RV College of Engineering®

Go, change the world

Department of AIML

ANN AND DL LAB

Automated GST Filing

Presented by

Aditya Tekriwal 1RV22AI003

ABHINAV 1RV22AI001

P SHREYAS 1RV22AI036

Faculty Mentors: Prof Somesh Nandi,Prof Anupama Kumar

Introduction

- The Automated GST Filing System is designed to simplify and streamline the Goods and Services Tax (GST) filing process for businesses. It automates key tasks such as data extraction, validation, reconciliation, and tax return submission, reducing manual effort and ensuring compliance with tax regulations.
- Traditional GST filing involves manual data entry, complex tax calculations, and frequent regulatory updates, making it prone to human errors and compliance risks. This system eliminates these challenges by leveraging automation and integration with the GST portal to ensure timely and accurate filings.
- The system enables users to upload invoices, validate tax details, calculate GST, and generate audit-ready reports. By incorporating real-time validation of GSTIN numbers, HSN/SAC codes, and tax rates, it ensures compliance with the latest GST rules and minimizes errors.
- A key feature of the system is its direct integration with the GST portal using government-approved APIs, allowing businesses to file GST returns seamlessly. The system also provides scheduled automatic filings, real-time notifications, and an intuitive user interface for ease of use.

Objectives

Automate the GST Filing Process

- The primary objective of this system is to reduce manual intervention in GST filing by automating tasks such as data extraction, validation, and return submission.
- Businesses can upload invoices directly, and the system will process the data, calculate GST, and generate returns, eliminating the need for repetitive manual work.

Ensure Compliance with GST Regulations

- The system is designed to keep businesses compliant with frequently changing GST rules and tax rates.
- It validates tax details, including GSTIN numbers, HSN/SAC codes, and tax slabs, ensuring that every filing adheres to the latest government policies.

Reduce Human Errors and Enhance Accuracy

- Manual tax filing is prone to errors that can result in penalties and financial losses.
- By implementing automated tax calculations and reconciliation mechanisms, the system helps minimize mismatches, incorrect filings, and missed deadlines.

Generate Audit-Ready Reports and Insights

- The system provides detailed tax reports, transaction summaries, and compliance logs that are essential for audits and financial analysis.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
1	A Part based Modeling Approach for Invoice Parsing Enes Aslan, Tugrul Karakaya, Ethem Unver and Yusuf Sinan Akgul	Proceedings of the 11th Joint Conference on Computer Vision, Imaging, and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 3: VISAPP	• Proposed Class-Free Parsing Method: A new method avoids reliance on predefined invoice classes, using a two-phase approach. • Phase 1: Invoice Part Detection: Employs techniques such as SVM, maximum entropy, and HOG to detect various invoice components. • Phase 2: Part-Based Modeling (PBM): Parses invoices by arranging detected parts in a deformable composition, inspired by techniques in face or body detection in images. • PBM Advantage: Handles any invoice type, ensuring versatility and reliability for participation banks. • Experimental Validation: Tested on real invoices, the system demonstrated effectiveness and robustness in performance.
2.	A High-Performance Document Image Layout Analysis for Invoices Mohammad Mohsin Reza, Md. Ajraf Rakib, Syed Saqib Bukhari, Andreas Dengel	DAS2018, 2018	• The paper proposes an advanced layout analysis method specifically designed for invoices. • Key features of the proposed method include: • Removal of table cell lines. • Merging of text lines. • Integration with anyOCR System: • The proposed invoice layout analysis is integrated into the anyOCR system, which was originally developed for processing historical and contemporary documents like books and magazines. • Performance Evaluation: • The performance of the advanced layout analysis is compared with: • The standard anyOCR pipeline. • The commercial ABBYY OCR system. • Results: The advanced layout analysis method achieved better OCR accuracy compared to both the standard and commercial systems.

