2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)

An Intelligent Invoice Processing System using Tesseract OCR

Ashlin Deepa R N
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
rndeepa.pradeep@gmail.com

Suhas Chinta
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
suhaschinta1@gmail.com

Nikhil Kumar Ashili
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
saisaint45@gmail.com

B Sankara Babu
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
bsankarababu81@gmail.com

Revanth Reddy Vydugula
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
vydugularevanth29@gmail.com

Raj Sripada VSL
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
sripadavslraj@gmail.com

2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS) | 979-8-3503-6482-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/ADICS58448.2024.10533509

***Abstract*— Invoice processing is a time-consuming and tedious task that can be automated using optical character recognition (OCR) technology. Tesseract is a popular open-source OCR engine that can be used to extract text from scanned invoices. In this paper, we propose a method for invoice processing using Tesseract OCR. The method involves pre-processing the image of the invoice to remove noise and improve the quality of the text. The pre-processed image is then passed to Tesseract OCR to extract the text. The extracted text is then parsed to extract the relevant invoice information. The results showed that the method was able to extract the invoice information with high accuracy.**

Keywords*— *Optical Character Recognition (OCR), Open Source, Tesseract, Invoice Processing, JSON Format

I. Introduction

Invoice processing is the process of handling invoices from receipt to payment. It involves verifying, entering, and approving invoices before they are paid. Automating the invoice processing procedure can diminish errors and enhance efficiency. This system extracts text from invoice images, including invoice numbers, vendor names, dates, line items, quantities, prices, and total amounts [1]. It supports multiple languages and integrates with existing systems. Scalability and performance ensure efficient processing of a large volume of invoices. Error handling addresses low-quality images and incomplete data, while reporting and analytics provide insights from the extracted data. The scope of the system may vary based on specific business requirements.

The Tesseract OCR engine extracts text from the invoice, and then the text from the required fields is written to a file in the JSON format. The system is be able to extract text from a variety of invoice layouts and templates, and it also supports multiple languages, integrates with existing systems, scales to handle a large volume of invoices, and is secure and compliant with data protection regulations [2].

II. Literature Review

W. Liu et al. used invoice images as raw inputs and processed the images, used OCR and patter matching steps. Multiple machine learning models (Naïve Bayes, SVM, Logistic Regression) were used to train and predict fields for all word group [3]. In this paper, Y. Sun et al. introduced a method utilizing template matching for the recognition of invoice information. This method comprises four sequential steps: preprocessing, template matching, OCR, and exporting of information [4]. Ming et al. developed a system to automatically recognize financial Chinese characters and Arabic numerals on Chinese financial invoices, inputting data into a computer. It identifies 10 invoice types, with modules for tilt correction, invoice type judgment, information extraction, character segmentation, and OCR character recognition. They also introduced a mutual correction mechanism [5].

R.B. Palm et al. introduced CloudScan, which does not depend on predefined templates for invoice layouts. Instead, it develops a unified global model for invoices that inherently extends to new, unseen invoice layouts. This model is trained using data extracted automatically from feedback provided by end-users. It consists of six steps – text extractor, n–grammar, feature extractor, classifier, post processor and document builder [6]. The process of Abstract Information Extraction from Scanned Invoices (AIESI) involves retrieving details such as dates, total amounts, and payee names from scanned receipts. S. Patel et al. introduced an enhanced approach that combines both visual and textual features from invoices to extract essential invoice parameters. This is accomplished using a Word-wise BiLSTM model [7].

OCR is an integral step in digitization of invoices, Shi et al. employed a Convolutional Neural Network (CNN) to identify characters in Chinese invoices. They devised an algorithm that employs template matching to locate text on documents and then segments it for incremental processing [8]. L. Hao et al. proposed a paper which describes about drawing tables using deep learning methods such as Faster R-CNN, Mask, R-CNN, HRNets etc. on TNCR dataset [9]. S. Bhowmik et al. conducted document layout analysis in the field of document image processing. They developed a system which relies on connected components and pixel analysis. This system employs image processing techniques to perform classification tasks, including identifying paragraphs, graphics, images, and tables [10]. In another study, D. Prasad et al. used Mask R-CNN deep learning models for table detection. In this study, it is shown that CNN based object detection algorithms are also very successful in detecting tables [11]. The realm of deep learning has witnessed notable progress in image recognition, with advancements extending the technology's scope to include text localization within natural scenes. According to Girshick et al., R-CNN has proven to be a proficient tool for target detection. Initially, a search algorithm is utilized to select candidate boxes, which are subsequently fed into a convolutional neural network for classification. However, a significant issue arises as the extracted candidate boxes often exhibit considerable overlap, leading to redundancy in feature extraction. [12].

