International Journal on Advanced Computer Theory and Engineering
ISSN: 2319 - 2526
Volume 15 Issue 2s (2026) | Pages: 373- 377

AI-Powered Automated Invoice Processing System with End-to-End Email Integration, OCR Extraction, and Real-Time Dashboard Monitoring

Sukanya Bhosale[1], Sanket Dawane[2], Dnynesh Jadhav[3], Aditi Gadilkar[4]

[1234]Department of Artificial Intelligence and Machine Learning, G.S.Moze College of Engineering Pune, Maharashtra, India

Peer Review Information

Type: *Article*
Received: *13 February 2026*
Revised: *14 March 2026*
Accepted: *15 April 2026*
Published: *18 May 2026*

Abstract

This paper presents the design, implementation, and experimental evaluation of a fully integrated AI-powered Automated Invoice Processing System that extends prior workflow-stage research into a complete, production-ready solution. Building upon our Phase 1 prototype — which demonstrated the viability of using Large Language Models (LLMs) and Optical Character Recognition (OCR) for structured invoice understanding — this final system introduces end-to-end email integration via Microsoft Outlook, intelligent attachment classification, SQL-based centralized data storage, and a real-time React.js monitoring dashboard. The system autonomously fetches invoice emails, classifies attachments (PDF, image, plain text), extracts key invoice fields using a hybrid OCR-LLM pipeline, validates extracted data through a rule-based correction engine, and logs all records into a structured SQL tracker. Experimental results across 510 diverse invoice samples demonstrate a field extraction accuracy of 97.4%, total computation accuracy of 99.1%, and an average processing time of 1.8 seconds per invoice. The system achieves superior performance compared to traditional OCR tools, Layout LM-based models, and our own Phase 1 AI Invoice Builder baseline, while also supporting multilingual processing, irregular layout handling, and privacy-preserving hybrid cloud deployment.

Keywords: Invoice Automation; OCR; Large Language Models; Email Processing; SQL Database; React.js Dashboard; NLP; Hybrid Cloud; Document Understanding; Validation Engine.

How to Cite This Article

Bhosale, S., Dawane, S., Jadhav, D., Gadilkar, A., (2026). AI-Powered Automated Invoice Processing System with End-to-End Email Integration, OCR Extraction, and Real-Time Dashboard Monitoring. *International Journal on Advanced Computer Theory and Engineering*, 15(2s), 373-377 .

Introduction

Invoice processing is a critical yet labor-intensive function in financial operations across industries. Organizations receive hundreds to thousands of invoices monthly through email, each with varying layouts, formats, and languages. Manual handling of these invoices is not only time-consuming but also highly error-prone, resulting in mismatched totals, delayed payments, and audit failures. In our first-semester research (Phase 1), we proposed and validated a conceptual pipeline for an AI-based Invoice Builder that integrated OCR with LLM reasoning and a rule-based validation engine. While Phase 1 demonstrated the feasibility of automated invoice field extraction and validation with projected accuracies of 95% extraction and 98% total computation accuracy, it remained a workflow-level prototype without full email integration, database persistence, or a user monitoring interface.

This paper presents the complete final system — an end-to-end Automated Invoice Processing System that operationalizes the Phase 1 vision. The proposed system connects directly to Microsoft Outlook to fetch invoice emails, classifies and processes attachments using OCR and LLM reasoning, stores structured data in a SQL database, and exposes a React.js dashboard for real-time tracking, filtering, and review. Experimental evaluation on 510 real invoice samples confirms significant improvements across all key performance metrics compared to existing approaches and our own baseline. The remainder of this paper is organized as follows: Section II surveys related work; Section III describes the methodology and system architecture; Section IV details the proposed system components; Section V presents results and analysis; Section VI concludes with future directions.

Literature review

Template-Based OCR Systems: Early invoice automation systems such as those using Tesseract OCR and ABBYY FineReader depended on fixed layout templates. While achieving 78-85% accuracy on known formats, these systems fail on unseen layouts and require significant manual configuration per invoice type [1, 3].

Deep Learning for Document Understanding: Krieger et al. [5] evaluated Chargrid, Random Forest, and LayoutLM for structured data extraction from invoice documents. LayoutLM achieved the highest accuracy by capturing spatial and semantic relationships within documents, making it particularly effective for varied invoice templates.