RV College of Engineering®

Literature Survey

Go, change the world

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
3	Comparative Semantic Document Layout Analysis for Enhanced Document Image Retrieval Emad Sami Jaha	IEEE ACCESS (Volume 12)	• Describes a framework for: • Pairwise comparative-based automatic image annotation. • Document ranking based on comparative characteristics. • Comparative feature extraction. • Experiments and Results: • Conducted several retrieval experiments on a large, complex dataset of handwritten documents. • Extended performance evaluation showed that comparative-based methods outperform non-comparative methods. • Highlights the potential of the proposed approach for extending to other practical applications beyond DIR.
4	Document Layout Analysis Using Multigaussian Fitting Laiphangbam Melinda; Raghu Ghanapuram; Chakravarthy Bhagvati	IEEE Access (Volume: 12)	• Achieves performance comparable to top algorithms like MHS (winner of the ICDAR-RDCL2015 competition) and Aletheia (developed by PRImA Research Lab). • Tested on Indic script newspapers and other documents, showing: • >98% accuracy in identifying running text. • 82% accuracy in identifying other document elements. • Advantages: • Requires only one parameter (number of Gaussians for histogram fitting), making the method easy to automate and adaptable to various documents. • Next Steps: • Ground truth data for Indic script newspaper documents is being generated for more extensive testing and validation.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
5	Scenario Driven In-depth Performance Evaluation of Document Layout Analysis Methods C. Clausner; S. Pletschacher; A. Antonacopoulos	2011 International Conference on Document Analysis and Recognition	• Key Features: • Interval-based geometric representation of regions ensures both efficiency and accuracy., Offers a wide range of sophisticated evaluation measures for in-depth analysis of layout analysis systems. • Extends beyond simple benchmarking by providing detailed insights. • Customizability: Supports user-defined profiles to tailor evaluations for diverse real-world application scenarios. • Applications and Validation: • Delivered as part of the EU-funded IMPACT project for evaluating large-scale digitization initiatives., Validated with the dataset from the ICDAR2009 Page Segmentation Competition. • Significance: Facilitates precise evaluation of layout analysis methods in large-scale digitization contexts.
6.	Layout Analysis for Robust Resume Parsing Merve Elmas Erdem; Rabia BayraktarDengel	2023 6th International Conference on Information and Communications Technology (ICOIACT)	• Study Goals: • Explore the contribution of layout analysis to resume parsing. • Compare state-of-the-art methods on hierarchical structure estimation for resume PDFs. • Proposed Model and Dataset: • Developed a resume layout analysis model trained on custom layout categories. • Utilized a custom-built dataset for training and evaluation. • Key Findings: • Fine-tuning general-purpose layout detection models with a small dataset improves accuracy for specific tasks. • Demonstrates effectiveness in accurately analyzing resume structures for detailed information extraction.

RV College of Engineering®

Literature Survey

Go, change the world

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
7	Object Detection in Invoices Andrei-Ştefan Bulzan; Cosmin Cernăzanu-Glăvan	2022 26th International Conference on System Theory, Control and Computing (ICSTCC)	• Importance of Key Field Information Extraction: • A sought-after task in document processing, particularly for structured data extraction. • Approach: • Treats information extraction from invoices as an object detection task rather than relying on rule-based or NLP methods. • Models Used:YOLOv5,Scaled YOLOv4,Faster R-CNN • Data Preprocessing Method: • Proposed a preprocessing approach to improve model generalization across diverse invoice layouts. • Dataset: • Created a custom dataset with high variability in invoice layouts due to: • Lack of suitable public datasets. • Need for optimized annotation procedures specific to the task. • Experimental Results: • Encouraging performance, confirming that object detection is a viable method for key field information extraction.
8	An Overview of Data Extraction From Invoices Thomas Saout; Frédéric Lardeux; Frédéric Saubions Dengel	IEEE Access (Volume: 12)	• Techniques Discussed: • Digitalization of invoices for automation. • Use of natural language processing (NLP) to extract relevant information. • Application of machine learning (ML) and deep learning (DL) for: • Handling layout variability. • Reducing end-user tasks. • Adapting to new contexts. • Focus Areas: • Reviews a broad range of techniques for: • Information extraction. • Structure recognition in invoices. • Special focus on table processing, particularly graph-based approaches. • Purpose of the Overview: • Not to evaluate systems and algorithms but to provide a survey of techniques for different data extraction tasks.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper

9	Table Detection in Invoice Documents by Graph Neural Networks
	
	Pau Riba; Anjan Dutta; Lutz Goldmann; Alicia Fornés; Oriol Ramos; Josep Lladós
		2019 International Conference on Document Analysis and Recognition (ICDAR)	• Significance of Tabular Structures:
			• Represent logical or quantitative relationships in documents.
			• Critical in digital mailroom applications for processing large volumes of administrative documents with accuracy.
			• Challenges in Table Recognition:
			• Particularly challenging in unconstrained formats (e.g., no rule lines, unknown row/column information).
			• Proposed Graph-Based Approach:
			• Detects tables in document images using structural perception, independent of:
			  ◦ Raw textual content.
			  ◦ Language and text recognition quality.
			• Utilizes Graph Neural Networks (GNNs) to capture local repetitive structural patterns of tables in invoices.
			• Experimental Validation:
			• Tested on two invoice datasets, demonstrating promising results.
			• Dataset Contribution:
			• Developed a new benchmark dataset derived from RVL-CDIP invoice data.
			• To be publicly released to support further research in the field.

