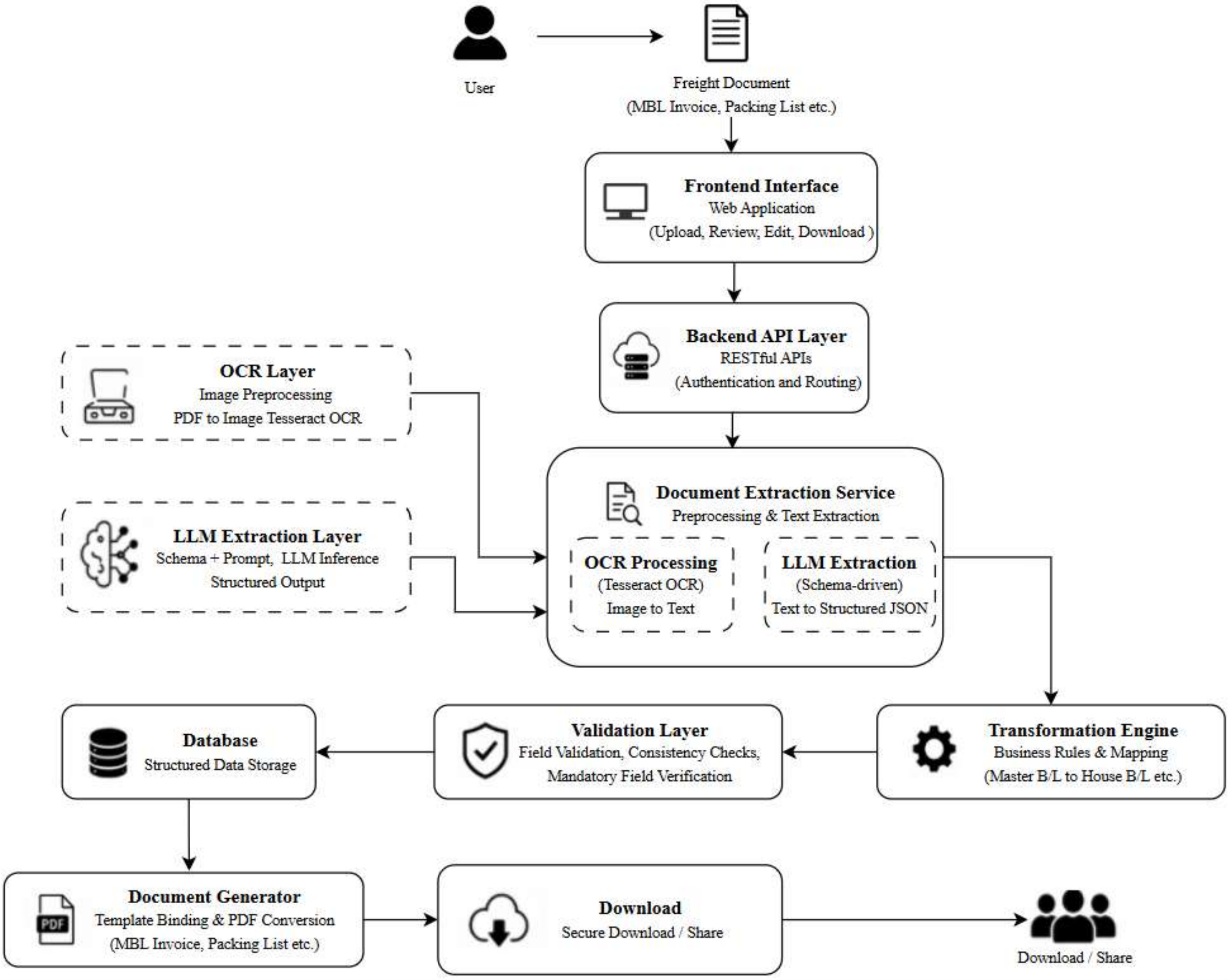

Fig. 1. Overall Architecture of the Proposed Freight Document Automation Platform

4.2.2 OCR-based Extraction

For scanned freight documents, this method is used because text exists only as rasterized visual content. Initially, scanned PDF pages are converted into high-resolution images to improve recognition accuracy. Image preprocessing operations are then applied before OCR execution, including grayscale conversion, contrast enhancement, and noise reduction. These preprocessing stages improve text visibility in documents containing stamps, faded print, or scanning artifacts. After preprocessing, optical character recognition is performed to recover textual content from page images. Since freight documents often contain irregular alignments, tabular cargo sections, signatures, and handwritten markings, OCR output is further normalized to reduce fragmented text sequences and line-order inconsistencies. The extracted OCR text is then consolidated into structured text blocks suitable for downstream schema-based processing. Although OCR introduces slightly higher computational overhead compared to direct PDF extraction, it enables the system to process historical freight documents and scanned operational records.