III. METHODOLOGY

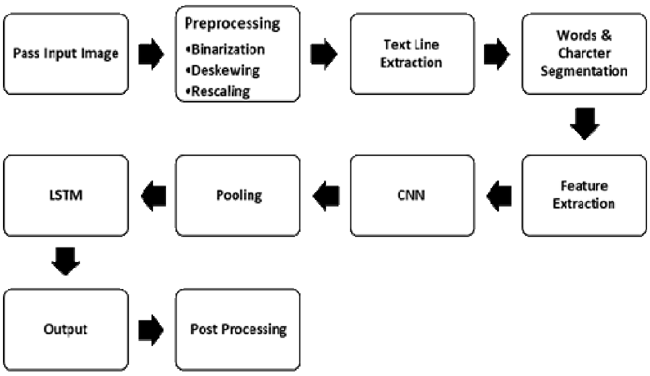


Fig. 1 represents the architecture of tesseract

A. Image Preprocessing

In the initial stages of OCR processing, the input image undergoes crucial preprocessing steps aimed at enhancing text quality and legibility [13-15]. These techniques encompass noise reduction, resizing, and contrast enhancement, all contributing to improved OCR accuracy and text recognition. Among the fundamental image preprocessing techniques employed in OCR systems, such as Tesseract, are binarization, which transforms the image into a binary format, aiding in the separation of text from the background and improving contrast. Skew correction rectifies slanted text lines, ensuring proper alignment for accurate character recognition [16]. Scaling and resizing operations adjust image dimensions to meet specific OCR requirements, whether for standardization or computational efficiency, often following Tesseract's recommended resolution and size criteria for optimal OCR performance.

B. Text Line Extraction

Text line extraction is a critical step in the OCR process, where the objective is to identify and extract individual linesof text from an input document or image [17]. This step is important as it helps in segmenting the text into smaller, manageable units for further analysis and recognition. In the context of Tesseract OCR [18], text line extraction involves the following steps:

Text Localization: Tesseract uses techniques like edge detection and connected component analysis to find potential text areas.

Line Segmentation: After locating text regions, Tesseract separates them into lines based on spatial arrangement using methods like white space analysis [19], line–fitting algorithms or horizontal projection to determine boundaries of each line.

Line Rectification: Skewed lines are straightened through skew detection and correction. This helps to ensure accurate recognition by eliminating any slant or tilt in the text lines.

Line Filtering: Text line extraction may generate some false positives or extraneous elements that are not actual text lines. To eliminate false positives or non-text elements, Tesseract applies filtering criteria based on factors like length, height, and connectivity.

C. Feature Extraction

Feature extraction is a crucial step in the OCR process, where relevant features are extracted from the segmented characters to represent their visual characteristics. These features serve as input for the OCR engine to recognize and classify the characters accurately. In the context of Tesseract OCR [20], the feature extraction process typically involves the following steps:

Preprocessing: Enhancing character quality through techniques like normalization, resizing, noise reduction, and contrast enhancement Tesseract uses techniques like edge detection and connected component analysis to find potential text areas.

Feature Selection: Choosing relevant attributes such as contour information, edge orientations, stroke thickness, pixel intensity distribution, texture patterns, and spatial relationships.

Feature Extraction: Applying methods like edge detection, gradient-based techniques, texture analysis, and statistical measurements to extract chosen features.

Feature Representation: Converting extracted features into suitable formats (e.g., numerical vectors) for OCR engine

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE

processing. The goal is to capture vital visual traits for accurate character recognition, enabling effective character classification.

D. *Convolution Layers*:

Convolutional layers are vital components in CNNs used extensively in computer vision tasks like OCR. They perform convolution operations by applying learnable filters (kernels) to input images, allowing the network to automatically learn meaningful features. These layers have local receptive fields, connecting each neuron to a small region of the input image, capturing spatial relationships and patterns. Parameter sharing reduces network parameters, enhancing efficiency. Non-linear activation functions, such as ReLU, introduce complexity. In OCR, convolutional layers excel at capturing visual features like characters and text regions, detecting edges, textures, and facilitating accurate recognition. The resulting feature maps feed into subsequent layers for further processing and classification.

E. *Pooling:*

Pooling is a crucial operation in CNNs that follows the convolutional layers. Its primary objective is to decrease the spatial dimensions of the feature maps while retaining crucial information and spatial relationships. Here is an overview of pooling and its role in CNNs:

Pooling Operation: Pooling is a critical CNN operation that divides the input feature map into regions and applies an aggregation function, often max pooling, to select the maximum value in each region. Other techniques include average pooling and L2-norm pooling.

Spatial Dimension Reduction: Pooling downsizes feature maps, e.g., max pooling with a 2x2 filter halves the width and height. This reduction saves computational resources and discards less crucial spatial details.