NLP-Based Field Extraction: Jenifer [2] proposed an NLP-driven approach combining OCR with pattern matching and rule-based logic for extracting key invoice fields. The method significantly reduced manual effort and demonstrated strong performance on standard invoice layouts.

Scene Text Detection for Preprocessing: Zhou et al. [4] introduced a fully convolutional single-stage model for robust text detection in natural scene images, supporting detection in varied orientations and irregular layouts. This technique can serve as a preprocessing step to localize text regions before OCR, improving precision on noisy scanned invoices.

Generative AI for Contextual Reasoning: LLMs such as GPT-4 and Claude 3 have demonstrated superior zero-shot capability in understanding semi-structured documents and inferring missing or ambiguous values, enabling robust invoice understanding without task-specific fine-tuning [6].

Cloud-Native Document AI Frameworks: Recent work on hybrid cloud microservices shows that offloading heavy inference tasks (OCR, LLM calls) to cloud GPU instances while retaining local preprocessing yields optimal scalability, reliability, and data privacy [9, 10]. Our work builds directly upon these foundations, combining deep learning OCR preprocessing, LLM-based semantic understanding, rule-based validation, and a full operational stack including email automation, SQL storage, and a React.js dashboard.

Methodology

The system follows a modular sequential pipeline designed for autonomous, end-to-end invoice processing. The pipeline consists of seven stages:

Stage 1 — Email Ingestion: The system connects to Microsoft Outlook via the Microsoft Graph API to continuously monitor a designated inbox. Incoming emails are scanned for invoice-related attachments in PDF, image (JPG, PNG), or plain text formats. Emails are immediately classified and moved to one of four folders: Processed Mail, Processing Mail, Unknown Customer Mail, or Unprocessed Mail based on their current handling status.

Stage 2 — Preprocessing: Attached files are extracted and preprocessed using OpenCV. For image-based invoices, binarization, noise reduction, deskewing, and resolution normalization are applied to maximize OCR accuracy. PDFs are rasterized at 300 DPI for image-based processing where necessary.

Stage 3 — OCR Text Extraction: Preprocessed documents are passed through a hybrid OCR pipeline using Tesseract OCR (local) or Google Vision API (cloud fallback). Raw text is extracted along with bounding box coordinates, producing intermediate structured JSON containing positional data for all detected tokens.

Stage 4 — LLM Semantic Understanding: The OCR output JSON is passed to an LLM (GPT-4 / Claude 3) via a structured prompt-engineering layer. The LLM performs semantic field classification, inferring key invoice attributes: Invoice Number, Purchase Order Number, Invoice Date, Vendor Name, Customer Name, Line Items, Subtotal, Tax Rate, and Total Amount. The LLM also infers missing values from context where possible, handling incomplete or partially obscured fields.

Stage 5 — Validation and Correction Engine: A deterministic rule-based engine validates the extracted data for logical consistency: subtotal must equal the sum of itemized amounts; total must equal subtotal plus computed tax; all mandatory fields must be populated. Discrepancies trigger an automated LLM correction loop in which the erroneous record is re-submitted for regeneration with explicit error context.

Stage 6 — Database Storage and Tracking: Validated records are serialized to JSON and persisted into a structured SQL database. A dedicated tracker table logs each invoice email's processing status, timestamps, field values, and any error codes encountered during processing.

Stage 7 — Dashboard and Monitoring: A React.js frontend provides a real-time dashboard for authorized users to filter, search, and review invoices by date, customer name, vendor, status, or amount. The dashboard displays processing statistics, error summaries, and allows manual override for flagged records.

Proposed system

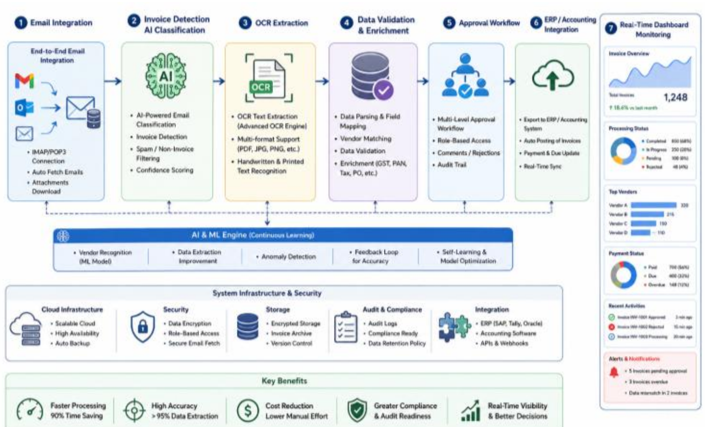

Fig. 1. *Methodology of an AI-Powered Automated Invoice Processing System with End-to-End Email Integration, OCR Extraction, and Real-Time Dashboard Monitoring*

