EFFIENCY OF TESSERACT OPTICAL CHARACTER RECOGNITION

LIM SIN KUAN

Faculty of Computer Science and Information Computing Technology New Era University College

Kajang, Malaysia

Limsinkuan0415@e.newera.edu.my

***Abstract*— Optical Character Recognition (OCR) is the process of identifying and converting texts rendered in images using pixels to a more computer-friendly representation. Tesseract is a popular open-source OCR engine, developed initially by Hewlett Packard and later sponsored by Google. Despite its popularity, there are concerns about the efficiency of Tesseract OCR engine in comparison to other OCR engines. This report aims to discuss the efficiency of Tesseract OCR engine and its performance in comparison to other OCR engines, addressing the problem of accurately identifying and converting text in images using Tesseract. (*Abstract*)**

Keywords—Optical Character Recognition (OCR), Tesseract, efficiency, comparison

I. INTRODUCTION

Optical Character Recognition (OCR) is the process that converts an image of text into a machine-readable text format. For example, if you scan a form or a receipt, your computer saves the scan as an image file. You cannot use a text editor to edit, search, or count the words in the image file. However, you can use OCR to convert the image into a text document with its contents stored as text data. [1]

Most business workflows involve receiving information from print media. Paper forms, invoices, scanned legal documents, and printed contracts are all part of business processes. These large volumes of paperwork take a lot of time and space to store and manage. Though paperless document management is the way to go, scanning the document into an image creates challenges. The process requires manual intervention and can be tedious and slow. [1]

Moreover, digitizing this document content creates image files with the text hidden within it. Text in images cannot be processed by word processing software in the same way as text documents. OCR technology solves the problem by converting text images into text data that can be analyzed by other business software. You can then use the data to conduct analytics, streamline operations, automate processes, and improve productivity. [1]

Modern OCR systems use intelligent character recognition (ICR) technology to read the text in the same way humans do. They use advanced methods that train machines to behave like humans by using machine learning software. A machine learning system called a neural network analyzes the text over many levels, processing the image repeatedly. It looks for different image attributes, such as curves, lines, intersections, and loops, and combines the results of all these different levels of analysis to get the final result. Even though ICR typically processes the images one character at a time, the process is fast, with results obtained in seconds. [1]

Tesseract is an open-source Optical Character Recognition (OCR) Engine that operates under the Apache 2.0 license. The current stable version is Major version 5, which began with its release, 5.0.0, on November 30, 2021. Newer minor versions and bugfix updates are accessible via GitHub. The latest source code can be found on the main branch of the GitHub repository. Issues related to the software can be tracked through the issue tracker, along with planning documentation. [1]

Tesseract offers two main modes of use: through the command line directly or via an API, allowing programmers to extract text from images. This OCR Engine provides robust support for numerous languages. It's worth noting that Tesseract doesn't come with an integrated graphical user interface (GUI), but there are various options available from third-party sources. Additional tools, wrappers, and training projects related to Tesseract are listed under AddOns. [1]

Tesseract stands as free software, and contributions are welcome. If you encounter a bug and manage to fix it, the recommended approach is to attach patch to bug report in the Issues List. [1]

II. OVERVIEW OF TESSERACT OPTICAL CHARACTER RECOGNITION

A. What is Tesseract?

Tesseract is an open-source OCR engine that was developed at HP between 1984 and 1994. Like a supernova, it appeared from nowhere for the 1995 UNLV Annual Test of OCR Accuracy [2], shone brightly with its results, and then vanished back under the same cloak of secrecy under which it had been developed. Now for the first time, details of the architecture and algorithms can be revealed.