10	Table Understanding in Structured Documents Martin Holeček; Antonín Hoskovec; Petr Baudiš; Pavel Klinger	2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)	• Unified Information Extraction: • Demonstrates the ability to extract specific information from structurally diverse tables or table-like structures using a single model. • Proposed Representation: • Comprehensive page representation using:,Graph over word boxes.,Positional embeddings.,Trainable textual features.,Frames table detection as a text box labeling problem. • Dataset and Baselines: • Introduced a new dataset of pro forma invoices, invoices, and debit note documents.,Developed multiple baseline approaches to solve the labeling problem. • Novel Neural Network Model: • Proposed a new model that achieves strong, practical results.,Analyzed the impact of: • Graph convolutions.,Self-attention mechanisms.Practical Contribution: • Provides a detailed performance analysis and highlights the effectiveness of the model for table detection and information extraction in challenging domains.

Gaps

Layout Flexibility

- Most systems struggle with highly variable invoice formats
- Limited adaptation capabilities for new invoice types
- Need for more robust approaches to handle unconventional layouts

Missing End-to-End Solutions

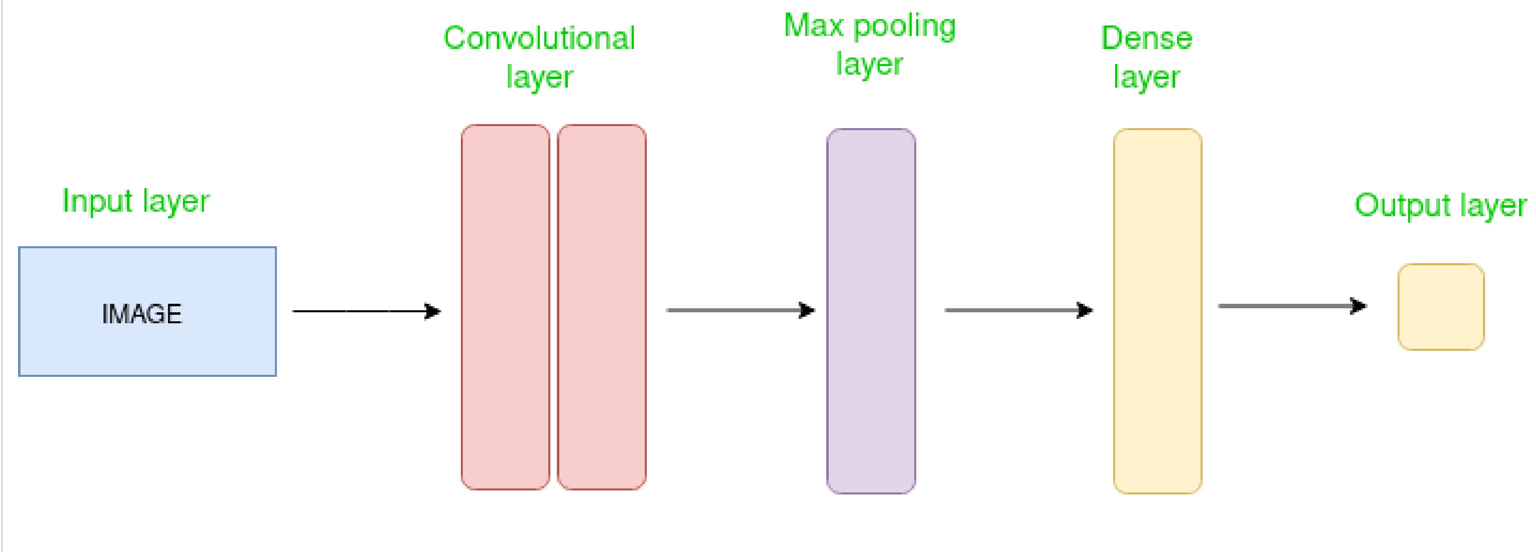
- Most research focuses on individual components rather than complete pipeline solutions
- Gap in understanding how different components interact and affect overall system performance
- Limited research on optimizing the entire invoice processing workflow

Industry-Specific Requirements

- Limited research on domain-specific invoice processing
- Need for customizable solutions for different industries
- Gap in handling industry-specific compliance requirements

CNN Architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.