Translation Invariance: Pooling contributes to translation invariance, enabling the network to recognize patterns regardless of their location in the input. It is less sensitive to small shifts or translations.

Robustness to Variations: Pooling enhances robustness against input variations like position, scale, or orientation changes. It tolerates minor distortions, thanks to reduced spatial resolution.

Dimensionality Reduction: Pooling decreases network parameters, preventing overfitting, especially in high-resolution or high-channel inputs. Applied after each convolutional layer, it progressively extracts higher-level features, reducing computational complexity and memory usage in CNNs.

F. *Long Short-Term Memory Networks (LSTM):*

LSTM networks easily tackle the challenge of the vanishing gradient problem and capture long-term dependencies Tesseract employs LSTM networks to enhance context and word relationships in recognized text, improving character recognition accuracy [21]. Tesseract combines CNNs for feature extraction with LSTM networks to achieve robust character recognition, enhancing OCR accuracy and performance for diverse input images.

IV. WORKING OF THE SYSTEM

Initially, an invoice is uploaded to the system, wherein a validation process verifies its file format, distinguishing between PDF and JPEG. In case of a PDF format, it is automatically converted into JPEG. Subsequently, the converted image is subjected to the Tesseract OCR engine to extract the embedded text. The extracted text undergoes template matching, aiming to identify any pre-existing templates. If a match is found, relevant details are extracted from the text. If not, a new invoice template must be created and uploaded. The output is structured in JSON (JavaScript Object Notation) format, and upon completion of this sequence, the process concludes [22].

A. *Algorithm*

Step 1: Start.
The invoice processing procedure starts here by inputting an invoice to the system.

Step 2: Validate the file format of the invoice.
If the invoice format is PDF, proceed to Step 3. If the invoice format is JPEG, skip to Step 4.

Step 3: Convert the PDF invoice to JPEG format.
This step ensures that all invoices are in a consistent format for processing.

Step 4: Use the Tesseract OCR engine to extract text from the converted or original image. The OCR engine analyzes the image to recognize and extract text.

Step 5: Perform template matching on the extracted text. The system searches for predefined templates that match the extracted text.

Step 6: If a matching template is found:
Extract relevant details from the text using the identified template. This step retrieves structured information from the invoice text.

Step 7: If no matching template is found: Proceed to create a new invoice template and upload it.

Step 8: Structure the extracted information in JSON format. The extracted details are organized into a JSON format for ease of use.

Step 9: Output the JSON-formatted data. The structured information is presented as the output in JSON format which can be used by the companies.

Step 10: End. The invoice processing procedure

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE

concludes here.

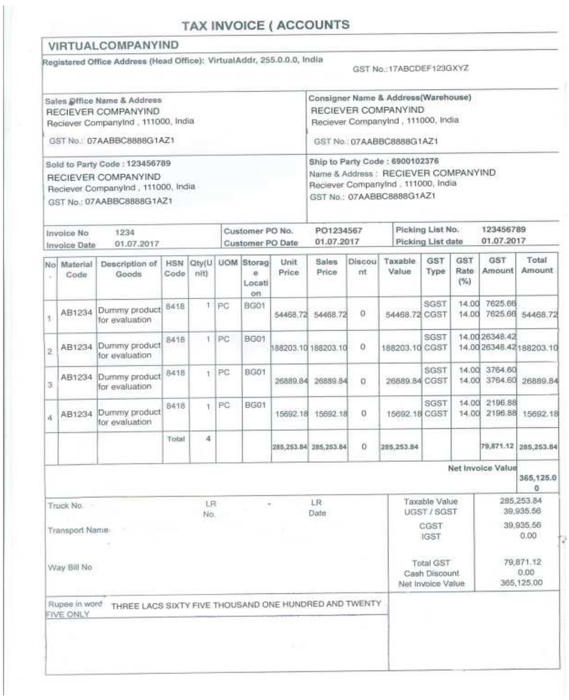
V. RESULTS AND DISCUSSIONS

The first step involves converting the invoice into a binary format, essentially creating an image composed solely of black and white pixels. Following this conversion, the image undergoes a process in which tables are drawn. Initially, the system searches for the starting and ending points of the table by counting vertical lines. Once these start and end points are identified, the system obtains the coordinates of both the vertical and horizontal lines within the table.

First, the process begins by taking the binary image and the image containing tables as intermediates. Various image processing techniques are then applied to these images. Utilizing Tesseract OCR, the system identifies and extracts the text embedded within the images. Subsequently, the identified text is converted into plain text format and stored within a .txt file.