4.2.3 Schema-based Extraction using AI

Once textual content is obtained through either PDF text extraction or OCR-based extraction, the normalized text is supplied to the schema-based AI extraction layer. In this stage, a predefined freight schema is constructed to define all mandatory target entities required for structured freight processing. The schema specifies freight-related fields including:

- Shipper Details
- Consignee Details
- Notify Party
- Vessel Information
- Voyage Number
- Container Identifiers
- Port of Loading
- Port of Discharge
- Freight Terms

The extracted document text and schema are jointly provided to the language model through structured prompting. Instead of allowing unrestricted text generation, the schema constrains the model to produce a predictable JSON-style output where each extracted value is mapped to a corresponding freight field.

This schema-guided approach improves extraction consistency by reducing semantic drift and preventing omission of critical freight entities. It also improves adaptability because the same extraction logic can be reused for future freight document categories by simply modifying schema definitions.

The complete extraction and generation sequence is illustrated (see Fig. 2), where both direct extraction and OCR-based extraction converge into the same structured AI extraction pipeline.

4.3 Artificial Intelligence Model Configuration

Multiple language model configurations were evaluated to determine the most reliable approach for structured freight field extraction under practical deployment constraints. Since freight document automation requires balancing extraction accuracy, inference speed, and deployment cost, the system compares cloud-based and local inference strategies under identical schema-guided input conditions.

4.3.1 Cloud-based Inference Models:

These types of models provide high extraction accuracy due to access to large-scale pretrained language architectures with stronger reasoning capabilities. In this configuration, extracted freight text and schema definitions are transmitted to a remote inference endpoint where structured field generation is performed. Cloud-based models generally produce more consistent freight entity recognition, particularly when source documents contain ambiguous field ordering or incomplete layout consistency. These models also demonstrate stronger contextual interpretation when freight terms appear in variable document positions. However, cloud inference introduces network dependency and operational cost, which may affect deployment scalability in high-volume document environments.

4.3.2 Local Inference Models:

These models execute directly within the system environment without external API dependency. This configuration improves deployment control and data privacy because freight document content remains within internal infrastructure. Local models are particularly useful when freight documents contain sensitive shipment data requiring restricted external exposure. Although local models reduce external dependency, extraction consistency may vary when handling highly irregular freight layouts because smaller local models typically possess reduced contextual reasoning capability compared to large cloud-hosted models