Tesseract began as a PhD research project in HP Labs, Bristol, and gained momentum as a possible software and/or hardware add-on for HP's line of flatbed scanners. Motivation was provided by the fact that the commercial OCR engines of the day were in their infancy, and failed miserably on anything but the best quality print. [2]

After a joint project between HP Labs Bristol, and HP's scanner division in Colorado, Tesseract had a significant lead in accuracy over the commercial engines, but did not become a product. The next stage of its development was back in HP Labs Bristol as an investigation of OCR for compression. Work concentrated more on improving rejection efficiency than on base-level accuracy. At the end of this project, at the end of 1994, development ceased entirely. The engine was sent to UNLV for the 1995 Annual Test of OCR Accuracy [3], where it proved its worth against the commercial engines of the time. In late 2005, HP

released Tesseract for open source. It is now available at http://code.google.com/p/tesseract-ocr.

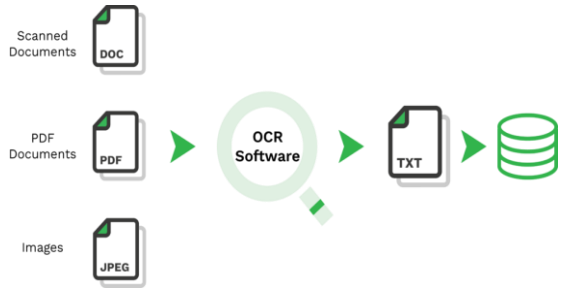

Figure 1

B. How Tesseract works

Tesseract converts the input image into binary format using thresholding. Outlines of components are stored on connected Component Analysis. Nesting of outlines is done which gathers the outlines together to form a Blob. Text lines are analyzed for fixed pitch and proportional text. Then the lines are broken into words by analysis according to the character spacing. Fixed pitch is chopped in character cells and proportional text is broken into words by definite spaces and fuzzy spaces. [4]

Overall Flow are shown as figure 2 and figure 3.

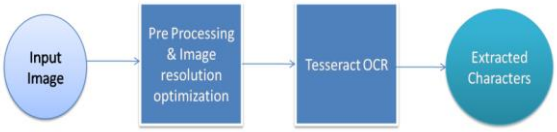

Figure 2

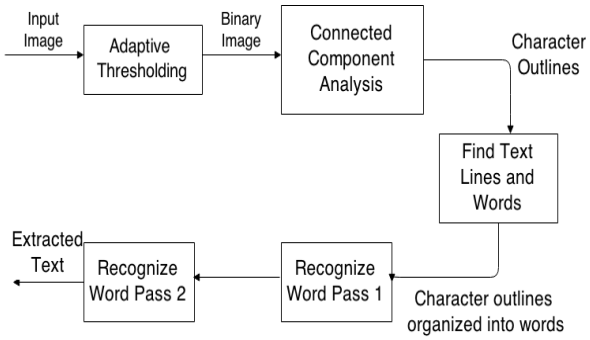

Figure 3

C. Advantages

OCR (Optical Character Recognition) technology has an excellent ability to present text information with high precision. When considering input methods, flatbed scanners stand out for their exceptional precision, producing commendable image quality. One of its unique strengths is the fast processing of OCR-converted data. This technology facilitates the rapid entry of large amounts of text for efficient information processing. An interesting application is converting paper forms to spreadsheets. This transformation not only simplifies storage, but also facilitates easy transmission via email or other digital channels. The most recent version of the software offers the capability to accurately reproduce tables while retaining their original layouts. This automated process proves significantly faster compared to the manual task of inputting information into the system. In its advanced iterations, the software goes a step further by not only re-creating tables and columns but also generating entire layouts, including webpage structures. [5]

D. Disadvantages

OCR text is highly effective when dealing with printed text, but it does not extend its efficiency to handwritten content. Handwritten text necessitates training for recognition by the computer. Additionally, OCR systems come with a significant cost. Furthermore, the generation of images consumes a considerable amount of space, and there's a potential for image quality to degrade during this process. The ultimate quality of the output image is heavily reliant on the quality of the initial image. As a result, thorough manual checking and correction of documents become essential. It's important to note that OCR is not entirely error-free, and some mistakes are likely to occur during the recognition process. For shorter amounts of text, the effort and resources invested might not justify the returns. [5]

III. Comparison between Tesseract and other OCR

A. Dataset and Metrics

In table 1 are reported some statistics of the OCR Evaluation Dataset that will be used for the comparison.