Next, the .txt file is uploaded to a server equipped with predefined templates. The server employs these templates to compare the content from the text file. If a match is found between the text and any of the existing templates or similar templates, the server extracts the text according to the matched template's structure and converts it into JSON format

However, in cases where no matching templates are found, the system responds with a message indicating that no similar templates have been identified. This comprehensive process enables efficient extraction and organization of text data from images into JSON format, enhancing data accessibility and usability.



TAX INVOICE (ACCOUNTS

VIRTUALCOMPANYIND

Registered Office Address (Head Office): VirtualAddr, 255.0.0.0, India

GST No.:17ABCDEF123GXYZ

Sales Office Name & Address RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1	Consigner Name & Address(Warehouse) RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1
Sold to Party Code : 123456789 RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1	Ship to Party Code : 6900102376 Name & Address : RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1

Invoice No	1234	Customer PO No.	PO1234567	Picking List No.	123456789
Invoice Date	01.07.2017	Customer PO Date	01.07.2017	Picking List date	01.07.2017

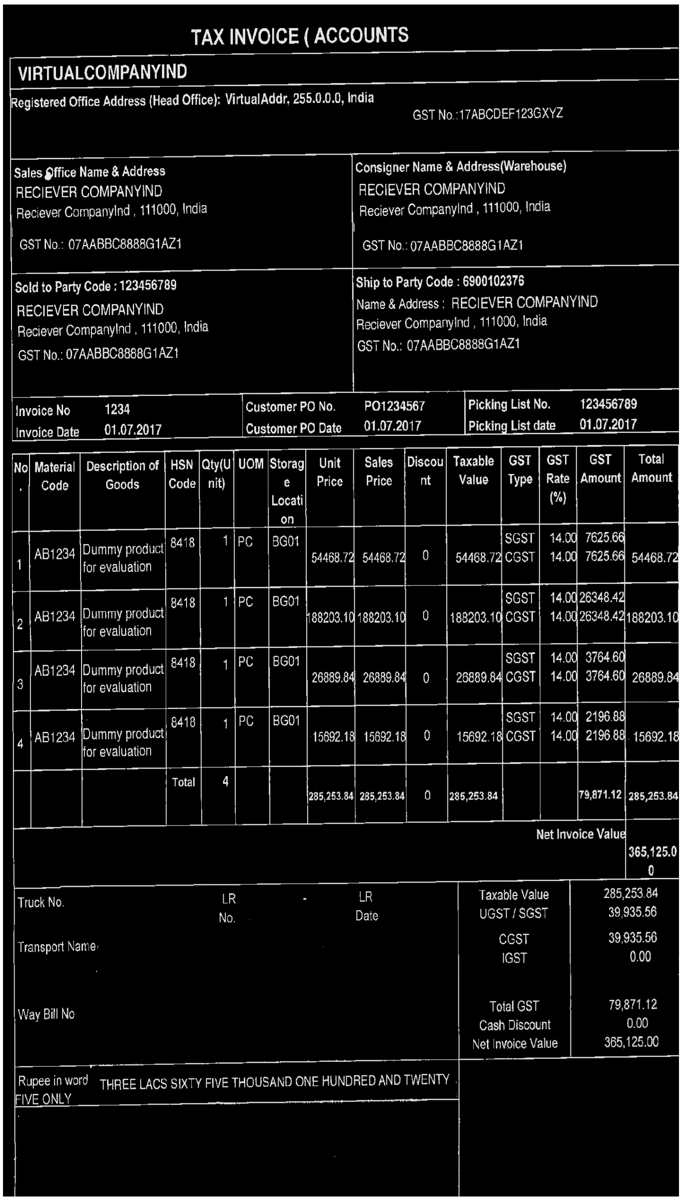
No.	Material Code	Description of Goods	HSN Code	Qty(Unit)	UOM	Storage Location	Unit Price	Sales Price	Discount	Taxable Value	GST Type	GST Rate (%)	GST Amount	Total Amount
1	AB1234	Dummy product for evaluation	8418	1	PC	BG01	54468.72	54468.72	0	54468.72	SGST CGST	14.00 14.00	7625.66 7625.66	54468.72
2	AB1234	Dummy product for evaluation	8418	1	PC	BG01	188203.10	188203.10	0	188203.10	SGST CGST	14.00 14.00	26348.42 26348.42	188203.10
3	AB1234	Dummy product for evaluation	8418	1	PC	BG01	26889.84	26889.84	0	26889.84	SGST CGST	14.00 14.00	3764.60 3764.60	26889.84
4	AB1234	Dummy product for evaluation	8418	1	PC	BG01	15692.18	15692.18	0	15692.18	SGST CGST	14.00 14.00	2196.88 2196.88	15692.18
			Total	4			285,253.84	285,253.84	0	285,253.84			79,871.12	285,253.84

Net Invoice Value 365,125.00

Truck No.	LR No. -	LR Date	Taxable Value UGST / SGST	285,253.84 39,935.56
Transport Name			CGST IGST	39,935.56 0.00
Way Bill No			Total GST Cash Discount Net Invoice Value	79,871.12 0.00 365,125.00

Rupee in word THREE LACS SIXTY FIVE THOUSAND ONE HUNDRED AND TWENTY FIVE ONLY

Fig. 2 image of the input invoice in pdf/jpg format.