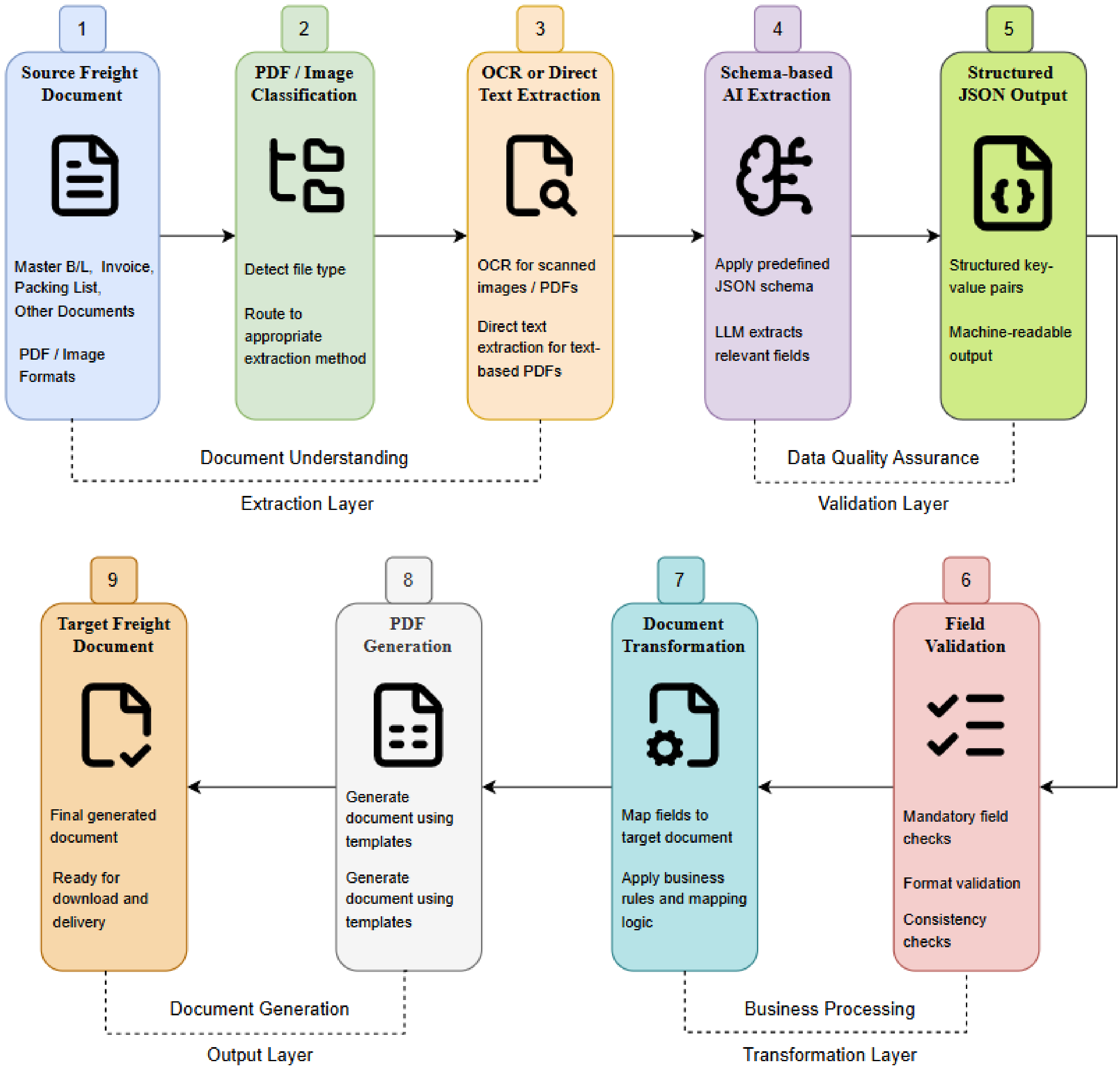

Fig. 2. Extraction and Generation Pipeline for Freight Document Automation

Lightweight Structured Extraction Models: These sets of models are evaluated for scenarios where rapid inference is required with limited computational resources. These models prioritize response speed and lower memory usage while still producing schema-aligned structured outputs. Lightweight extraction is useful in real-time freight operations where rapid preliminary extraction is preferred before manual review. However, lightweight models may show reduced performance when freight documents contain complex tabular sections or multi-page field dependencies.

4.4 Freight Field Validation

After schema-based extraction, generated freight entities undergo a validation stage before they are accepted for downstream transformation. The purpose of validation is to ensure that extracted outputs satisfy predefined freight schema constraints and preserve operational consistency across target freight documents. The validation layer acts as an intermediate control stage between extraction and document generation, preventing incomplete or structurally inconsistent freight records from propagating into final output documents.

The validation process consists of four primary checks:

- Mandatory field existence, where essential freight entities such as shipper, consignee, and container identifiers must be present.
- Empty-field detection, where extracted fields are examined for null or blank values.
- Type verification, where field values are checked against expected data formats such as text, numeric values, or date structures.
- Mapping consistency, where source freight entities are verified against expected target schema definitions.

A freight field is considered valid only when all applicable validation conditions are satisfied. To quantify validation completeness, a validation score is computed as follows:

$$V = Nv / Nt \qquad (1)$$

Where V is the validation score, Nv = number of validated fields and Nt = total expected freight fields

A validation score close to 1 indicates that most extracted freight entities satisfy schema constraints and are suitable for downstream document transformation. This metric is particularly useful when comparing extraction outputs across different AI models, as it reveals whether structurally complete extraction has been achieved even before semantic accuracy is measured.

The comparative model performance is summarized in **Table 1**, where extraction accuracy, response time and deployment suitability are analyzed.

