Automating Invoice Processing via Optical-Character Recognition and Binary Classification

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

William Peter Greig

Spring 2024

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Kathryn A. Neeley, Department of Engineering and Society

Briana Morrison, Department of Computer Science

Automating Invoice Processing via Optical-Character Recognition and Binary Classification

CS4991 Capstone Report, 2024

William Greig
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
wpg6zmk@virginia.edu

ABSTRACT

Axcelis Technologies Inc., a semiconductor equipment manufacturer, decided to streamline and reduce manual invoice processing by utilizing an automated system. To streamline this process, I developed an automated invoice processor by leveraging optical-character recognition and a binary classifier to identify and extract part numbers from a variety of invoices. In my solution, I used Python and Adobe Cloud to build a software program able to sift through PDFs, CSVs, and Word documents. I used pyinstaller to create a packaged executable that could be run on any computer to ensure other Axcelis employees could use the invoice processor. When tested on pre-alerts, this software was effective in collecting part numbers from invoices with high accuracy and was dramatically faster than manual processing. Future work includes reducing non-part numbers such as phone numbers and shipping metrics, which were mistakenly extracted.

1. INTRODUCTION

Axcelis Technologies Inc., an ion-implanter manufacturer based in Massachusetts, receives 15-30 daily invoices and pre-alerts. Invoices detail shipment information about manfacturing parts and come from a variety of suppliers and customers. Axcelis' Trade Compliance department actively use these invoices to update their database for classification and tracking purposes. Crucially, because these documents come from various companies, there is no standardized format. Some invoices are Word Documents while others are scanned PDFs or Excel spreadsheets. Additionally, part numbers do not have a standard format. Part numbers can range from 6-15 characters, usually containing numbers and letters. Within any given invoice, there are other numbers such as phone numbers and financial numbers that, without context, can be mistaken for part numbers.

Problematically, the variation in document formatting forced Axcelis to process these invoices manually. A team member would need to open each invoice individually, extract the part numbers into a separate spreadsheet, and subsequently upload them directly into a database. This process was both inefficient and time-consuming. On average, Axcelis spent 1-2 hours daily processing these invoices.

As an intern at Axcelis, I was tasked with designing and implementing a software solution to reduce this manual processing work. In my approach, I designed a multi-stage processing system to automatically identify part numbers from each invoice. I used Google Tesseract's optical character recognition library recognition (OCR) to help extract text information from each invoice regardless of its formatting. To distinguish part numbers from normal numbers, I employed a binary classifier to mark potential part numbers. The objective of this project is to reduce the amount of time spent on manual processing and enable employees to focus on other items.

2. RELATED WORKS

Building automation software to extract information from documents is common but involves a nuanced understanding of the underlying extracted data. The development of an effective processing system necessitates the use of optical character recognition—a pattern recognition machine learning approach used to identify a variety of characeters including letters, digits, and syntactic symbols from images. There are many OCR-powered systems such as Adobe's document conversion.

There are widely available OCR engines to use for development purposes. One such engine is Google Tesseract—a powerful open-source OCR engine built and sponsored by Google. I opted to use Google Tesseract as opposed to other engines due to its high accuracy compared to other models (Vawani, 2017). I utilized AI researchers' insight into identifying patterns of desired metrics to better design an heuristic filter, and applying more effective machine learning classifiers (Kazakos, 2021). For a practical implementation, I followed and adapted Adrian Rosebrock's tutorial on scanning receipts using OCR to instead scan through invoices (Rosebrock, 2021).

3. PROJECT DESIGN

This section includes an overview of the automated invoice processor, a detailed explanation of technical components, and a discussion of software requirements and design limitations.

3.1 System Architecture Overview

A notated system diagram of the automated invoice processor is shown in Figure 1. As a general overview, various file formats and documents are fed into the invoice processor. As shown in Figure 2, depending on the type of file, a different processing approach would be taken to properly extract text information from the invoice. The resulting raw text data is passed through a part identifier which is comprised of a binary classifier—indicating whether a series of characters is a part number or not. All the identified part numbers are then listed on a spreadsheet to be uploaded or classified in Axcelis' database.

Figure 1: Automated Invoice Processor System Overview

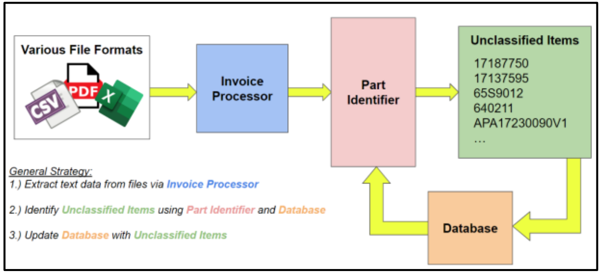
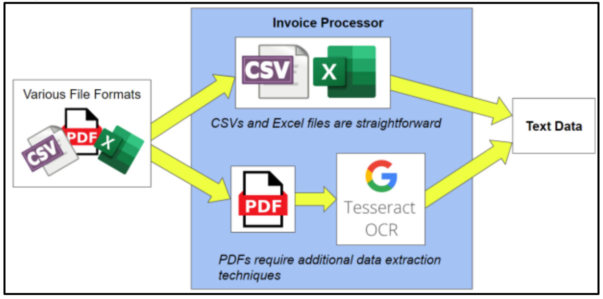

Various File Formats
CSV
PDF
X
Invoice Processor
Part Identifier
Unclassified Items
17187750
17137595
65S9012
640211
APA17230090V1
...
Database
General Strategy:
1.) Extract text data from files via Invoice Processor
2.) Identify Unclassified Items using Part Identifier and Database
3.) Update Database with Unclassified Items

Figure 2 : Invoice Processor Detailed View


Various File Formats
CSV
PDF
X
Invoice Processor
CSV
X
CSVs and Excel files are straightforward
PDF
Tesseract OCR
PDFs require additional data extraction techniques
Text Data

3.2 Requirements

3.2.1 Client Needs

Axcelis' Trade Compliance department identified three core functionalities and attributes for this software system to have. First, this system needed to work for any given file type and work independently of different document formatting. Second, it needed to be fast and effective in extracting relevant data elements. Third, it needed to protect confidential information such as supplier contact information or part pricing disclosed on these invoices.