TAX INVOICE (ACCOUNTS

VIRTUALCOMPANYIND

Registered Office Address (Head Office): VirtualAddr, 255.0.0.0, India

GST No.:17ABCDEF123GXYZ

Sales Office Name & Address RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1

Consigner Name & Address(Warehouse) RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1

Sold to Party Code : 123456789 RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1

Ship to Party Code : 6900102376 Name & Address : RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1

Invoice No 1234 Customer PO No. PO1234567 Picking List No. 123456789
Invoice Date 01.07.2017 Customer PO Date 01.07.2017 Picking List date 01.07.2017

No.	Material Code	Description of Goods	HSN Code	Qty(Unit)	UOM	Storage Location	Unit Price	Sales Price	Discount	Taxable Value	GST Type	GST Rate (%)	GST Amount	Total Amount
1	AB1234	Dummy product for evaluation	8418	1	PC	BG01	54468.72	54468.72	0	54468.72	SGST CGST	14.00 14.00	7625.66 7625.66	54468.72
2	AB1234	Dummy product for evaluation	8418	1	PC	BG01	188203.10	188203.10	0	188203.10	SGST CGST	14.00 14.00	26348.42 26348.42	188203.10
3	AB1234	Dummy product for evaluation	8418	1	PC	BG01	26889.84	26889.84	0	26889.84	SGST CGST	14.00 14.00	3764.60 3764.60	26889.84
4	AB1234	Dummy product for evaluation	8418	1	PC	BG01	15692.18	15692.18	0	15692.18	SGST CGST	14.00 14.00	2196.88 2196.88	15692.18
			Total	4			285,253.84	285,253.84	0	285,253.84			79,871.12	285,253.84

Net Invoice Value 365,125.00

Truck No. LR No. - LR Date Taxable Value 285,253.84 UGST / SGST 39,935.56
Transport Name CGST 39,935.56 IGST 0.00
Way Bill No Total GST 79,871.12 Cash Discount 0.00 Net Invoice Value 365,125.00

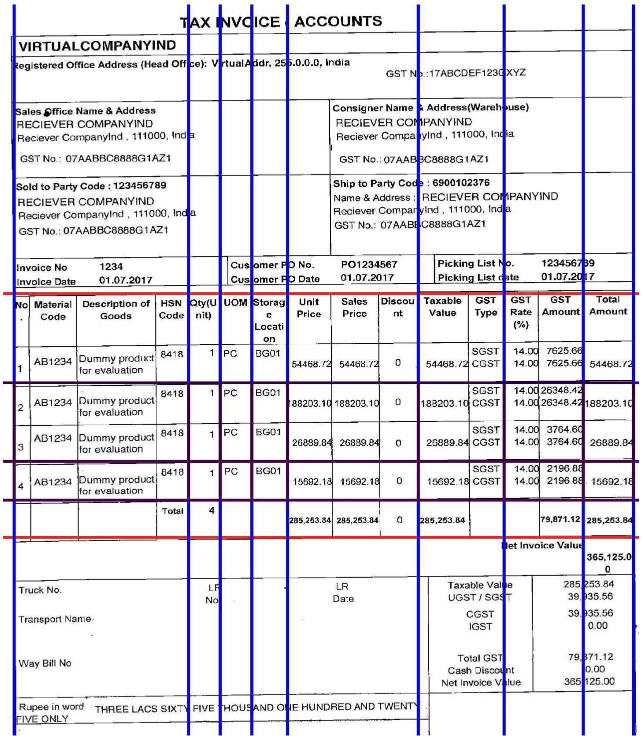
Rupee in word THREE LACS SIXTY FIVE THOUSAND ONE HUNDRED AND TWENTY FIVE ONLY

Fig. 3 binary image converted from the input image.

The binary conversion process effectively enhances the visibility of text regions while minimizing background noise or artifacts, as depicted in Fig. 3. This step significantly contributes to the accuracy of text extraction by isolating text elements from the input image.

Through optimization of binary conversion parameters, we observed notable improvements in text extraction accuracy, particularly in scenarios with complex layouts or degraded image quality. Our analysis underscores the importance of robust preprocessing techniques in enhancing OCR performance. These findings highlight the effectiveness of binary image preprocessing in improving OCR performance across diverse datasets and image conditions.