Table 1. Comparative Analysis of Model Performance

Model	**Accuracy (%)**	**Response Time**	**Deployment**
GPT-OSS-120B	94	Medium	Cloud
Gemini	95	Fast	Cloud
Groq	96	Fast	Cloud

Mistral	91	Medium	Local
Ollama	87	Medium	Local

.

4.5 Extraction Accuracy Measurement

After validation, extraction accuracy is measured by comparing generated freight fields with manually verified reference values obtained from the source freight document. Accuracy evaluation focuses on field-level correctness rather than sentence-level similarity because freight automation requires exact operational values for downstream document generation. A field is considered correct when the extracted value matches the expected freight entity without any semantic distortion. Minor formatting differences may be normalized before comparison if they do not alter any meaning related to the industry.

The extraction accuracy is calculated using:

$$Accuracy = N_c / N_t \times 100 \qquad (2)$$

Where Nc is the number of correctly extracted fields and Nt is the total number of evaluated fields

This metric expresses extraction reliability as a percentage across all freight entities. For example, if 24 out of 25 freight fields are correctly extracted, the resulting accuracy is 96%. Accuracy measurement is applied across multiple freight documents to evaluate extraction consistency under variation of document quality, layout complexity, and source format.

4.6 Transformation Logic

After validation and accuracy confirmation, structured freight entities are transformed into target document fields through rule-based mapping. The transformation engine converts extracted source freight values into target freight schema representations using predefined mapping rules. In the current implementation, Master B/L fields are transformed into a structure which resembles the attributes of House B/L.

Transformation logic includes both direct mapping and conditional mapping. Direct mapping is used when source and target freight fields are semantically identical. Conditional mapping is applied when source freight entities require business-rule interpretation before mapping.

This design allows the same transformation engine to be reused for future freight document categories such as:

- Invoices
- Packing lists
- Delivery orders

The transformation stage therefore serves as the central reusable layer of freight document automation. An example of mapping freight fields with respective source document and target document attributes, along with the mapping rule and validation constraints, is given in Table 2.

Table 2. Instance of Freight Field Mapping

Source Document Fields	**Target Document Fields**	**Mapping Rule**	**Validation Constraint**
Shipper Name	Shipper Name	Direct Mapping	Mandatory

Consignee Name	Consignee Name	Conditional Mapping	Mandatory
Container Number	Container Number	Direct Mapping	Alphanumeric Check
Port of Loading / Discharge	Port of Loading / Discharge	Direct Mapping	Standard Port Format
Gross Weight	Gross Weight	Numeric Normalization	Numeric Validation

4.7 Portable Document Format Generation

Once transformed freight data passes validation, the final structured output is rendered into a target freight document using predefined PDF templates. The PDF generation stage converts structured freight attributes into layout-consistent operational documents suitable for business use.

The generation process includes:

1. Field Injection: The validated attributes from the structured JSON object of the target document are inserted as values using Angular Server Side Rendering (SSR), in the predefined HTML Template.
2. Layout Formatting: The fields are aligned along with spacing according to the freight document standards using Tailwind CSS.
3. HTML to PDF Conversion: The template binded with values is converted into PDF when Playwright library initialises the headless browser with Chromium Engine. The function of printing the template as PDF is invoked which retains the layout and styling as well.
4. Download Preparation: The PDF document which is generated is available for users as an individual file. Multiple documents can also be downloaded as a .zip file.

This generation layer ensures that extracted freight intelligence becomes directly usable in operational workflows rather than remaining as raw structured data. The current implementation demonstrates House B/L generation, but the same rendering framework can support multiple freight document templates.

V. RESULTS

The proposed freight document automation framework was evaluated using multiple freight document samples representing both digitally generated PDFs and scanned freight documents. The experiments focused on measuring extraction consistency, validation completeness, and structured field generation performance across different AI model configurations.

The extraction workflow successfully processed both direct-text and OCR-based freight documents. For text-native PDFs, direct extraction produced faster response time because textual content was directly available in machine-readable form. In scanned freight documents, OCR preprocessing increased processing time but enabled reliable recovery of freight entities required for schema-based extraction.

Structured extraction generated freight attributes including shipper details, consignee information, container identifiers, vessel details, and freight terms with high consistency across repeated test cases. The generated outputs were then evaluated using field-level correctness and validation score metrics described in Section IV.

