

Design and Implementation of a Lightweight Invoice OCR and Data Extraction System Using Tesseract

Prof. Parag Patil, Prof. Amol Potgantwar, Piyush Bhor, Harsh Sharma, Aarush Patil

Department of Information Technology

Sandip Institute of Technology and Research Centre (SITRC), Nashik, India

Parag.patil@gmail.com , amol.potgantwar@sitrc.org

piyushbhor41@gmail.com , hsk94692@gmail.com , theaarushpatil@gmail.com

Abstract: Invoices are commonly received by businesses as scanned documents, photographs or image files. Entering details manually into accounting systems is time-consuming and can lead to errors. This paper presents the design and preliminary implementation of a lightweight invoice OCR and data extraction system that aims to reduce manual effort. The proposed system uses OpenCV for image preprocessing, Tesseract OCR for text recognition, and regular expressions with keyword-based rules to extract important invoice details, including the invoice number, date, vendor name, and total amount. The extracted information is checked using basic validation rules and can be stored in CSV or SQLite format. A web interface is also being developed to allow users to upload invoices, review extracted details, correct errors, and export the results. Three configurations are proposed to compare OCR performance on original images, images with basic preprocessing, and images with additional skew correction. At the current stage, the upload interface and file validation have been implemented, while the main extraction modules and comparative testing remain under development. Therefore, this study presents the system design and planned evaluation rather than claiming measured accuracy results.

Keywords: invoice processing, optical character recognition, Tesseract, image preprocessing, rule-based information extraction, and document digitization

I. INTRODUCTION

Invoices are widely used in business transactions, and their details must be recorded accurately for accounting and payment. Small and medium-sized businesses often receive invoices as printed documents, scanned pages, or photographs. Employees usually manually enter details such as the invoice number, date, vendor name, and total amount into spreadsheets or accounting software. This process takes time, and mistakes such as incorrect digits, missing fields, or incorrect amounts can create problems during payment and record verification [1], [2].

Optical character recognition (OCR) can help reduce manual data entry by converting text in invoice images into machine-readable text. However, the quality of the output depends on the condition of the document. Images with shadows, poor lighting, or tilted text can lead to recognition errors [3], [7]. Even when the text is recognized correctly, the required invoice details must be identified and organized. Although commercial systems use deep-learning models and cloud-based services for this task, their cost and technical requirements may make them less suitable for small businesses or student projects [5].

This study therefore asks a practical question: How accurately can a purely classical pipeline, made of image preprocessing, Tesseract OCR, and keyword-anchored regular expressions, extract the main fields of an invoice, and how much does each preprocessing step contribute compared with running OCR on the raw image?



A. Contributions

The main contributions of this study are as follows.

- A lightweight invoice-processing system was designed using image preprocessing, Tesseract OCR, and rule-based field extraction. The system was designed to operate offline without a GPU or paid cloud services.
- A rule-based method is proposed to identify invoice details using different keyword variations and regular expressions, rather than depending on fixed positions on the page.
- An evaluation procedure was defined to compare three configurations: raw OCR, basic image preprocessing, and preprocessing with skew correction. The comparison will consider extraction accuracy, character recognition accuracy, processing time, and performance across different invoice layouts.
- This study discusses the main limitations of the proposed approach, including variations in invoice layouts, OCR errors, line-item extraction, and support for invoices in various languages.

II. RELATED WORK**A. Tesseract-Based Invoice Processing**

Several studies have explored the use of Tesseract OCR for processing invoices and receipts. Deepa et al. presented an invoice-processing system that combined image cleaning with Tesseract OCR to extract important details from structured invoices [1]. Kamisetty et al. compared Keras OCR, EasyOCR, and Tesseract for converting invoice images into JSON and CSV formats [2]. Kumar et al. used OpenCV with Tesseract to process bill and receipt images containing shadows and watermarks [3], while an earlier IEEE study examined OCR techniques for extracting text from printed bills and invoices [4]. However, the existing studies discussed here leave room for further evaluation of field-level accuracy across invoices from various vendors and their layouts.

B. Image Preprocessing for OCR

Recognition accuracy depends strongly on the input image. Bieniecki et al. studied preprocessing for camera-captured documents and showed that correcting geometric distortion and illumination problems improves OCR output [7]. Global thresholding using Otsu's method [9] and locally adaptive binarization, such as Sauvola's method [10], are the usual choices for converting a grayscale page to black and white, with local methods coping better with uneven lighting. Image operations in this study relied on the OpenCV library [11].