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE



TAX INVOICE (ACCOUNTS

VIRTUALCOMPANYIND

Registered Office Address (Head Office): VirtualAddr, 255.0.0.0, India

GST No.:17ABCDEF123GXYZ

Sales Office Name & Address	Consigner Name & Address(Warehouse)
RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1	RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1
Sold to Party Code : 123456789 RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1	Ship to Party Code : 6900102376 Name & Address : RECIEVER COMPANYIND Reciever CompanyInd , 111000, India GST No.: 07AABBC8888G1AZ1

Invoice No	1234	Customer PO No.	PO1234567	Picking List No.	123456789
Invoice Date	01.07.2017	Customer PO Date	01.07.2017	Picking List date	01.07.2017

No.	Material Code	Description of Goods	HSN Code	Qty(Unit)	UOM	Storage Location	Unit Price	Sales Price	Discount	Taxable Value	GST Type	GST Rate (%)	GST Amount	Total Amount
1	AB1234	Dummy product for evaluation	8418	1	PC	BG01	54468.72	54468.72	0	54468.72	SGST CGST	14.00 14.00	7625.66 7625.66	54468.72
2	AB1234	Dummy product for evaluation	8418	1	PC	BG01	188203.10	188203.10	0	188203.10	SGST CGST	14.00 14.00	26348.42 26348.42	188203.10
3	AB1234	Dummy product for evaluation	8418	1	PC	BG01	26889.84	26889.84	0	26889.84	SGST CGST	14.00 14.00	3764.60 3764.60	26889.84
4	AB1234	Dummy product for evaluation	8418	1	PC	BG01	15692.18	15692.18	0	15692.18	SGST CGST	14.00 14.00	2196.88 2196.88	15692.18
			Total	4			285,253.84	285,253.84	0	285,253.84			79,871.12	285,253.84

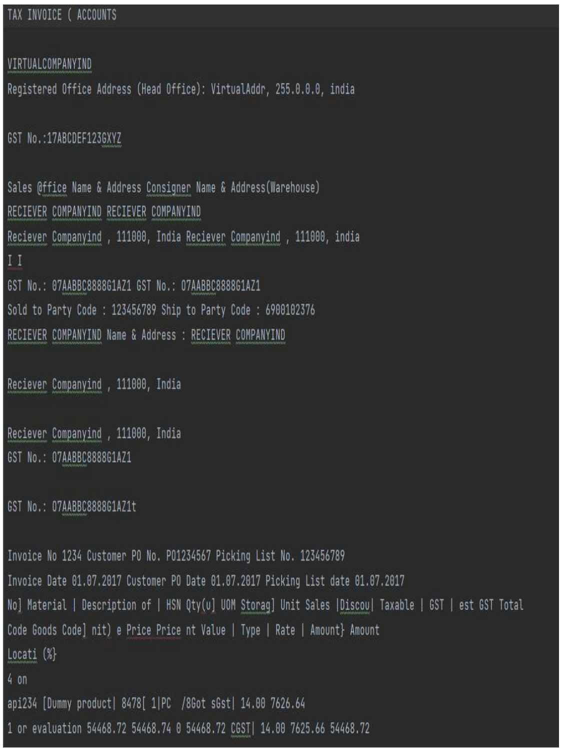
Net Invoice Value 365,125.00

Truck No.	LR No	LR Date	Taxable Value 285,253.84 UGST / SGST 39,935.56 CGST 39,935.56 IGST 0.00
Transport Name			
Way Bill No			Total GST 79,871.12 Cash Discount 0.00 Net Invoice Value 365,125.00