5.1 Model Performance Comparison

Different AI models were evaluated using identical freight document inputs and schema definitions. The comparative analysis shows that cloud-based models produced higher extraction consistency when documents contained irregular layout ordering or partially fragmented freight text.

The observed extraction accuracy trend is illustrated in Fig. 3.

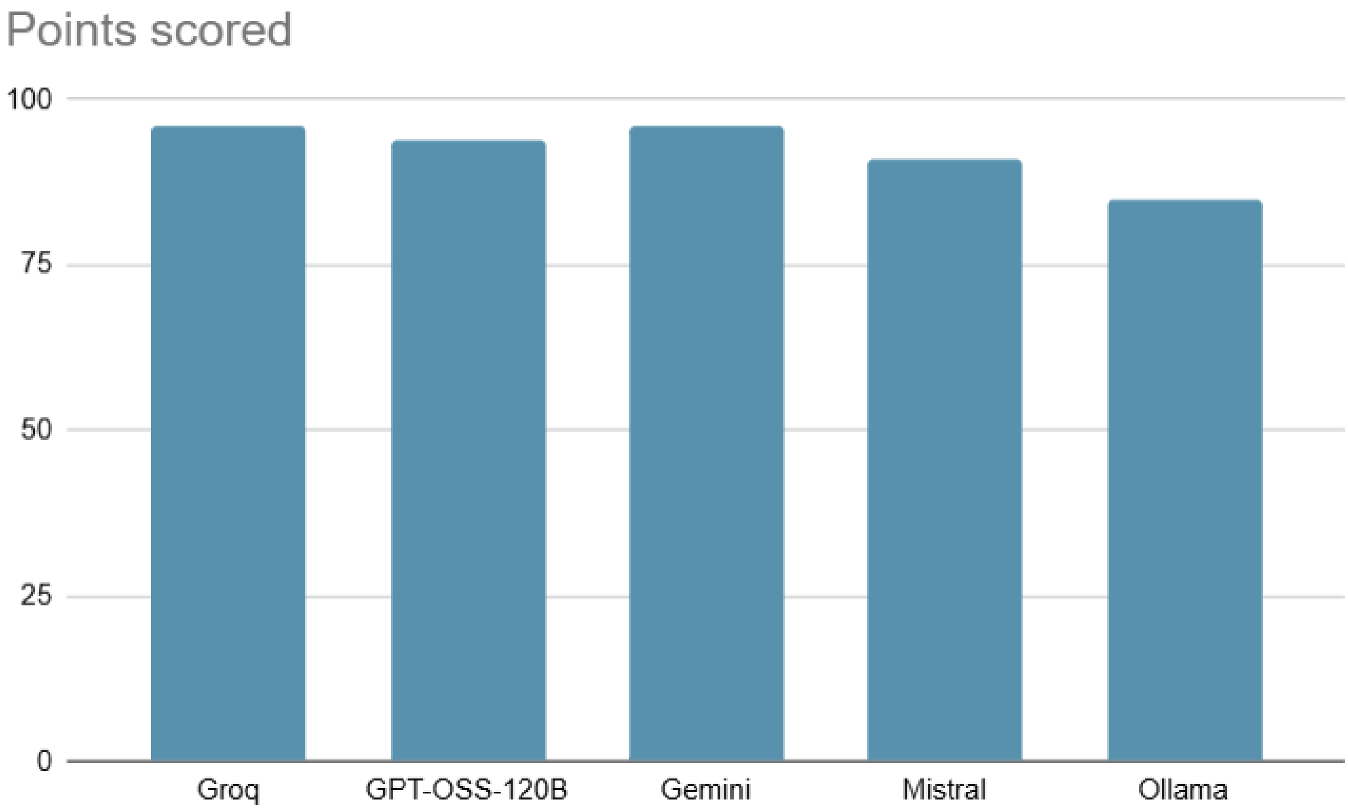

Fig. 3. Extraction accuracy comparison across evaluated AI models

Local inference models provided acceptable performance while improving deployment control and reducing external dependency.

5.2 Validation Outcome

Validation results indicate that schema-guided extraction improves structural integrity before transformation. Mandatory freight entities were consistently identified in most test documents, while detection of missing fields reduced incomplete output propagation into final generation. Higher validation scores were observed when extracted text originated from digitally generated PDFs compared to scanned freight documents, because OCR noise occasionally affected field completeness.

