Automated Data Entry Through Image-Processing for Advanced Medical Inventory

Aditya S. Thorat[1], Sumit Thakur[2], Sanket Solanke[3], Mayur Tidke[4], Aradhana Pawar[5]
[1, 2, 3, 4]Students, [5]Guide, Department of Computer Engineering, Dr. D. Y. Patil College of Engineering and Innovation, Varale, Talegaon Dabhade, Pune, Maharashtra, India.

Abstract: There's a new Artificial Intelligence (AI) model that aims to change how data entry is done by precisely recognizing and extracting important information from bill images. This model extensively processes visual data using the solid functionalities of Tesseract OCR and OpenCV, transforming it into a structured format that can be utilized for smooth data entry. The addition of this feature to our existing application "Advanced Inventory for Medical Distributors (AIMD)" improves on its use beyond the medical field. Integration of this technology guarantees increased efficiency in data entry across different sectors significantly quickening the process of information retrieval with accuracy being maintained. The ability of this model to turn complex visual data into readily available digital information is a major leap forward in automated data management which has far reaching implications for simplifying operational workflows.

Keywords: AI Model, Optical Character Recognition (OCR), Document Image Processing, Structured Data, Automated Data Management, Bill Image Analysis.

I. INTRODUCTION

Data entry looks simple on paper, yet it grinds day-to-day operations to a halt in just about every sector you can name. Someone still has to sit there and copy numbers by hand, especially when it comes to brittle old invoices, and that tedium opens the door to slip-ups that cost time and money. The backlog piles up because speed and accuracy refuse to ride in the same truck. Techspeak keeps telling us that smarter vision systems and sharper algorithms could clear that logjam, so we decided to see what would actually stick.

In the following pages we lay out a fresh model that hunts down the vital digits buried in a photo of a bill and spits them out in a neat table without any human reaching for the keyboard. Tesseracts text muscle teams with OpenCVs picture smarts to map, read, and tidy the snapshot before it ever hits a spreadsheet. The upgrade drops into Advanced Inventory for Medical Distributors, or AIMD, a program that already keeps pharmacies moving, and it turns out the same trick works just as well for freight firms, construction crews, and just about anyone else elbow-deep in paperwork. By wiring this engine into the daily grind we hope to shave off both the delay and the guesswork that still haunt entry desks from dawn till quitting time.

II. LITERATURE REVIEW

The widespread problem of slow and mistake-prone manual data entry from paper documents bill images, poses a big challenge in many industries. This old-school method creates work slowdowns, holds up information processing, and drives up costs showing a pressing need for automated and precise solutions. The rise of AI and computer vision tech offers a game-changing way to tackle these issues by making it possible to recognize and pull out information from visual data.

A lot of studies have paved the way for automatic text extraction and document processing. The creation and use of Optical Character Recognition (OCR) technologies stand out among these contributions. For example, Ashlin Deepa et al. [3] showed an automatic invoice processing system that used Tesseract OCR to good effect. This system pulled out key information such as vendor details and totals, and turned it into a structured JSON format. Their work shows how useful Tesseract OCR can be in processing business documents, though they point out that high-quality image input matters for the best results. Adding to this, Kaundilya et al. [4] looked deeper into automatic text extraction from images using OCR systems. They highlighted how well these methods work to digitize text content in general. Su et al. [2] also took a close look at the basic abilities of image processing for text recognition. They explored technologies made to spot and understand text in many different image settings. Together, these studies show that OCR and image processing work well and can be relied on to turn visual text into digital information automatically.

Also, progress in machine learning (ML) and deep learning has improved the strength of image-based data extraction. Some related studies have looked at wider uses like inventory management and predictive analytics using ML [7, 8], but their basic ideas about data processing and feature extraction still apply. In particular, using neural network algorithms to recognize and validate text elements, as Malphedwar et al. [9] explain for handwritten equations, shows how powerful advanced image processing methods, including convolutional layers, can be for handling complex visual data to extract text . This ability is key for reliable OCR systems that can read various visual inputs.

