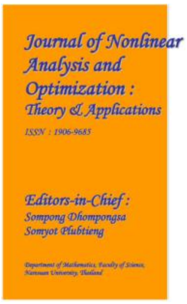
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.10 : 2024
ISSN :**1906-9685**



Journal of Nonlinear Analysis and Optimization : Theory & Applications
ISSN : 1906-9685
Editors-in-Chief : Sompong Dhompongsa Somyot Plubtieng
Department of Mathematics, Faculty of Science, Naresuan University, Thailand

QUICK CLAIM-AUTOMATED INVOICE EXTRACTION

DR.G.PRASANNA PRIYA, ASSISTANT PROFESSOR Of Computer Science Department (CSE),Vignan's Institute of Information Technology, Visakhapatnam, India

K.J.SAI SANKAR,Student Department of Computer Science & Engineering,
K.PAUL RAJU, Student Department of Computer Science & Engineering,
G.R.PAVAN SRIVATSA, Student Department of Computer Science & Engineering,
G.ADIKESAVA REDDY, Student Department of Computer Science & Engineering,
G.JAYARAM, Student Department of Computer Science & Engineering,
Vignan's Institute of Information Technology (A), Visakhapatnam, Andhra Pradesh, India

ABSTRACT: Quick Claim deals with making the bill reimbursement process. By using these clients can easily upload the bills on the internet so that the invoice clerks can easily track the bills understand the expenses in a better way and deliver the amount to the client. It avoids the manual work and makes the process simple and fast. In today's world, it is difficult for clients to submit bills to the banks and get a refund. The client needs to carry the bills to the invoice clerk and submit all the corresponding essential details. So, to make this process easy for the clients the Quick Claim is brought through which one can easily upload the bills and get the refunds.

The primary aim is to automate the extraction of key information from the invoices, such as vendor details, dates, amounts, and itemized expenses. The project seeks to integrate OCR capabilities seamlessly into the workflow, allowing for efficient data extraction and subsequent storage of the extracted information in a secure database. The goal is to create a user-friendly solution that enhances the accuracy and speed of expense reimbursement processes within the organization. The workflow for the expense reimbursement tool project encompasses multiple sequential stages, commencing with the acquisition of invoice imagery and culminating in the storge of parsed within a database.

1. INTRODUCTION

This system helps companies by automatically reading and understanding information from invoices. It uses a tool called OpenCV to find words on the invoices and then a system called OCR to change those words into text the computer can understand. There's also a website where you can upload invoices and see all the important details neatly organized. This saves a lot of time and effort because people don't have to manually type in all the information. Plus, it can help catch any funny business, like if someone is trying to trick the company with fake invoices or numbers. Large and medium scale companies deal with a large number of invoices on a daily basis. The companies need to keep track of the products being sold through the invoices, the cash flow generated, taxes which are paid and other product related analytics which are useful for their business. Companies have a dedicated set of data entry staff who manually enter these details into the systems or databases. This is a laborious task and time consuming. This entire process can be automated with the help of a system which extracts the data from these invoices and updates the databases with the product details. In general, all invoices have a standard template consisting of the seller company's name and address as well as the consumer's name and address. This is followed by an invoice number which may be of a different format as adopted by the company. Then there is a table with product details followed by the transactional details like tax and total amount. This would help design a system which provides flexibility to the user in selecting the fields of interest, based on their use case. Hence, such a system which aims to capture this functionality of invoices and provide automation resulting in saving countless hours of human effort will be described in our paper. There are two parts to the system, an opensource computer vision library (OpenCV)

2.REVIEW OF LITERATURE

Several The literature surrounding automatic invoice extraction utilizing optical character recognition (OCR) technology encompasses a diverse range of studies and advancements. Initially introducing OCR's pivotal role in converting scanned documents into machine-readable text, the review delves into the various techniques employed, from traditional template-based methods to modern deep learning approaches. Specific focus is directed towards studies and projects dedicated to invoice extraction, highlighting methodologies, algorithmic implementations, and performance evaluations. Common challenges such as invoice layout variations and language complexities are identified, prompting discussions on recent innovations aiming to overcome these obstacles. Furthermore, the integration of OCR with complementary technologies like natural language processing (NLP) and machine learning is explored, showcasing synergistic potentials for enhanced extraction accuracy.