Hardware requirements

Hardware Requirements:

Server-Side Hardware Requirements

● Processor: ○ Minimum: Intel Xeon or AMD equivalent with at least 4 cores
○ Recommended: 8+ cores for better performance, especially with high transaction volumes
● RAM: ○ Minimum: 32GB ○ Recommended: 64GB or more, depending on the anticipated data volume and concurrent users
● Storage: ○ Minimum: 256GB SSD (Solid State Drive) for the operating system and application files
○ Recommended: 1TB+ SSD for faster data access and improved performance
● Network: ○ Reliable and high-speed internet connection with sufficient bandwidth minimum 4G recommended
● GPU: ○ NVIDIA RTX series: (e.g., RTX 4000, 5000, 6000) for a balance of performance and cost.
○ GPU Memory: 16GB or more of VRAM is recommended for most deep learning models.

Client-Side Hardware Requirements

● Processor: ○ Minimum: Intel Core i5 or AMD equivalent
○ Recommended: Intel Core i7 or higher for smoother performance
● RAM: ○ Minimum: 8GB ○ Recommended: 16GB or more for better multitasking and responsiveness
● Storage: ○ Minimum: 256GB SSD ○ Recommended: 512GB SSD or more for ample storage space

Software requirements

○ Python: (Version: 3.11) - Widely used for its versatility, extensive libraries, and strong community support. ○ Deep Learning Frameworks:

● TensorFlow or PyTorch: (2.11 and 2.0.1) - Powerful libraries for building and training deep learning models.

● Supporting Libraries:

● NumPy:(1.23.5) For numerical computing.

● Pandas:(1.1.2) For data manipulation and analysis.

● Scikit-learn: (1.5.0) For machine learning algorithms (e.g., for simpler models or preprocessing).

● OpenCV:(4.9.0) For computer vision tasks (e.g., image processing for OCR). ○ Node.js: (Nodejs 14) - Excellent for building scalable and real-time applications.

● Web Frameworks:

○ Django/Django REST Framework: (Version:4.1.0) - Robust framework for building APIs and web applications.

○ Flask: (Version:2.2.0) - Lightweight and flexible framework for building APIs.

○ Express.js: (Version: V4) - Popular Node.js framework for building APIs.

● Databases:

○ MongoDB: (Version:4.2.0) - NoSQL database for flexible data modeling.

1. Data Collection

Efficient GST automation requires collecting comprehensive and accurate data from various sources. The key data sources include:

- Invoices:
  - Sales invoices (B2B, B2C, and export transactions).
  - Purchase invoices (input tax credit claims).
- Tax Records:
  - Previous GST returns (GSTR-1, GSTR-2A, GSTR-3B).
  - Annual returns (GSTR-9 and GSTR-9C).
- Master Data:
  - GSTIN details for the business and vendors/customers.
  - HSN/SAC codes for goods and services.
- Bank Statements:
  - To verify transaction amounts and reconcile with invoices.
- Regulatory Updates:
  - Latest GST rules and compliance requirements from the government portal.

2. Data Preprocessing

1. Invoice Data Processing

- Input:
  - Raw sales and purchase invoices (e.g., PDFs, Excel files, or database entries).
  - GSTINs, HSN/SAC codes, tax amounts, and transaction details.
- Process:
  - Extract data using OCR or parsing tools for unstructured files.
  - Validate GSTINs, tax rates, and invoice formats against GST rules.
  - Categorize invoices by type (B2B, B2C, export).
  - Calculate tax components (CGST, SGST, IGST) based on transaction value.
- Output:
  - Structured and validated invoice data ready for reconciliation.

2. Input Tax Credit (ITC) Reconciliation

- Input:
  - Purchase invoices and GSTR-2A data from the GST portal.
  - Vendor details and tax payment status.
- Process:
  - Match purchase invoices with GSTR-2A entries.
  - Identify mismatches (e.g., missing or excess claims).
  - Flag invalid ITC claims (e.g., ineligible goods/services, reverse charge).
- Output:
  - Reconciled ITC report with valid claims and discrepancies flagged for resolution.

3. Tax Liability Computation

- Input:
  - Sales invoice data, purchase invoice data, and applicable tax rates.
- Process:
  - Aggregate sales and compute output tax liability (CGST, SGST, IGST).

- Deduct eligible ITC from total tax liability.
- Apply reverse charge mechanism where applicable.
- Output:
  - Net tax payable for the period, segregated by tax types.

4. GST Return Preparation

- Input:
  - Processed sales and purchase data, reconciled ITC, and tax liability computations.
- Process:
  - Map data to GST return formats (e.g., GSTR-1, GSTR-3B).
  - Summarize data by categories like taxable, exempt, zero-rated, and export transactions.
  - Validate return data against GST filing rules.
- Output:
  - GST returns in the required format (JSON, CSV, or direct API upload).