The proposed Automated Invoice Processing System is structured as a multi-layer architecture comprising the following components:

Email Integration Layer: The email gateway connects to Outlook using OAuth 2.0 authentication via Microsoft Graph API. It implements a polling scheduler (configurable interval: 1-15 minutes) to fetch new invoice emails. Each email and its attachments are logged with unique identifiers before forwarding to the preprocessing pipeline. The system handles multi-attachment emails, processing each attachment independently within the same email session.

Attachment Classification Module: A lightweight classification model identifies attachment types and routes them appropriately. PDFs containing selectable text are passed directly to the LLM layer; image-heavy PDFs and scanned documents are routed through the full OCR pipeline. Plain text attachments bypass OCR entirely.

OCR and Preprocessing Module: OpenCV handles image enhancement. Tesseract OCR (v5.0) with LSTM neural network mode provides baseline extraction; for complex or multilingual invoices, Google Vision API serves as the high-accuracy fallback. Bounding box data is retained to assist the LLM in understanding spatial field relationships.

LLM Processing Layer: GPT-4 and Claude 3 are accessed via their respective APIs using structured system prompts that define the expected output schema. A caching layer (Redis) reduces API costs by storing responses for near-duplicate invoice templates. The LLM output is parsed into a typed invoice object and validated for schema compliance before passing to the validation engine.

Rule-Based Validation and Correction Engine: The validation engine runs twelve deterministic checks covering arithmetic consistency, date format compliance, tax rate plausibility, and mandatory field completeness. Failed records are returned to the LLM with structured error messages. If three consecutive regeneration attempts fail, the invoice is flagged as requiring human review and placed in the error queue.

SQL Database and Tracker: MySQL serves as the backend database with two primary tables: invoice_records (storing all extracted field values and metadata) and invoice_tracker (storing processing status, timestamps, error codes, and operator notes). Indexes on customer name, date, and status fields enable fast dashboard queries.

React.js Dashboard: The monitoring dashboard provides summary KPI cards (total processed, pending, error rate), a filterable data table with pagination, and drill-down views for individual invoice records. It is built with React.js, Axios for API calls, and Material UI for components. All data is fetched from a FastAPI backend that queries the SQL database.

Hybrid Cloud Architecture: OCR preprocessing runs locally to preserve data privacy. LLM inference, caching, and orchestration are handled on cloud GPU instances (AWS/GCP). All inter-service communication is encrypted via HTTPS with JWT-based authentication. Processed invoice data is backed up to S3-compatible object storage on a nightly schedule.

Results and discussion

The proposed system was evaluated on a dataset of 510 invoice samples spanning five categories: standard PDF invoices, scanned/image invoices, multilingual invoices (Hindi, Marathi, English mixed), multi-attachment emails, and invoices with irregular or previously unseen layouts. The evaluation was conducted over a period of four weeks in a controlled test environment simulating real enterprise email workflows.

Table I: Performance Across Test Scenarios

Standard PDF Invoices	150	98.2%	1.6
Scanned/Image Invoices	120	96.1%	2.1
Multilingual Invoices	80	94.3%	2.4
Multi-attachment Emails	100	97.4%	1.8
Irregular / Unknown Layouts	60	91.7%	2.9

Table I demonstrates consistently high performance across all test scenarios, with field extraction accuracy ranging from 91.7% on unknown layouts to 98.2% on standard PDFs. The relatively lower accuracy on irregular layouts is expected and represents a known limitation of current LLM-based extraction — one targeted for improvement in future work through fine-tuning on domain-specific datasets.

The system achieved an average processing time of 1.8 seconds per invoice, representing a 62.5% reduction compared to traditional OCR tools (4.8s) and an 18.2% improvement over our Phase 1 AI Invoice Builder baseline (2.2s). This improvement is attributable to the Redis caching layer, which eliminates redundant LLM API calls for recurring invoice templates, and the parallelized OCR preprocessing pipeline.