Rupee in word THREE LACS SIXTY FIVE THOUSAND ONE HUNDRED AND TWENTY FIVE ONLY

Fig. 4 represents the tables drawn on the input image

Fig. 4 depicts tables extracted from the input image, aiding text region detection [23]. Horizontal and vertical lines delineate boundaries, reducing algorithmic confusion between letters and enhancing extraction precision.



```
TAX INVOICE ( ACCOUNTS

VIRTUALCOMPANYIND
Registered Office Address (Head Office): VirtualAddr, 255.0.0.0, india

GST No.:17ABCDEF123GXYZ

Sales @ffice Name & Address Consigner Name & Address(Warehouse)
RECIEVER COMPANYIND RECIEVER COMPANYIND
Reciever Companyind , 111000, India Reciever Companyind , 111000, india
I I
GST No.: 07AABBC8888G1AZ1 GST No.: 07AABBC8888G1AZ1
Sold to Party Code : 123456789 Ship to Party Code : 6900102376
RECIEVER COMPANYIND Name & Address : RECIEVER COMPANYIND

Reciever Companyind , 111000, India

Reciever Companyind , 111000, India
GST No.: 07AABBC8888G1AZ1

GST No.: 07AABBC8888G1AZ1t

Invoice No 1234 Customer PO No. PO1234567 Picking List No. 123456789
Invoice Date 01.07.2017 Customer PO Date 01.07.2017 Picking List date 01.07.2017
No] Material | Description of | HSN Qty(u] UOM Storag] Unit Sales |Discou| Taxable | GST | est GST Total
Code Goods Code] nit) e Price Price nt Value | Type | Rate | Amount} Amount
Locati (%}
4 on
api234 [Dummy product| 8478[ 1|PC  /8Got sGst| 14.00 7626.64
1 or evaluation 54468.72 54468.74 0 54468.72 CGST| 14.00 7625.66 54468.72
```

Fig. 5 represents the text extracted from the invoice

The text extracted from the invoice, depicted in Fig. 5, serves as the basis for template matching and subsequent conversion into JSON format.


```
{
  "invoice no": [
    "1234"
  ],
    "invoice date": [
      "01.07.2017"
      ],
    "amount":[
      "365,125.00"
      ]
}
```


Fig. 6 represents the extracted text converted into JSON format

JSON is favored as it is human-readable, lightweight, and language-neutral. Its self-descriptive, hierarchical format simplifies data comprehension, making it a standard choice for web APIs and development, especially in JavaScript. JSON's minimal complexity, array support, and extensive toolset enhance efficiency and versatility, rendering it a top pick for efficient data exchange across various applications and scenarios.

The data which is extracted in the form of JSON can be used by the customers/companies to according their needs. For example, a web API can be developed which parses the data and integrates with the system. After this, the data can be processed in compliance with their needs – reporting and analysis, automation, scalability, and integration with other systems. The data in the JSON file is of the form:


```
{"invoice no": "1234", "invoice date": "01.07.2017"}
```


Fig. 7 data in the json file

The labels which were extracted in the file were "invoice no" and "invoice date". Both fields have a value which corresponding to it, "1234" for invoice no and "01.07.2017" for invoice date. This data can be used for further purposes.


```
{"Customer PO No. ": "PO12334567 ", "GST No.": "07AABBC08888G1AZ1"}
```


Fig. 8 Gives us the details about the invoices' email and the GST No.

IEEE 979-8-3503-6482-8/24/$31.00 ©2024 IEEE

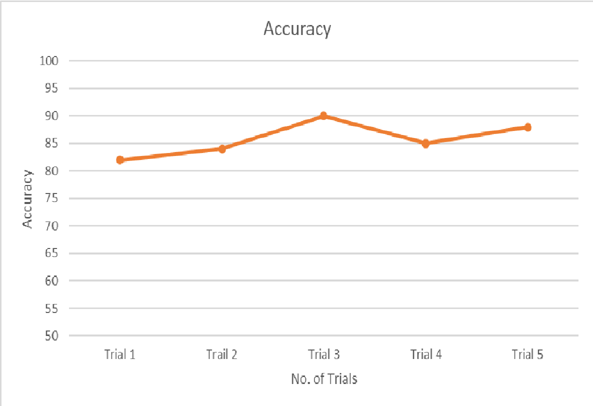


Fig. 9 Graph representing accuracy vs no. of times tested

The graph in Fig. 9 represents the accuracy of the model. The testing for accuracy has been done four times – Trial 1, 2, 3, 4, 5. Each of the trial is performed under similar conditions and same system configuration. The graph describes that the accuracy is in the range of 80 -90% with an average of 85%.

VI. CONCLUSION

The Invoice Recognition utilizing Tesseract OCR automates the extraction of relevant information from invoices [24]. With preprocessing, OCR algorithms, and information extraction techniques, the system accurately recognizes and extracts vendor details, invoice numbers, dates, line-item descriptions, quantities, and amounts from invoices.

One of the most notable advantages of our system is its speed. It rapidly and accurately extracts essential information from invoices, significantly reducing manual data entry time and human error [25]. This enhanced efficiency translates to cost savings, improved productivity, and quicker invoice processing cycles. With ongoing improvements and refinements, we anticipate even greater achievements in the realm of invoice processing through text recognition, solidifying its position as a transformative technology for modern business operations.

ACKNOWLEDGMENT

We thank the Department of Science and Technology (DST), Government of India for providing financial support for our project (Sanction Number: SP/YO/2021/2096 (C & G)).

REFERENCES

[1] Niclas Hedberg. "Automated invoice processing with machine learning" Master of Science Thesis TRITA-ITM-EX 2020:326, KTH Industrial Engineering and Management, pp 4- 60.