Training Set	
Number of images	7411
Mean height	19.033
Mean width	106.816
Mean text length	16.993
Test Set	
Number of images	2332
Mean height	20.119
Mean width	108.159
Mean text length	21.674

Table 1. Some Statistics about OCR Evaluation Dataset

The training (test) set contains only fields extracted from the documents contained the training (test) set of the FUNSD dataset. The performances of the OCR tools will be compared only with respect to the Test set. [6]

To measure the similarity between the extracted text and the ground truth text contained in the annotations, the following measures will be used:

$$accuracy(s_1, s_2) = 1 \text{ if } s_1 = s_2, \ 0 \text{ otherwise}$$

$$similarity(s_1, s_2) = 1 - LevenshteinDistance(s_1, s_2)/max\{|s_1|, |s_2|\}$$

Figure 4. Equations for comparison

The OCR tools will be compared with respect to the mean accuracy and the mean similarity computed on all the examples of the test set. I decided to also use the similarity measure to take into account some minor errors produced by the OCR tools and because the original

annotations of the FUNSD dataset contain some minor annotation errors, Figure 5. [6]

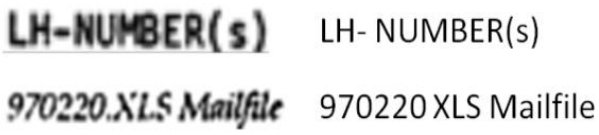

Figure 5

B. *Experiment setup*

1. OCR Tools

   I will use the Pytesseract python wrapper to interact with Tesseract. For the experiments I will use Tesseract 5.0.0 alpha for Windows, Azure OCR provided by Microsoft, Amazon Textract provided by Amazon, and Google OCR. [6]

2. Pre-processing

   The presented comparison will use the default parameters of the OCR tools, no additional tuning of them will be performed. Azure OCR expects a minimum resolution size of 50x50 for the input images. For this reason, all the images with a lower resolution will be resized to have a minimum side length of 50 pixels, the resizing will be done by padding the original image. Because our aim is to compare the OCR tools without tuning any particular parameter, I decided to perform this resizing transformation only for the Azure OCR tool because it is needed in order to use it.

3. Post-processing [6]

   The annotations contained in FUNSD don't consider new line characters, for this reason I replaced all the new line characters from the outputs of the tested OCR tools with a space character and then replaced all the occurrences of two or more consecutive space characters with only one space character from both the ground truth and the outputs of the OCR tools. [6]

C. *Cost*

Tesseract OCR	Free
Azure OCR	**0-1 Million** transactions/month **1.00** USD per **1000** transactions **1-10 Million** transactions/month **0.65** USD per **1000** transactions **10-100 Million** transactions/month **0.60** USD per **1000** transactions **> 100 Million** transactions/month **0.40** USD per **1000** transactions
Amazon Textract	**0-1 Million** transactions/month **1.50** USD per **1000** transactions **> 1 Million** transactions/month **0.60** USD per **1000** transactions
Google OCR	**0-5 Million** transactions/month **1.50** USD per **1000** transactions **> 5 Million** transactions/month **0.60** USD per **1000** transactions

Table 2

D. *Results*

In Table 2 are shown the results obtained by the different tools on the Test Set of the OCR Evaluation Dataset, in terms of accuracies and similarities.

	Accuracy	Similarity
Tesseract OCR	0.306	0.569
Azure OCR	0.633	0.915
Amazon Textract	0.345	0.513
Google OCR	0.576	0.892

Table 3. Results on the test set

Azure OCR and Google OCR show the best performances in both metrics, Tesseract OCR and Amazon Textract are the worst.