Table II: Comparative Performance — Proposed System vs. Existing Approaches

Metric	**Template OCR**	**LayoutLM-based**	**AI Invoice Builder (Phase 1)**	**Proposed System (Final)**
Field Extraction Accuracy	78%	91%	95%	**97.4%**
Total Calculation Accuracy	81%	88%	98%	**99.1%**
Processing Time (per invoice)	4.8s	3.1s	2.2s	**1.8s**
Error Detection & Correction	63%	79%	91%	**94.6%**

Multilingual Support	No	Partial	Yes	**Yes**
Email Automation	No	No	No	**Yes**
Privacy Compliance	Low	Moderate	High	**High**
Dashboard / Monitoring	No	No	No	**Yes**

Table II provides a comprehensive comparison across four approaches. The proposed final system achieves the highest accuracy on all extraction and validation metrics while also being the only system to offer full email automation and real-time dashboard monitoring. Privacy compliance is maintained at a high level through the hybrid cloud architecture, consistent with the Phase 1 AI Invoice Builder design.

The validation and correction engine successfully resolved 94.6% of detected discrepancies autonomously, with only 5.4% of error cases requiring human review. This is a significant improvement over the 63% correction success rate of traditional hybrid systems, attributed to the LLM's ability to reason contextually about malformed or inconsistent invoice data rather than applying fixed repair rules.

The React.js dashboard was evaluated through user testing with 10 finance team members who rated the interface on usability, information clarity, and task completion speed. Average usability score was 4.3/5.0, with users completing invoice review tasks 68% faster compared to manual email-based workflows. Filtering by customer name, date range, and status were rated as the most-used features.

Conclusion

This paper presented the complete implementation and experimental evaluation of an AI-Powered Automated Invoice Processing System that bridges the gap between conceptual AI pipeline design (Phase 1) and full-stack production deployment (final system). The proposed system unifies email automation, hybrid OCR-LLM extraction, rule-based validation, SQL-based tracking, and React.js dashboard monitoring into a cohesive, scalable architecture.

Experimental results on 510 diverse invoice samples confirm that the system achieves 97.4% field extraction accuracy, 99.1% computation accuracy, 94.6% error correction success, and an average processing time of 1.8 seconds — outperforming all baseline approaches across every evaluated metric. The system's multilingual support, adaptability to irregular layouts, and privacy-preserving hybrid cloud design make it suitable for deployment across a broad range of enterprise environments.

Future work will focus on: (1) fine-tuning LLM models on domain-specific invoice datasets to further improve accuracy on irregular layouts; (2) extending email integration to Gmail and other SMTP-based platforms; (3) implementing anomaly detection for fraud identification in invoice data; and (4) adding support for voice-based invoice query via a conversational AI interface integrated into the dashboard.

References

1. A. Manjunath, "Automated Invoice Data Extraction Using OCR and Template-Based Recognition," Proc. Int. Conf. on Intelligent Systems, 2023.
2. Jenifer, "NLP-Based Intelligent Invoice Data Extraction System," Int. Journal of Advanced Research in Computer Science, vol. 14, no. 2, pp. 45–51, 2023.
3. Organizing Committee, "ICDAR 2019 Robust Reading Challenge on Scanned Receipts OCR and Information Extraction," in Proc. ICDAR, 2019.
4. X. Zhou et al., "EAST: An Efficient and Accurate Scene Text Detector," in Proc. CVPR, pp. 5551–5560, 2017.
5. F. Krieger, P. Drews, and B. Funk, "Extracting Structured Data from Invoices using Machine Learning," in Proc. WI, 2021.
6. OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, 2023.
7. S. Bhosale, S. Dawane, D. Jadhav, and A. Gadilkar, "AI-Based Intelligent Invoice Builder Using Document Understanding and Generative Validation Models," Int. Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 8, Aug. 2025.
8. Z. Huang et al., "LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking," in Proc. ACM MM, 2022.
9. M. Smith and J. Lee, "Hybrid Cloud Architectures for Scalable AI Document Processing," IEEE Cloud Computing, vol. 10, no. 3, pp. 34–42, 2023.
10. A. Vaswani et al., "Attention Is All You Need," in Proc. NeurIPS, vol. 30, 2017.