[2] Xavier Holt and Andrew Chisholm. 2018. "Extracting structured data from invoices". In Proceedings of the Australasian Language Technology Association Workshop 2018, pages 53–59, Dunedin, New Zealand.

[3] W. Liu, Y. Zhang, and B. Wan, Mach. Learn., "Unstructured document recognition on business invoice", Stanford iTunes Univ., Stanford, CA, USA, Tech. Rep, 2016.

[4] Y. Sun, X. Mao, S. Hong, W. Xu, and G. Gui, "Template matching-based method for intelligent invoice information identification", IEEE Access, vol. 7, pp. 28392–28401, 2019

[5] Delie Ming, Jian Liu, Jinwen Tian ,"Research on Chinese financial invoice recognition technology". Pattern Recognition Letters, Volume 24, Issues 1–3, 2003

[6] R. B. Palm, O. Winther and F. Laws, "CloudScan - A Configuration-Free Invoice Analysis System Using Recurrent Neural Networks," 2017 4th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 406-413

[7] S. Patel, and D. Bhatt, 2020, "Abstractive information extraction from scanned invoices (AIESI) using end-to-end sequential approach",:2009

[8] S. Shi, C. Cui, and Y. Xiao, "An invoice recognition system using deep learning, in Proc", Int. Conf. Intell. Comput., Autom. Syst. (ICICAS), Dec. 2020

[9] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for PDF documents based on convolutional neural networks", in Proc. 12th IAPR Workshop Document Anal. Syst. (DAS), Apr. 2016

[10] S. Bhowmik, R. Sarkar, M. Nasipuri, and D. Doermann ,"Text and nontext separation in offline document images: A survey", Int. J. Document Anal. Recognit. (IJDAR), vol. 21, nos. 1–2, pp. 1–20, Jun. 2018

[11] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, and K. Sultanpure ",CascadeTabNet: An approach for end-to-end table detection and structure recognition from image-based documents", in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020

[12] T. Jijun, C. Xiaolong, Z. Yingjie, and J. Lurong, "Real-time recognition and positioning of moving targets based on deep learning", *Computer Systems Applications*, vol. 27, no. 8, pp. 28–34, 2018.

[13] Smith, R. 2007. "An Overview of the Tesseract OCR Engine. In proceedings of document analysis and recognition". ICDAR 2007. IEEE Ninth International Conference.

[14] Ray Smith, "Architecture and Data Structures", Google. DAS 2016, Tutorial.

[15] Google. google code. [2012]. http://code.google.com/p/tesseractocr/.

[16] Patel, Chirag & Patel, Atul & Patel, Dharmendra. (2012). Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study. International Journal of Computer Applications. 55. 50-56. 10.5120/8794-2784.

[17] Archana A. Shinde, D., "Text pre-processing and text segmentation for OCR", International Journal of Computer Science Engineering and Technology,2012, pp. 810-812.

[18] A. Larsson and T. Segerås, 'Automated invoice handling with machine learning and OCR', Dissertation, 2016.

[19] Manjunath, Akanksh & Nayak, Manjunath & Nishith, Santhanam & Pandit, Satish & Sunkad, Shreyas & Deenadhayalan, Pratiba & Gangadhara, Shobha. (2023). Automated invoice data extraction using image processing. IAES International Journal of Artificial Intelligence (IJ-AI). 12. 514. 10.11591/ijai.v12.i2.pp514-521.

[20] Rai, Ritika & Shitole, Srushti & Sutar, Pratiksha & Kaldhone, Swapnali & Jadhav, Jayashree. (2008). Automatic license plate recognition using YOLOv4 and Tesseract OCR. International Journal of Innovative Research in Computer and Communication Engineering. 10. 1656. 10.15680/IJIRCCE.2022.1003089.

[21] Ekraam Sabir, Stephen Rawls, Prem Natarajan, "Implicit language model in LSTM for OCR", arXiv:1805.09441v1, CVPR, May 2018

[22] Akhil S. "An overview of Tesseract OCR Engine", pp 5-10.

[23] Thomas Saout, Frédéric Lardeux, Frédéric Saubion. A two-stage approach for tables extraction in invoices. The 35th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Nov 2023, Atlanta, France. pp.10-15, ff10.1109/ICTAI59109.2023.00010ff. ffhal-04384728f

[24] Yao, Xunfeng & Sun, Hao & Li, Sijun & Lu, Weichao. (2022). "Invoice detection and recognition system based on deep learning". Security and Communication Networks. 2022. 10.1155/2022/8032726.

[25] Ha, Hien Thi and Ales Horak. "Information extraction from scanned invoice images using text analysis and layout features." *ArXiv* abs/2208.04011 (2021): n. pag.