Figure 6

Looking at the Scatter Plots of the different combinations of the OCR results, Figure 5, it is possible to see that there is not a clear correlation between the obtained results, except for the pair: Azure OCR and Google OCR. In particular, although Tesseract OCR and AWS Textract perform similarly overall their results are not strongly correlated. [6]

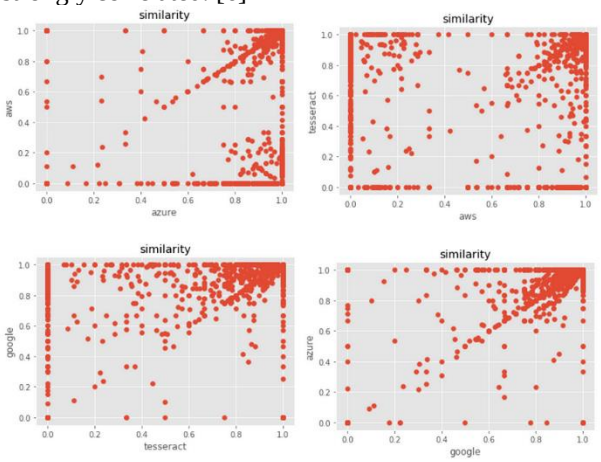

Figure 7. Scatter Plots between the similarities of the results obtained by the OCR tools

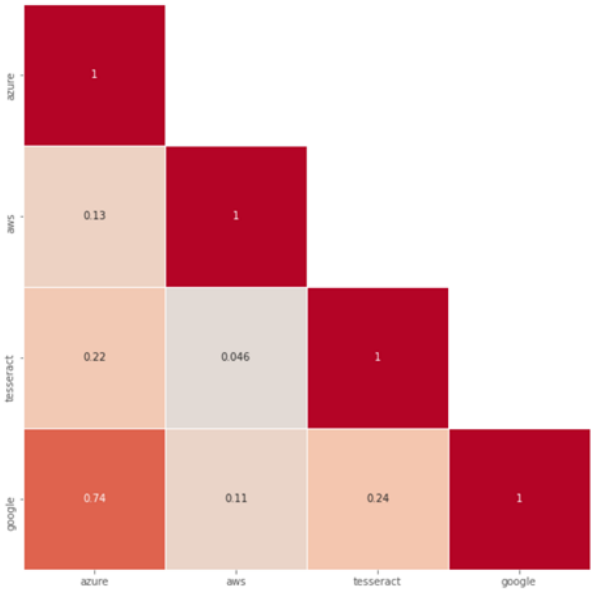

Figure 8. Correlation Matrix between the similarities of the results obtained by the OCR tools

The differences between Azure OCR and Google OCR and the other tools could be motivated by a lot of factors: maybe Azure OCR and Google OCR were trained to address images similar to the ones contained in the OCR Evaluation Dataset or maybe not, but at least the important differences between these two OCR tools and the other ones show that the decision of which OCR tool to use in a project should also be driven by an initial evaluation of the expected performances of the tools. [6]

IV. Future Work and Scope

Enhancing Accuracy through Advanced Machine Learning: One promising avenue for improving Tesseract OCR is by implementing advanced machine learning techniques, such as deep learning models, to enhance its accuracy. Training the system on larger and more diverse datasets could lead to improved recognition of various fonts, styles, and languages.

Contextual Understanding for Complex Layouts: Tesseract could benefit from incorporating context-aware algorithms that better understand and interpret complex document layouts, such as multi-column formats or tables. By analyzing relationships between different text elements, the OCR engine could reconstruct intricate structures more accurately.

Improved Handwriting Recognition: Expanding Tesseract's capabilities to handle handwritten text by integrating handwriting recognition models could open up new avenues for applications. This would involve extensive training to enable the system to decipher a wide range of handwriting styles.

Optimized Preprocessing Algorithms: Enhancements in preprocessing techniques, such as image enhancement, noise reduction, and contrast adjustment, could significantly improve OCR accuracy by providing cleaner input to the recognition engine. Developing adaptive preprocessing strategies based on input image quality can aid in capturing fine details.

Multi-Language Support Refinement: Tesseract already supports a variety of languages, but further work could focus on fine-tuning recognition for specific dialects, regional variations, and less commonly spoken languages. This would broaden its usability for a more diverse user base.