Building on this previous research, our paper presents a new AI model designed to cause a revolution in data entry by recognizing and extracting important information from bill images with high accuracy. This work takes inspiration from previous studies and aims to build on the insights and needs outlined in them. One such study is the review paper "Advanced Inventory for Medical Distributor (AIMD)" by Thorat et al. [10]. It pointed out the challenges faced in traditional inventory management systems and how AI can help streamline processes like data entry. While earlier research has shown the effectiveness of Tesseract OCR and various image processing techniques for text recognition, our model combines the strong capabilities of Tesseract OCR with OpenCV. This allows us to carefully process visual data and turn it into a structured, usable format that is easy for data entry. We hope this integration will tackle common issues linked to real-world bill images and increase the usefulness of automated data entry across different fields. By using these proven technologies specifically for bill images, our approach aims to significantly improve both efficiency and accuracy in data entry. This represents a major advancement in automated data management and operational efficiency.

III. METHODOLOGY

Our approach focuses on converting document images into structured, meaningful data such as JSON. The process involves several key stages — from cleaning the raw image to extracting semantic relationships and formatting the output. Here's how the pipeline works:

A. Image Preprocessing

Before diving into text extraction, the document image must be as readable as possible. Preprocessing ensures that the text stands out clearly, reducing noise and distortion.

- Binarization converts grayscale images into black and white based on intensity thresholds [1]. Equation:

$$P_{new}(x,y) = \begin{cases} 255, & \text{if } P_{original}(x,y) > \text{Threshold} \\ 0, & \text{otherwise} \end{cases}$$

- Noise Reduction involves applying filters like Gaussian or median blur to smooth out random artifacts [2]. Equation:

$$P_{new}(x,y) = \sum_{i,j} \text{Kernel}\,(i,j) \cdot P_{original}(x-i, y-j)$$

- Deskewing corrects tilted text using line angle analysis and rotation transformations, often powered by Hough Transforms [3].

- Contrast Enhancement improves visual clarity by adjusting pixel intensities for better separation of text-background [4].

B. Optical Character Recognition (OCR)

OCR is the core process of recognizing text in an image and converting it to a machine-readable format.

- Character Segmentation breaks the image into recognizable symbols.

- Feature Extraction captures geometric patterns and structures like curves, intersections, and edges.

- Classification then predicts the characters using deep learning models like Convolutional Neural Networks (CNNs) [5]. Equation (Softmax):

$$P(\text{char}_k|\text{features}) = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

Many modern OCR engines use a hybrid of CNNs and Recurrent Neural Networks (RNNs) or LSTMs to better interpret sequential context [6].

C. Layout and Structure Detection

Beyond raw text, we need to understand the document's layout — identifying tables, headings, and the spatial relationship between components.

- Geometric Analysis examines bounding box positions and alignment to group textual blocks [7].

- Line/Rule Detection uses edge detection (like Sobel or Canny filters) and Hough Transforms to identify table lines or form separators [3].

- Table Detection involves models like YOLOv3 or Faster R-CNN, trained to detect tables as visual objects in documents [8], [9].

- Cell Detection may rely on:
    - Rule-based methods, using detected lines,
    - Structure-based methods, relying on alignment and spacing,
    - Or Graph Neural Networks (GNNs), which model layout elements as interconnected nodes [10].
- Reading Order Analysis ensures text is interpreted in logical order, especially important in multi-column documents or forms [11].

D. Semantic Understanding and Relationship Extraction

This stage interprets the extracted text to understand its meaning within the document.

- Named Entity Recognition (NER) is used to extract entities like dates, amounts, and names [12].
- Relationship Extraction links labels and values (e.g., "Invoice Number: XYZ123") using pre-trained Transformer models like BERT [13], [14]. Attention Equation:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Confidence Scores are provided for each field to indicate the reliability of the extraction [15].

E. Post-processing and Output Formatting

The final stage involves cleaning, structuring, and formatting the output data.

- Field Mapping ensures the extracted values are aligned with predefined JSON fields.
- Type Conversion standardizes data (e.g., converting "12 March 2024" to a date object).
- Custom Cleaning Logic corrects common OCR issues — such as misrecognized characters or misplaced delimiters — with domain-specific rules [16].