In addition to the points, it's crucial to address the practical implications of automatic invoice extraction using OCR technology, including its potential to reduce manual labor, minimize errors, and accelerate invoice processing cycles for businesses of all sizes. Furthermore, considerations should be given to the regulatory landscape and compliance requirements, particularly in industries with stringent data privacy and security regulations. Additionally, the review may discuss the scalability of OCR solutions and their adaptability to evolving business needs, emphasizing the importance of flexible and customizable approaches to accommodate diverse invoice formats and organizational workflows. Moreover, insights into the cost-effectiveness and return on investment (ROI) of implementing OCR-based invoice extraction systems can provide valuable guidance for decision-makers seeking to optimize operational efficiency and resource utilization within their organizations.

2. PROPOSED SYSTEM

The proposed system focuses on developing an expense reimbursement tool using Python Flask and MongoDB, with integration of the OCR API from asprise.com. It will begin by setting up a Flask application to handle web requests and responses, utilizing MongoDB as the database for storing user and invoice data. The key feature will be the integration of the asprise.com OCR API, enabling automatic extraction of invoice details. User authentication and authorization will be implemented to ensure security, while user-friendly interfaces for both mobile and web platforms will streamline the reimbursement process. Robust data security measures will safeguard sensitive financial information. Thorough testing and quality assurance will be conducted before deployment, with ongoing maintenance to support future enhancements. Overall, the project aims to meet the evolving needs of clients and organizations by modernizing and optimizing the expense reimbursement process through a secure and efficient online platform.

3. METHODOLOGY

1. Planning

- **Objective Definition:** Define the goals of the system. This might include automating the creation of invoices based on customer orders, integrating with existing accounting software, and ensuring compliance with financial regulations.
- **Scope:** Determine what functionalities the system will include, such as generating invoices, sending reminders, handling different tax rates, integrating with payment gateways, etc.

2. Requirements Gathering

- **Functional Requirements:** List all the functions the system should perform, like user authentication, data entry, invoice generation, invoice distribution, etc.
- **Non-functional Requirements:** Identify requirements related to usability, performance (e.g., processing time for generating and sending an invoice), security (e.g., data encryption, secure access), and scalability.
- **Regulatory Requirements:** Ensure the system complies with relevant financial and data protection laws (like GDPR, HIPAA in healthcare, or specific tax authority requirements in different countries).

3. System Design

- **Architecture Design:** Design the overall system architecture, considering whether it will be a web-based application, a desktop application, or integrated within an existing ERP system.
- **Database Design:** Plan the database schema that will store user data, product information, tax rates, invoice records, payment status, etc.
- **User Interface Design:** Sketch the user interface for different types of users (e.g., admin panels for backend users and dashboard for clients).
- **API Integration:** Design how the system will integrate with other software, such as CRM systems, accounting software, or payment gateways.

4. Implementation

- **Environment Setup:** Set up the development, testing, and production environments.
- **Development:** Write code according to the design specifications. Use programming languages and frameworks that suit the project requirements, like Python for backend (with Django or Flask), and React or Angular for frontend.
- **API Integration:** Implement integration with third-party services (e.g., payment gateways, email service providers for sending invoices).

5. Testing

- **Unit Testing:** Test individual components to ensure they function correctly in isolation.
- **Integration Testing:** Test the interaction between different components and external integrations.
- **System Testing:** Evaluate the system as a whole to ensure it meets all the specified requirements.
- **User Acceptance Testing (UAT):** Allow end-users to test the system and provide feedback to ensure the system meets their needs and expectations.

6. Deployment

- **Deployment Strategy:** Decide on a deployment model (e.g., on-premises, cloud-based) and execute it.
- **Monitoring Tools Setup:** Implement tools to monitor system performance and errors to ensure ongoing stability.
- **User Training:** Provide necessary training and documentation to the users.

7. Maintenance and Updates

- **Regular Updates:** Regularly update the system software to patch vulnerabilities, update dependencies, and introduce new features.
- **Feedback Loop:** Establish a feedback mechanism to continuously improve the system based on user feedback and changing requirements.

5.OUTPUT AND FUTURE SCOPE

The output of an automatic invoice generation project includes a comprehensive system that automates the creation and distribution of invoices based on predetermined triggers and user actions. Here's what you can expect from a fully implemented system:

1. **Automated Invoice Creation:** The system automatically generates invoices when sales or service transactions are completed, using predefined templates and filling in details such as customer information, product or service descriptions, prices, taxes, and totals.
2. **User-Friendly Interface:** A clean and intuitive user interface that allows users (typically business staff or managers) to manage invoices, customer data, and view reports on sales, outstanding invoices, and received payments.
3. **Integration with Existing Systems:** Seamless integration with existing CRM, ERP, or accounting software, facilitating the smooth flow of data across platforms without the need for manual entry.
4. **Compliance and Accuracy:** Ensures that all invoices are compliant with local tax laws and regulations. The system also reduces human error, enhancing the accuracy of financial data.
5. **Real-time Reporting and Analytics:** Advanced reporting capabilities that allow users to generate real-time financial reports and analytics, helping in better financial decision-making.
6. **Scalability:** A system designed to handle increasing amounts of work or to be capable of expansion to accommodate growth in users, data volume, and transaction frequency.
7. **Security Features:** Robust security measures to protect sensitive financial data, including encryption, secure access controls, and regular security audits.
8. **Notification System:** An automated alert system for upcoming due payments, overdue invoices, and other relevant financial notifications, both for internal users and customers.

The future enhancements and extensions of an automatic invoice generation system can include several innovative and value-adding features:

1. **Machine Learning for Predictive Analytics:** Implementing machine learning algorithms to predict future sales trends, optimize invoice payment terms based on customer payment behavior, and enhance fraud detection.
2. **Blockchain for Invoice Verification:** Using blockchain technology to create immutable records of invoices, which can help in preventing fraud and ensuring the authenticity of transactions in a transparent manner.
3. **Advanced Customization Options:** Allowing more customized invoicing solutions, including personalized invoice design, flexible billing cycles, and dynamic pricing models based on customer engagement and loyalty.
4. **Mobile Compatibility:** Developing mobile applications or enhancing mobile compatibility to allow users to manage invoicing on-the-go, facilitating faster responses and approvals, thus speeding up the payment cycle.
5. **Artificial Intelligence (AI) Chatbots:** Implementing AI-driven chatbots to assist customers and staff with common inquiries about billing, invoice status, and basic troubleshooting, improving customer service.

5. CONCLUSION

In conclusion We have performed the character recognition using mainly the two methods of machine learning. Both the methods are highly efficient for the English language recognition. We have compared the efficiency and accuracy of both the methods, they have slight difference between the accuracy and efficiency Neural network have edge over the correlation and template matching method. When we use the template matching for the recognition of the characters it is highly accurate for few of the English language fonts used widely in the world. The method we have used here has various usages across the world it is used in voice convertors, text reading form the number plates etc. are its main uses. It has been also used for the text form which is used in the postal services and sales tax departments, it is highly efficient for that use When we do the character recognition in offline mode it takes some as it take times to get complied and it also have some problem in trouble shooting Which is a huge problem with the off line character recognition.

Hence we have done the work on the online character recognition using the neural network system and its algorithm back propagation theorem. Which was highly efficient and accurate than the previous methods, but still it's not 100% accurate for all hand written characters. Which is an issue with this system, it is not completely accurate for all the writing styles in English language. So in future we have to work upon the accuracy of the system and accuracy of the system. And there is need of work that should be done on the regional languages of India. They are still unexplored, for example the languages of upper north for them we can make voice recognizing software and it can convert that language input into the other language like English

REFERENCES:

[1] G. Bradski, “The OpenCV library,” Dr. Dobb’s Journal of Software Tools, vol. 120, pp. 122–125, 2000.

[2] S. Deshpande and R. Shriram, “Real time text detection and recognition on hand held objects to assist blind people,” in 2016International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Sep. 2016, pp. 1020–1024.doi: 10.1109/ICACDOT.2016.7877741.

[3] X. Zhou et al., “EAST: an efficient and accurate scene text detector,” in 2017 IEEE Conference on Computer Vision and PatternRecognition (CVPR), Jul. 2017, pp. 2642–2651. doi: 10.1109/CVPR.2017.283.

[4] K. Karthick, K. B. Ravindrakumar, R. Francis, and S. Ilankannan, “Steps involved in text recognition and recent research in OCR;a study,” International Journal of Recent Technology and Engineering, vol. 8, no. 1, pp. 3095–3100, 2019.

[5] J. Yuan, H. Li, M. Wang, R. Liu, C. Li, and B. Wang, “An OpenCV-based framework for table information extraction,” in 2020IEEE International Conference on Knowledge Graph (ICKG), Aug. 2020, pp. 621–628. doi: 10.1109/ICBK50248.2020.00093.

[6] R. C. Gonzalez, R. E. Woods, and S. L, “Morphological pmage processing,” 2004.

[7] “MATLAB, version 7.10.0,” in Natick, Massachusetts: The MathWorks Inc, 2010.

[8] C. Bartz, H. Yang, and C. Meinel, “STN-OCR: a single neural network for text detection and text recognition,” 2017.