5. Bank Reconciliation

- Input:
  - Bank statements and sales/purchase transactions.
- Process:
  - Match invoice amounts with payment entries in bank statements.
  - Identify pending payments or mismatched transactions.
- Output:
  - Reconciled report linking transactions to payments.

6. Regulatory Compliance Check

- Input:
  - Processed data, GST rules, and compliance updates.
- Process:
  - Check transactions for adherence to the latest GST regulations.
  - Flag non-compliant entries for review.
- Output:
  - Compliance report highlighting errors or potential risks.

1. Data Input

- Input:
  - Preprocessed data (e.g., structured invoice, transaction, or tax data).
  - Features extracted from data, such as tax amounts, transaction types, GSTINs, etc.
  - Labels (if supervised learning is used), such as compliance status or error classification.

2. Data Splitting and Preparation

- Process:
  - Split the data into training, validation, and testing sets.
  - Normalize or standardize numerical features to ensure uniform scaling.
  - Encode categorical features using one-hot encoding or label encoding.
  - Augment data (if necessary) to handle imbalances or enrich training diversity.
- Output:
  - Cleaned and split datasets ready for training the ANN/DL model.

3. ANN/DL Model Design

- Input:
  - Prepared data and selected features.
  - Network architecture parameters (e.g., number of layers, neurons per layer, activation functions).
- Process:
  - Design the neural network using a deep learning framework (e.g., TensorFlow, PyTorch, or Keras).
  - Configure input, hidden, and output layers based on the problem (e.g., classification or regression).
  - Specify activation functions, loss function, and optimizer.
- Output:
  - A compiled ANN/DL model ready for training.

4. Model Training

- Input:
  - ANN/DL model, training data, and validation data.
- Process:
  - Train the model using forward and backward propagation.

  - Apply techniques like dropout, regularization, or early stopping to prevent overfitting.
- Output:
  - A trained ANN/DL model with optimized weights and biases.

5. Model Evaluation

- Input:
  - Trained model and testing data.
- Process:
  - Evaluate the model's performance on unseen data using metrics like accuracy, precision, recall, F1 score, or mean squared error (depending on the task).
- Output:
  - Performance metrics and a confusion matrix (for classification tasks) or error analysis report.

6. Model Deployment

- Input:
  - Final trained model and live/production data.
- Process:
  - Deploy the model using a web service or API (e.g., Flask, FastAPI, or TensorFlow Serving).
- Output:
  - A deployed ANN/DL solution providing predictions or classifications for GST automation.

1. Model Testing

- Input:
  - Trained ANN/DL model.
  - Testing dataset (unseen data not used during training or validation).
- Process:
  - Use the testing dataset to make predictions with the model.
  - Compare predictions against ground truth labels.
  - Compute evaluation metrics such as accuracy, precision, recall, F1 score, confusion matrix (for classification), or mean absolute error and mean squared error (for regression).
- Output:
  - Performance metrics indicating the model's accuracy and robustness on unseen data.

2. Validation Against Business Requirements

- Input:
  - Predicted outputs from the model.
  - Expected outputs and domain-specific benchmarks.
- Process:
  - Validate predictions against domain requirements (e.g., compliance checks, error detection thresholds).

  - Identify discrepancies and refine thresholds or post-processing logic if required.
- Output:
  - Validation report outlining whether the model meets business and regulatory expectations.

3. Edge Case Testing

- Input:
  - Edge case scenarios or anomalous inputs (e.g., missing data, extreme values, or unusual invoice formats).
- Process:
  - Test the model’s response to edge cases and anomalous inputs.
  - Evaluate the model's robustness and ability to handle outliers without significant degradation in performance.
- Output:
  - Analysis of the model’s stability and behavior under atypical conditions.

4. System Integration Testing

- Input:
  - Deployed model integrated into the GST automation workflow.
- Process:
  - Test the end-to-end system with real or simulated production data.

  - Validate the seamless interaction between the model, APIs, and other components like data ingestion, preprocessing, and reporting modules.
- Output:
  - Integrated system validation report ensuring smooth workflow and real-time functionality.

5. User Acceptance Testing (UAT)
- Input:
  - Predictions and reports generated by the system.
  - Feedback from end-users, such as finance teams or tax professionals.
- Process:
  - Present test cases to end-users and gather feedback on model outputs and usability.
- Output:
  - Signed-off approval from end-users confirming system readiness for deployment.

6. Deployment Readiness Validation
- Input:
  - Finalized model, system workflows, and test/validation reports.
- Process:
  - Confirm all testing and validation criteria have been met.
- Output:
  - Deployment readiness certification indicating the system is prepared for production use.

THANK YOU