Efficiency in Hardware Utilization: Leveraging hardware acceleration, such as GPUs or TPUs, could speed up the OCR process, especially for large-scale applications or real-time recognition scenarios. Optimization of resource utilization would contribute to faster results.

Integration of Natural Language Processing (NLP): Incorporating NLP techniques could enable Tesseract to not only recognize text but also extract meaning and context from the recognized content. This would be particularly useful in scenarios where understanding the semantic context is vital.

Cloud-Based Collaboration and Scalability: Developing cloud-based OCR services powered by Tesseract would enable collaborative document processing, making it easier for multiple users to access and contribute to OCR tasks. This approach could also ensure scalability for handling varying workloads.

User-Friendly Interface and Automation: Creating a more intuitive and user-friendly interface, along with automation features, would make Tesseract accessible to a wider range of users, even those without programming expertise. This could simplify the integration of OCR into various applications.

Feedback Loop for Continuous Improvement: Implementing mechanisms for users to provide feedback on recognition errors and challenges faced during usage can contribute to the ongoing refinement of Tesseract. Regular updates and improvements based on user feedback could ensure the OCR engine evolves to meet changing demands.

In conclusion, Tesseract OCR holds tremendous potential for enhancement and expansion in various directions. By combining cutting-edge technologies, refining existing processes, and addressing user needs, the efficiency and accuracy of Tesseract OCR can be substantially improved, opening up new possibilities for document digitization, data extraction, and beyond.

V. Conclusion

In conclusion, Optical Character Recognition (OCR) serves as a critical process, bridging the gap between visual information and digital content by converting pixel-based textual images into machine-readable formats. Tesseract, an open-source OCR engine with its origins traced back to Hewlett Packard and subsequently supported by Google, has gained widespread recognition. However, questions have arisen regarding its efficiency when juxtaposed with other OCR engines.

This report delved into the efficiency of the Tesseract OCR engine and its relative performance when measured against alternative OCR solutions. The core challenge tackled was the accurate extraction and conversion of text from images through the Tesseract engine. Through a comprehensive exploration of its capabilities, limitations, and comparative analysis, we have shed light on the nuanced landscape of OCR technologies.

As the digital transformation continues to shape diverse domains, the accurate and swift transformation of visual content to machine-interpretable data remains pivotal. While Tesseract has showcased its competence, it's vital to acknowledge that OCR is a multifaceted endeavor, influenced by factors such as recognition accuracy, processing speed, adaptability to diverse layouts, and language support. Recognizing Tesseract's strengths and areas for potential enhancement paves the way for a nuanced understanding of its role within the broader OCR ecosystem.

In this context, further research, development, and refinement could propel Tesseract's efficiency to new heights. By addressing concerns and leveraging the growing pool of OCR advancements, Tesseract could continue to evolve, potentially closing the efficiency gap and solidifying its place as a formidable tool in the realm of Optical Character Recognition.

VI. References

[1] R. Smith, "The Extraction and Recognition of Text from," 1987.

[2] A. S, "An overview of Tesseract OCR Engine," *An overview of Tesseract OCR Engine*, 2016.

[3] S. Rice, "The fourth annual test of OCR accuracy," 1995.

[4] N. Kashyap, "ResearchGate," [Online]. Available: https://www.researchgate.net/figure/Architecture-of-Tesseract-Tesseract-converts-the-input-image-into-binary-format-using_fig1_276108387.

[5] pulkitagarwal03pulkit, "Advantages and Disadvantages of Optical character Reader (OCR)," *Advantages and Disadvantages of Optical character Reader (OCR)*, 2022.

[6] F. Ricciuti, "how-to-compare-ocr-tools-tesseract-ocr-vs-amazon-textract-vs-azure-ocr-vs-google-ocr," *how-to-compare-ocr-tools-tesseract-ocr-vs-amazon-textract-vs-azure-ocr-vs-google-ocr*.

[7] N. Otsu, " A threshold selection method from gray-level histograms," *Automatica*, , 1975.