C. The Tesseract Engine

Smith described the architecture of the Tesseract OCR engine, including its methods for identifying text lines, segmenting words, and recognizing characters [8]. Tesseract is an open-source tool that can run locally without an internet connection. These features make it a suitable option for invoice processing systems that need to keep costs low and operate on ordinary computers.

D. Learning-Based Information Extraction

Learning-based methods use machine learning models to identify information from documents by considering both the text and layout. For example, CloudScan uses neural networks to extract invoice details without relying on predefined templates [12]. Chargrid represents document information in a two-dimensional grid of characters, whereas LayoutLM combines text and layout information to understand the document content [13], [14]. Datasets such as SROIE and CORD are also available for evaluating document text extraction and information extraction systems [15], [16]. Although these approaches offer alternatives to rule-based extraction, their implementation may require suitable training data and additional computational resources.



E. Surveys and Post-Processing

Saout et al. surveyed invoice data extraction and concluded that rule-based, OCR-driven methods remain practical for well-defined fields, even as deep learning alternatives grow [5]. Avyodri et al. reviewed OCR post-processing and noted that validation after recognition still improves reliability [6].

F. Research Gap

Previous studies have explored OCR-based invoice processing and advanced learning-based methods. However, there is still a need to study how individual image preprocessing steps affect the accuracy of lightweight, rule-based invoice extraction systems. In particular, comparing raw OCR with basic preprocessing and preprocessing that includes skew correction can help demonstrate the contribution of each stage. This study addresses this need by proposing a threeconfiguration evaluation approach that examines the field extraction accuracy, character recognition accuracy, processing time, and performance across different invoice layouts.

III. PROPOSED SYSTEM**A. Overview**

The proposed system allows users to upload an invoice as an image or a single-page PDF and extract four important fields: the invoice number, invoice date, vendor name, and total amount. The extracted information is checked, stored, and displayed through a web interface, where users can review the results and correct any errors. The system uses lightweight tools such as OpenCV and Tesseract OCR, allowing it to run on a regular laptop without requiring a GPU. Once the required software was installed, the system was designed to work without an Internet connection.

B. Three-Layer Architecture

Ingestion and Extraction Layer: receives the uploaded file, stores it locally, applies OpenCV preprocessing, and runs Tesseract to obtain the raw text.

Field Extraction and Storage layer: applies rule-based parsing, validates each field, and writes the result to CSV or SQLite.

Reporting and User Interface layer: a Flask or Streamlit dashboard where the user uploads invoices, searches and filters earlier records, corrects errors, and exports reports.

C. Image Preprocessing

The system accepts invoice images in JPG and PNG formats and single-page PDF files. Before OCR was performed, the images were processed to improve text readability. The preprocessing steps included converting the image to grayscale, reducing the noise, and converting it into a black-and-white image using thresholding. Two thresholding methods were considered: global and adaptive. Adaptive thresholding is useful for images with uneven lighting because it adjusts to local brightness variation. The system also includes skew correction to straighten tilted text lines, which may help improve OCR results.

D. OCR and Field Parsing

After preprocessing, the invoice image is passed to the Tesseract OCR through the pytesseract library to recognize text. The extracted text was then searched using regular expressions and keywords to identify the required invoice details. For example, invoice numbers may be identified using terms such as "Invoice No," "Inv #," or "Bill No." Different date patterns are used to locate the invoice date, whereas keywords such as "Total," "Grand Total," and "Amount Due" help identify the total amount. Because invoice layouts vary among vendors, the proposed method searches the extracted text instead of relying on fixed page coordinates. The vendor name is identified from the initial lines of the document, where the issuing company name appears.



E. Validation and Storage

After extracting the invoice details, the system checks the values before storing them in the database. The invoice date is checked against supported date formats, the total amount must be numeric, and the vendor name must not be empty; the system also aims to handle common OCR errors, such as the letter “O” being recognized instead of the digit “0,” by normalizing the values before validation. If a field fails these checks, the record is flagged for manual review so that the user can correct the information instead of losing the extracted data. Validated records are intended to be stored in CSV files or an SQLite database.

F. Compared Configurations

Three configurations were defined such that preprocessing was the only difference between the runs.

Configuration A (baseline): Tesseract applied directly to the original image.

Configuration B (basic): grayscale conversion, denoising, and global thresholding.

Configuration C (proposed): grayscale conversion, denoising, adaptive thresholding, and skew correction.

The same parsing rules, validation rules, and Tesseract settings were used in all three.