IV. RESULTS

The developed AI model, leveraging Tesseract OCR and OpenCV, successfully automates the extraction of critical information from bill images, transforming visual data into a structured JSON format suitable for data entry. The system's performance was evaluated across various aspects, including image preprocessing effectiveness, OCR accuracy, layout and structure detection, and the overall quality of extracted and formatted data.

a. Input:

Sr.	Particulars	Pack	HSN	Mfg.	Batch No.	Exp.	Qty	Free	MRP	Rate	Dis%	GST %	Amount
1	Medicine 1q23	Nos.	General	Cipila	ABC	10-2024	5	1	150	100	0	5	500
2	Medicine 1q23	Nos.	General	Cipila	ABC	10-2024	10	2	150	80	0	5	800
3	Medicine 1q23	10s	General	Cipila	ABC	10-2024	1	0	150	100	0	5	100
4	Medicine 1q23	10s	General	Cipila	ABC	10-2024	1	0	150	100	0	5	100
5	Medicine 1q23	10s	HS1234	Cipila	ABC	10-2024	2	0	150	100	10	5	200
6	Medicine 1q23	10s	HS1234	Cipila	ABC	10-2024	1	0	150	100	8	5	100
7	Zt	Nos.	H456	Cipila	N.A.		10	0	1500	1000	0	12	10000

Figure 1: Input Image

The Input given to the program is an image containing a table that consists the data of expenses on medicines ordered by a medical distributor. The table image is a cropped version of the actual bill received by the distributor. The designed model is expected to extract the text data from the table while retaining the relation between rows and columns.

b. Output:

The output of the program is expected to accurately be the text in the table from the image. The text is expected to be returned in JSON format. JSON (JavaScript Object Notation) is a widely used data-interchange format that's both human-readable and machine-parsable, making it ideal for exchanging data across different systems. It's particularly popular in web development, especially for transferring data between a server and a web application.
The achieved output is as follows:

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930


```json
[
  {
    "Sr.": 1,
    "Particulars": "Medicine 1923",
    "Pack": "Nos.",
    "HSN": "General",
    "Mfg.": "Ciplla",
    "Batch No.": "ABC",
    "Exp.": "10-2024\n\u2610",
    "Qty": 5,
    "Free": 1,
    "MRP": 150,
    "Rate": 100,
    "Dis%": 0,
    "GST %": 5,
    "Amount": 500
  },
  {
    "Sr.": 2,
    "Particulars": "Medicine 1q23",
    "Pack": "Nos.",
    "HSN": "General",
    "Mfg.": "Ciplla",
    "Batch No.": "ABC",
    "Exp.": "10-2024",
    "Qty": 10,
    "Free": 2,
    "MRP": 150,
    "Rate": "80\n0",
    "Dis%": 0,
    "GST %": 5,
    "Amount": 800
  },
```


Figure 2: JSON_Output-1


```json
  {
    "Sr.": 3,
    "Particulars": "Medicine 1q23",
    "Pack": "10s",
    "HSN": "General",
    "Mfg.": "Ciplla",
    "Batch No.": "ABC",
    "Exp.": "10-2024",
    "Qty": 1,
    "Free": 0,
    "MRP": 150,
    "Rate": 100,
    "Dis%": 0,
    "GST %": 5,
    "Amount": 100
  },
  {
    "Sr.": 4,
    "Particulars": "Medicine 1923",
    "Pack": "10s",
    "HSN": "General",
    "Mfg.": "Ciplla",
    "Batch No.": "ABC",
    "Exp.": "10-2024",
    "Qty": 1,
    "Free": 0,
    "MRP": "5555\n150",
    "Rate": 100,
    "Dis%": 0,
    "GST %": 5,
    "Amount": "\u314e\u314e\n100"
  },
```


Figure 3: JSON_Output-2

```
  {
    "Sr.": 5,
    "Particulars": "Medicine 1q23",
    "Pack": "10s",
    "HSN": "H51234",
    "Mfg.": "Ciplla",
    "Batch No.": "ABC",
    "Exp.": "10-2024\n\u2610",
    "Qty": 2,
    "Free": 0,
    "MRP": 150,
    "Rate": 100,
    "Dis%": 10,
    "GST %": 5,
    "Amount": "200\n88"
  },
  {
    "Sr.": 6,
    "Particulars": "Medicine 1923",
    "Pack": "10s",
    "HSN": "H51234",
    "Mfg.": "Ciplla",
    "Batch No.": "ABC",
    "Exp.": "10-2024",
    "Qty": 1,
    "Free": 0,
    "MRP": 150,
    "Rate": "100\n8",
    "Dis%": 8,
    "GST %": 5,
    "Amount": 100
  },
```