VI. Discussion

The experimental results demonstrate that freight document automation benefits significantly from combining schema-guided extraction with validation-based transformation. One major observation is that schema-driven prompting reduces field ambiguity when compared to unrestricted extraction. Freight entities with predictable business semantics such as shipper, consignee, and port information were extracted more consistently as the model was constrained by predefined field expectations.

The comparison between OCR-based and direct PDF extraction also highlights that document quality remains a major factor affecting extraction reliability. Although OCR enables broader document compatibility, scanned freight documents remain more sensitive to alignment distortions, stamp interference, and partial text fragmentation.

Another important observation is that validation logic acts as a critical control mechanism. Even when extraction generates semantically plausible values, schema validation prevents structurally incomplete records from reaching downstream document generation.

The transformation layer further demonstrates that freight document automation is not limited to one document pair. Although the current implementation uses Master B/L to House B/L conversion, the same logic can support broader freight workflows by extending schema definitions and mapping rules.

VII. CONCLUSION

This paper presents a modular freight document automation framework that combines OCR-based extraction, schema-guided AI processing, validation, and structured document generation within a unified workflow. The proposed system demonstrates that freight documents can be transformed into machine-readable structured outputs and converted into target freight documents through reusable extraction and mapping logic.

The current implementation validates the framework using Master Bill of Lading (Master B/L) to House Bill of Lading (House B/L) conversion, while maintaining architectural flexibility for additional freight document categories. Experimental results confirm that schema-guided extraction improves field consistency and that validation logic enhances structural reliability before document generation.

Future work will focus on expanding the framework toward multi-document freight automation, improving multi-page extraction capability, and integrating domain-adaptive model optimization for higher freight-specific extraction accuracy.

VIII. REFERENCES

[1] C. Popa and I. Goia, "The Analysis of Freight Forwarding Services Using the Business Process Modelling Tools," Faculty of Navigation and Naval Management, Naval Academy "Mircea cel Batran", Constanta, Romania, 2022.

[2] S. Subhashini and S. Preetha, "An Empirical Analysis of Service Quality Factors Pertaining to Ocean Freight Forwarding Services," Maritime Business Review, vol. 3

[3] A. M. Sharma, D. Batra, and S. Sharma, "Documentation in logistics sustainability – challenges and opportunities," in Supply Chain Management, 1st edition, 2024

[4] A. Sassioui, Y. Elouargui, M. El Kamili, R. Benouini, E. M. Benyoussef, M. Chergui, and M. Ouzzif, "LMGDoc: Light Multigranular GNN for Efficient Document Understanding," IEEE Access, vol. 13

[5] D. Wang, Z. Ma, A. Nourbakhsh, K. Gu, and S. Shah, "DocGraphLM: Documental Graph Language Model for Information Extraction," in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), Taipei, Taiwan, Jul. 23–27, 2023

[6] T. Bisen, M. Javed, P. Nagabhushan, and O. Watanabe, "Segmentation-Less Extraction of Text and Non-Text Regions From JPEG 2000 Compressed Document Images Through Partial and Intelligent Decompression," IEEE Access, vol. 11

[7] J. Park, W. Kang, S. Park, K. Lee, H. I. Koo, and N. I. Cho, "Development of OCR Service for Page-Level Recognition for Camera-Captured Document Images," IEEE Access, vol. 13

[8] S. Adhikary, P. Sen, D. Roy, and K. Ghosh, "A Case Study for Automated Attribute Extraction from Legal Documents Using Large Language Models," Artificial Intelligence and Law, 2024

[9] W. Ke, Y. Zheng, Y. Li, H. Xu, D. Nie, P. Wang, and Y. He, "Large Language Models in Document Intelligence: A Comprehensive Survey, Recent Advances, Challenges, and Future Trends," ACM Transactions on Information Systems, vol. 44

[10] T. Wu, Y. Chen, and S. Wang, "End-to-End Information Extraction from Document Images," IEEE Transactions on Image Processing, 2024

[11] M. Vafaie, S. Hertling, I. Banse-Strobel, K. Dubout, and H. Sack, "End-to-end Information Extraction from Archival Records with Multimodal Large Language Models," in Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), Seoul, Republic of Korea, Nov. 2025

[12] V. Gružauskas and D. Ragavan, "Robotic Process Automation for Document Processing: A Case Study of a Logistics Service Provider," Vadyba / Journal of Management, vol. 36