3.2.2 System Limitations

There are two main limitations to the current system's design: scalability and lack of contextualization. In terms of scalability, there is still a requirement for employees to collect invoices and pre-alerts from supplier emails. This project is throttled based on how these invoices are collected and, as it currently stands, is still partially manual.

Part numbers can greatly vary in how they are presented. Without context, phone numbers and financial information can very easily be misidentified as part numbers. As the system is designed and implemented currently, there is no utilization of context to inform classification, resulting in overidentification of part numbers for most invoices. Heuristic measures have been taken to reduce this but are not as effective in removing all misidentified parts.

3.3 Key Components

3.3.1 Invoice Processor

The invoice processor is a foundational aspect of this project. To efficiently build this system, I partitioned documents into two categories: text extractable and non-text extractable. This partitioning strategy is illustrated in Figure 2. Text extractable documents that use ASCII characters in their raw data formats such as Word Documents, Excel spreadsheets, or CSV files not need any additional processing methods to get text data. On the other hand, non-text extractable documents such as PDFs containing image scans required leveraging Google Tesseract's OCR-engine to programtically convert image data to text data.

3.3.2 Binary Classifier

One of the key components of this project was building a system to identify part numbers against other numbers and information contained on the invoices. This involved designing and testing the efficacy of a binary classifier. For this project, I explored three separate models: linear regression, random forest regression, and neural networks. To train the part identifier, I combined the database containing classified part numbers as well as the English dictionary. Within this database, there are over 23000+ classified parts and 50000+ words each of which with a binary classification of 1 (denoting a part) and 0 (denoting not a part). This might not be the most effective or comprehensive training data to utilize for the identifier; however, the English dictionary was a readily available and straightforward source of information to train against.

4. RESULTS

To test the binary classifiers, I tested using 23 invoices (formatted as PDFs) which had a total of 132 parts. After testing, all binary classification models were able to identify at least 123 of the parts contained in the invoices. However, there were numerous extraneous numbers and words that were misidentified such as phone numbers and invoice tracking numbers. In terms of performance, the neural network had the highest accuracy of 88.7%, thrashing the random forest regression and linear regression's accuracy of 76.1% and 72.0%, respectively. However, despite the neural network configuration performing best, it was also the slowest due to input transformation. I ultimately chose the random forest regression model due to its fast computation speed and reasonable accuracy.

While deploying this project at Axcelis, I was able to cut processing time from 1-2 hours to only 10-15 minutes. This drastic decrease in manual processing time could be improved even further by increasing the accuracy of the classifiers—reducing the need to double-check part numbers—as well as integrating procedure to collect invoices and pre-alerts automatically from Axcelis' email inbox.

5. CONCLUSION

With any company, there are always improvements in business processes that could be facilitated with software solutions. At Axcelis Technologies, I designed and developed a software system to reduce manual processing of invoices and pre-alerts. This solution leveraged machine learning techniques—specifically optical-character recognition and binary classification—to accurately extract part numbers from a variety of differently formatted invoices. I used skills and practical experience acquired from software engineering and machine learning courses at the University of Virginia to design and implement this software system.

During testing, the invoice processor has demonstrated high accuracy in identifying and extracting part numbers for classification. While deployed, this system can save upwards of 1-2 hours of manual processing time for Axcelis. Work still needs to be done in collecting invoices automatically from Axcelis' inbox as well as improving accuracy by including contextual part identification.

6. FUTURE WORK

Future work on the invoice processor includes reducing its dependence on the manual collection of invoices and pre-alerts. Axcelis receives invoices by email and employees must manually sift through the inbox to find pertinent documents, which consequently throttles the processor's automation efficiency. To resolve this, vendors could upload their documents to a web portal that could be connected directly to the invoice processor. Alternatively, a script could be written to automatically search through Axcelis' inbox and download pre-alerts for processing.

To reduce incorrect part identification, a redesign of the binary classifier might be necessary. Instead of assessing indivudal words, the binary classifier would need to consider the context surrounding each word to more accuracy assess whether it might be a part number or not. This redesign would likely include a natural language processing component.

7. ACKNOWLEDGMENTS

Thank you to Yola Karraz, Katie Wright, Diane Hofhaug, and Andrew Rose for their help during the development of this software. Their expertise on import and export control was crucial in identifying part number patterns and capturing training data.

REFERENCES

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Retrieved from https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/33418.pdf

Kazakos, E., Gkamas, A., & Bompotas, T. (2021). Autonomous vehicles: An overview of key technologies and challenges. Retrieved from https://kth.diva-portal.org/smash/get/diva2:1461111/FULLTEXT01.pdf

Rosebrock, A. (2021, October 27). Automatically OCR'ing receipts and scans with OpenCV and Tesseract. PyImageSearch. Retrieved from https://pyimagesearch.com/2021/10/27/automatically-ocring-receipts-and-scans/