TABLE I: DESIGN-LEVEL COMPARISON OF THE THREE CONFIGURATIONS (NOT MEASURED)

Property	Config. A	Config. B	Config. C
Grayscale and denoising	No	Yes	Yes
Thresholding	None	Global	Adaptive
Skew correction	No	No	Yes
Expected cost	Lowest	Low	Slightly higher

IV. IMPLEMENTATION

TABLE II: COMPONENTS OF THE PROPOSED SYSTEM

Component	Tool / Technology Used
Programming language	Python 3.9 or higher
Image processing	OpenCV (opencv-python)
OCR engine	Tesseract OCR through pytesseract
Component	Tool / Technology Used
Field parsing	Python re module (regular expressions and keyword anchors)
Data handling	Pandas for tabular handling and CSV export
Database	SQLite3 (built into Python)
Web interface	Flask or Streamlit
Hardware requirement	Intel Core i5 or equivalent, 8 GB RAM, no GPU

The project follows an iterative agile process in which preprocessing, OCR, and parsing are built and tested separately before integration. The work is planned over 12 weeks: literature review and sample collection (weeks 1-2), preprocessing (3-4), OCR integration (5-6), parsing (7-8), storage, validation, and interface (9-10), and testing and documentation (11-12).

V. EXPERIMENTAL SETUP

A. Dataset

The evaluation dataset will include invoices collected from local vendors, with personal information removed, along with publicly available invoice and receipt datasets such as SROIE and CORD, where their licences permit use. The



dataset will contain clean scans, tilted images, and photographs taken under different lighting conditions to examine how image quality and invoice layout affect extraction performance. For each invoice, the target fields and complete text will be manually recorded to create reference data for comparison with the system's output.

B. Hypotheses

H1: Basic image preprocessing (Configuration B) is expected to improve field extraction accuracy compared with raw OCR (Configuration A) because reducing noise and improving contrast may help in reducing recognition errors.

H2: Adding adaptive thresholding and skew correction (Configuration C) is expected to further improve the extraction accuracy, particularly for photographed invoices and images with tilted text.

H3: Additional preprocessing steps are expected to increase the processing time per invoice slightly.

C. Evaluation Metrics

Field-level extraction accuracy: percentage of invoices for which each target field is extracted correctly.

OCR character accuracy: percentage of characters recognized correctly against the ground truth text.

Processing time: average time from upload to structured output.

Robustness: Consistency of accuracy across vendors, layouts, and image quality groups.

D. Evaluation Protocol

Preparation: All invoices were grouped by image quality and layout, and their ground truth was recorded.

Processing: Every invoice is run through Configurations A, B, and C with identical OCR settings and parsing rules.

Logging: Raw OCR text, extracted fields, validation flags, and processing time were stored for each run.

Scoring: The extracted fields were compared with the ground truth automatically, and the character accuracy was computed from the OCR text.

VI. CURRENT STATUS AND PRELIMINARY OBSERVATIONS

At the current stage, the web dashboard, invoice upload feature, drag-and-drop option, and file validation have been implemented. These features have been tested to determine how the system handles supported and unsupported file types. The image preprocessing, OCR, field extraction, and data storage modules are still under development or planned for integration. Because the complete pipeline has not yet been tested, the system's extraction accuracy, character recognition accuracy, and processing time have not been measured. Further testing is required to compare the three configurations and determine how well the system works with invoices from different vendors.

TABLE III: IMPLEMENTATION STATUS AT THE TIME OF MANUSCRIPT WRITING

Module	Status
Web dashboard	Implemented
Invoice upload button and drag-and-drop area	Implemented
File type and input validation	Implemented
Image preprocessing (OpenCV)	In development
Tesseract OCR integration	Planned
Module	Status
Rule-based field parser	Planned
Field validation and CSV/SQLite storage	Planned
Comparative evaluation	Not yet conducted



VII. DISCUSSION AND LIMITATIONS

A. Discussion

Previous studies have suggested that image preprocessing and validation can improve the reliability of OCR-based information extraction [2], [6]–[7]. Research has also shown that rule-based methods are useful for extracting specific fields from structured documents [5]. The proposed system applies these ideas to invoice processing using OpenCV, Tesseract OCR, and keyword-based rules. However, the effectiveness of this approach has not been experimentally confirmed. The planned evaluation will help determine whether the proposed preprocessing configurations improve the extraction accuracy and the performance of the system across different invoice layouts.

B. Limitations

Keyword-anchored rules may miss fields on unusual layouts; therefore, some vendors will require additional patterns [4].

Only the four header-level fields are targeted; line-item and table extraction are beyond the scope of this study.

Only English invoices were considered for the first evaluation.

Tesseract can confuse similar characters; therefore, manual correction is still part of the workflow.