Figure 4: JSON_Output-3


```
  {
    "Sr.": "7\nM",
    "Particulars": "Zt",
    "Pack": "Nos.",
    "HSN": "H456",
    "Mfg.": "Ciplla",
    "Batch No.": "1\nN.A.",
    "Exp.": null,
    "Qty": 10,
    "Free": 0,
    "MRP": 1500,
    "Rate": 1000,
    "Dis%": 0,
    "GST %": 12,
    "Amount": 10000
  }
]
```


Figure 5: JSON_Output-4

c. Evaluation:

The model's performance was evaluated using a simple accuracy metric, which is the ratio of correctly predicted values to the total number of values in the generated JSON objects. In our evaluation dataset, the model achieved an average accuracy of 81%. However, this accuracy depends heavily on the quality of the input image. Key factors such as image clarity, noise, and geometric distortions, like tilt or skew, greatly affect the model's ability to accurately extract information. While large-scale training is often seen as a main method for improving model accuracy and generalization, it also carries risks. Expanding the training scale can lead to problems, including the introduction of bias from unrepresentative datasets or a higher chance of overfitting. In overfitting, the model learns the training data too well and struggles to perform effectively on new, unseen images.

V. CHALLENGES AND LIMITATIONS

1. Limited Flexibility with New Document Formats/Products: The model may struggle to extract information accurately when it encounters new or very different bill formats. Adjusting to new layouts or product descriptions not covered during training might need retraining or fine-tuning. This could result in initial inaccuracies or poor recommendations until enough new data is collected and processed.

2. Difficulty in Extracting Relational Data from Tables: A major challenge in image-based data extraction is accurately recognizing text within tables and keeping its relational structure (e.g., linking item names with quantities and prices). Complex table layouts, different line styles, or unclear delimiters can make it hard for the model to correctly parse tabular data and maintain the relationships.

3. Security and Privacy Concerns: Managing sensitive information from bills, like financial data or personal details, requires strict security measures. The system must ensure that data is well protected against cyber threats, unauthorized access, and data breaches. This adds complexity and cost to implementation.

4. Image Quality Dependency: As noted in the literature, OCR systems, including those using Tesseract, are sensitive to input image quality. Poor image quality (e.g., blurriness, low resolution, tilted angles, inconsistent lighting, or complex backgrounds) can greatly affect the accuracy of text recognition and information extraction, leading to errors.

5. User Training and Adaptation: For an AI-powered data entry system to be successful, users need proper training. A learning curve for those unfamiliar with AI systems might cause initial delays in adoption or operational mistakes, affecting how efficient the system seems.

6. Complexity of Integration with Existing Systems: Adding a new AI3 model for data entry into current applications and workflows can be complicated. It requires careful mapping and synchronization of data, which may create challenges for smooth deployment.

VI. CONCLUSION

This research activity aimed to create a new AI model that transforms data entry processes. It does this by recognizing and extracting key information from bill images. The model uses the strong features of Tesseract OCR and OpenCV to carefully process visual data and convert it into a structured format. This approach meets the growing demand for faster and more accurate information processing in many industries.

Our combined approach marks an important advancement in automated data management. We have built a basic framework for smart information extraction from complex visual documents. This shows the significant potential of AI in making workflows more efficient. The model offers a valuable improvement in achieving better efficiency in data entry.

Although the system has reached a promising level of performance, the first deployment and evaluation revealed areas that need more work. We found chances to improve accuracy in data extraction, especially from different and complicated document layouts. Additionally, ensuring smooth compatibility with a wider range of real-world situations and various input qualities offers paths for future improvement. These observations highlight the evolving nature of AI development in practical settings and provide clear paths for ongoing progress.

REFERENCES

[1] A. Mostofi and V. Jain, “Inventory management and control of deteri- orating pharmaceutical products sing industry 4.0,” in 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA), pp. 394–400, IEEE, 2021.