Heavily blurred, handwritten, or severely damaged invoices may not be readable by OCR.

Learning-based systems [12]–[14] are likely to outperform rule-based approaches on highly irregular layouts; the aim here is a low-cost, explainable alternative, not state-of-the-art (SOTA) accuracy.

C. Threats to Validity

The evaluation may be limited by the size of the invoice dataset. If the collected invoices come from only a small number of vendors, the results may not represent the performance of the system for invoices from other vendors. In addition, the same team was responsible for developing the extraction rules and preparing the test data, which could introduce bias into the evaluation process. Therefore, the results should be interpreted with these limitations in mind and may not apply to all invoice layouts, countries, or document qualities.

VIII. CONCLUSION AND FUTURE WORK

This study presents the design of a lightweight invoice OCR and data extraction system that combines OpenCV-based image preprocessing, Tesseract OCR, and rule-based text field extraction. The proposed system identifies important invoice details, validates the extracted information, stores the results, and displays them through a web interface. Three configurations were defined to study the effects of different preprocessing steps on extraction accuracy and processing time. At the current stage, the web interface and file validation have been implemented, and the remaining modules are still under development. Therefore, the performance of the system has not yet been established through experimental results. Further implementation and testing are required to evaluate its effectiveness.

Future work includes the following: (1) table and line-item extraction; (2) multilingual support through additional OCR language packs; (3) template learning so that the system remembers the layout of repeat vendors; (4) export formats for accounting software, such as Tally and QuickBooks; (5) batch processing of many invoices; and (6) evaluation on public benchmarks and comparison with learning-based extractors.

REFERENCES

1. R. N. Ashlin Deepa et al., “An Intelligent Invoice Processing System Using Tesseract OCR,” in *Proc. Int. Conf. Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, IEEE, 2024, doi: 10.1109/ADICS58448.2024.10533509.
2. V. N. S. R. Kamisetty, B. S. Chidvilas, and S. Revathy, “Digitization of Data from Invoice using OCR,” in *Proc. 6th Int. Conf. Computing Methodologies and Communication (ICCMC)*, IEEE, 2022, doi: 10.1109/ICCMC53470.2022.9754117.



3. V. Kumar et al., "Extraction of Information from Bill Receipts Using Optical Character Recognition," in *Proc. Int. Conf. Smart Electronics and Communication (ICOSEC)*, IEEE, 2020.
4. H. Sidhwa, S. Kulshrestha, S. Malhotra, and S. Virmani, "Text Extraction from Bills and Invoices," in *Proc. 2018 Int. Conf. Advances in Computing, Communication Control and Networking (ICACCCN)*, IEEE, 2018, pp. 564–568, doi: 10.1109/ICACCCN.2018.8748309.
5. T. Saout, F. Lardeux, and F. Saubion, "An Overview of Data Extraction From Invoices," *IEEE Access*, vol. 12, pp. 19872–19886, 2024, doi: 10.1109/ACCESS.2024.3360528.
6. R. Avyodri, S. Lukas, and H. Tjahyadi, "Optical Character Recognition (OCR) for Text Recognition and its Post-Processing Method: A Literature Review," in *Proc. 1st Int. Conf. Technology Innovation and Its Applications (ICTIIA)*, IEEE, 2022.
8. W. Bieniecki, S. Grabowski, and W. Rozenberg, "Image Preprocessing for Improving OCR Accuracy," in *Proc. Int. Conf. Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, IEEE, 2007, doi: 10.1109/MEMSTECH.2007.4283429.
9. R. Smith, "An Overview of the Tesseract OCR Engine," in *Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR)*, vol. 2, IEEE, 2007, pp. 629–633, doi: 10.1109/ICDAR.2007.4376991.
10. N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
11. J. Sauvola and M. Pietikäinen, "Adaptive Document Image Binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.
12. G. Bradski, "The OpenCV Library," *Dr. Dobbs's Journal of Software Tools*, 2000.
13. R. B. Palm, O. Winther, and F. Laws, "CloudScan – A Configuration-Free Invoice Analysis System Using Recurrent Neural Networks," in *Proc. 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR)*, 2017.
14. A R. Katti et al., "Chargrid: Towards Understanding 2D Documents," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
15. Y. Xu et al., "LayoutLM: Pre-training of Text and Layout for Document Image Understanding," in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2020.
16. Z. Huang et al., "ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2019.
17. S. Park et al., "CORD: A Consolidated Receipt Dataset for Post-OCR Parsing," in *Workshop on Document Intelligence at NeurIPS*, 2019.