[2] Y.-M. Su, H.-W. Peng, K.-W. Huang, and C.-S. Yang, “Image processing technology for text recognition,” in 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 1–5, IEEE, 2019.

[3] A. Deepa, S. Chinta, N. K. Ashili, B. S. Babu, R. R. Vydugula, and R. S. VSL, “An intelligent invoice processing system using tesseract ocr,” in 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), pp. 1–6, IEEE, 2024.

[4] C. Kaundilya, D. Chawla, and Y. Chopra, “Automated text extraction from images using ocr system,” in 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 145– 150, IEEE, 2019.

[5] K. V. Sakhare and I. Kulkarni, “Predictive analysis of end to end inventory management system for perishable goods,” in 2022 3rd International Conference for Emerging Technology (INCET), pp. 1–5, IEEE, 2022.

[6] K. Praveen, P. Kumar, J. Prateek, G. Pragathi, and J. Madhuri, “Inventory management using machine learning,” International Journal of Engineer- ing Research & Technology (IJERT), vol. 9, no. 06, pp. 866–869, 2020.

[7] N. Dhaliwal, P. K. Tomar, A. Joshi, G. S. Reddy, A. Hussein, and M. B. Alazzam, “A detailed analysis of use of ai in inventory management for technically better management,” in 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 197–201, IEEE, 2023.

[8] W. Saena and V. Suttichaya, “Predicting drug sale quantity using machine learning,” in 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI NLP), pp. 1–6, IEEE, 2019.

[9] S. A. Abhishek, S. Raj, and L. Malphedwar, “Evaluation of handwritten mathematical equations,” International Journal of Advanced scientific Research and engineering trends, 2004.

[10] Aditya Thorat, Sanket Solanke, Sumit Thakur, Mayur Tidke, & Aradhna Pawar. (2024). ADVANCED INVENTORY FOR MEDICAL DISTRIBUTOR (AIMD). *International Education and Research Journal (IERJ)*, *10*(11).

[11] Y. Cao, C. Liu, and W. Song, “Binarization of document images using local features and support vector machine,” IEEE Access, vol. 6, pp. 45966–45975, 2018.

[12] T. Saha et al., “Efficient document image denoising using deep wavelet autoencoder,” in Proc. ICDAR, IEEE, 2019, pp. 127–134.

[13] B. Shi et al., “Robust scene text recognition with automatic rectification,” in CVPR, IEEE, 2016.

[14] Y. Afifi and M. Hussain, “Contrast enhancement using adaptive gamma correction with weighting distribution,” IEEE Trans. Image Process., vol. 26, no. 5, pp. 2272–2284, 2017.

[15] T. Wang et al., “End-to-end text recognition with convolutional neural networks,” in ICPR, IEEE, 2018.

[16] C.-Y. Lee and S. Osindero, “Recursive recurrent nets with attention modeling for OCR,” in CVPR, IEEE, 2016.

[17] T. Y. Lin et al., “Feature pyramid networks for object detection,” in CVPR, IEEE, 2017.

[18] J. Zhao et al., “CUTIE: Learning to understand documents with convolutional universal text information extractor,” in AAAI, 2019.

[19] Z. Huang, K. Yu, and J. Qian, “Table detection in document images via modified YOLOv3,” in ICPR, IEEE, 2020.

[20] Y. Xu et al., “LayoutLMv2: Multi-modal pre-training for visually-rich document understanding,” in ACL, 2021.

[21] M. Denk and S. Reisswig, “BERTgrid: Contextualized embedding for 2D document representation and understanding,” in EMNLP, 2019.

[22] L. Yao et al., “Named entity recognition in electronic health records using transfer learning from language models,” JAMIA, vol. 26, no. 11, pp. 1389–1398, 2019.

[23] J. Devlin et al., “BERT: Pre-training of deep bidirectional transformers for language understanding,” in NAACL, 2019.

[24] R. Xu et al., “LayoutXLM: Multimodal pre-training for multilingual visually-rich document understanding,” in ACL, 2022.

[25] J. Gu et al., “FormNet: Structural encoding beyond sequential modeling in form understanding,” in ACL, 2022.

[26] Y. Li et al., “DocBank: A benchmark dataset for document layout analysis,” in COLING, 2020.