Optical Character Recognition as a Cloud Service in Azure Architecture

Onyejegbu L. N. Department of Computer Science, University of Port Harcourt. Port Harcourt. Rivers State. Nigeria.

Ikechukwu O. A. Department of Computer Science, University of Port Harcourt. Port Harcourt. Rivers State. Nigeria

ABSTRACT

Cloud computing and Optical Character Recognition (OCR) technology has become an extremely attractive area of research over the last few years. Sometimes it is difficult to retrieve text from the image because of different size, style, orientation and complex background of image. There is need to convert paper books and documents into text. OCR is still imperfect as it occasionally mis-recognizes letters and falsely identifies scanned text, leading to misspellings and linguistics errors in the OCR output text. A cloud based Optical Character Recognition Technology was used. This was powered on Microsoft Windows Azure in form of a Web Application Programming Interface that load images to an Optical Character Recognition server, process with necessary recognition, export parameters, and obtain the results of the processing. The key idea is to bring together the advantages of the Optical Character Recognition technology and cloud computing in one place in other to enable quicker access and faster turn out time, processing period, and increasing efficiency across the board for application users. The methodology adopted is the object oriented methodology. This was achieved using JAVA programming language.

General Terms

Cloud Computing: Is the delivery of on-demand computing resources—everything from applications to data centers—over the Internet on a pay-for-use basis.

Optical Character Recognition (OCR): Is a technology that enables you to convert different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data.

Microsoft Azure: cloud-based application platform for developing, managing, and hosting applications off-site.

ABBYY: It a company founded in 1989 as BIT Software, and renamed in 1997. It creates artificial intelligence technologies products and services to extract information from sources in which it would be otherwise digitally inaccessible. Amongst its products and services are dictionary tools, translation and business card reading. SDK (Software development kit), a programming package that enables a programmer to develop applications for a specific platform.

Keywords

Cloud computing, optical character recognition (OCR), Microsoft Window Azure, image, text, ABBYY.

1. INTRODUCTION

There is a paradim shift in the manner people learn, trade, communicate and share knoweldge with the drastic introdrodcution of modern computers into every aspect of life The dawn of digital computers has made it unavoidable that everything processed by the digital computer must be processed in digital form. For example, most famous libraries in the world like the Boston public library has over 6 million books, for public consumption, inescapably has to change all its paper-based books into digital documents in order that they could be stored on a storage drive. It can also be projected that every year over 200 million books are being published[22], many of which are disseminated and published on papers [6].

Thus, for several document-input jobs, Opitcal Character Reconigtion is the utmost economical and swift process obtainable. Every year, the innovation liberates storage space once meant for file organizers and boxes loaded with paper records. Before OCR can be utilized, the source material must be scanned with the use of an optical scanner to read the page as bitmaps.(patterns of dots).. The Optical Character Recogniton software then processes these scans to differentiate between images and text and determine what letters are represented in the light and dark areas.[7]. Older OCR systems match these images against stored bitmaps based on specific fonts. The hit-or-miss results of such pattern-recognition systems helped establish OCR's reputation for inaccuracy[7].

Moreover, the concept of cloud computing is quickly gaining popularity amongst the business world due to its ease of use and the flexibility it provides to businesses. Cloud computing is a term used to denote an abstract structure that offers services like storage, soft wares, and computing to all the computers that are connected to it over the network or the internet. Enabling quicker access and faster turn out time, cloud computing is reducing the processing period and increasing efficiency across the board for organizations.

Consequently, our goal is to bring together all the advantages of the Optical Character Recognition technology and cloud computing in one place in other to enable quicker access and faster turn out time, processing period, and increasing efficiency across the board for application users. Sometimes it is difficult to retrieve text from the image because of different size, style, orientation, complex background of image etc. There is need to convert paper books and documents into text. Optical Character Recognition is still unsatisfactory as it intermittently mis-recognizes letters and distinguishes scanned text falsely leading to wrong spellings and grammar errors in the Optical Character Recognition output text. Thus there is need for a cloud based Optical Character Recognition technology that would enable quicker access and faster turn out time, processing period, and increasing efficiency in the retrieval of text from images for various purposes. Poor and insignificant results have been observed from different Optical Character Recogniton systems when the input source is substantially out of condition, of the past, with low print quality, and having imperfections and distortions such as stains, rips, blots, and marks. The study brings together all the advantages of the OCR technology and cloud computing in one place for application users. Through a cloud based Optical Character Recognition system users can scan physical documents, convert the images into editable PDF formats, and save the resulting file on the storage. In addition, the speed and accuracy assigned by the OCR cloud computing builds employee morale, as it encourages employees to work more efficiently. This paper concentrated on developing a cloud based Optical Character Recognition System using ABBYY Cloud Standard Development Kit which leverages Microsoft Windows Azure cloud-based deployment and service-based interfaces to extract, process, synthesize text image in a flexibly secure and scalable architecture.

2. RELATED WORK ON OPTICAL CHARACTER RECOGNITION AND CLOUD COMPUTING

[24] proposed a post-processing context-based error correction algorithm for detecting and correcting OCR non-word and real-word errors. The algorithm used is based on Google's online spelling suggestion which harnesses an internal database containing a huge collection of terms and word sequences gathered all over the web, convenient to suggest possible replacements for words that have been mispelled during the OCR process. Future improvement on the algorithm suggested a support over multiprocessing platforms so as to operate in a parallel fashion over a bunch of concurrent processors or even over a bunch of distributed computing machines. This is one of the crux of this present work .

[12] proposed an Optical Character recognition application for android mobiles that performs image to speech conversion, this technology was mainly designed for people who are unable to read any type of text documents. Open source software Tesseract was used as the basis for the project. Future recommendations was that OCRs features should be updated so that language translation will be included to help people from different countries who can't understand a particular language. This present research is also implementing such features. The works of [8] has described that the extraction of text lines from document images is one of the important steps in the process of an Optical Character Recognition (OCR) system. In case of handwritten document images, presence of skewed, touching or overlapping text line(s) makes this process a real challenge to the researcher. The technique used extracts 87.09% and 89.35% text lines successfully from the said databases respectively.

[19], proposed a cloud-based framework which comprises of a set of virtual machines and a job management system. Result shows that the framework improves performance, scalability, upgradeability and flexibility

[25] proposed a distributed parallel genetic algorithm (DPGA) of placement strategy for virtual machines deployment on cloud platform.Experimental results show that the proposed placement strategy of VM deployment canensure QoS for users and it is more effective and more energy efficient than other placement strategies on the cloud platform

2.1 Review of Optical Character Recognition Technologies- Tesseract

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0, and development has been sponsored by Google since 2006. The Tesseract engine was originally developed as proprietary software at Hewlett Packard labs in Bristol, England and Greeley, Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some migration from C to C++ in 1998. A lot of the code was written in C, and then some more was written in C++. Since then all the code has been converted to at least compile with a C++ compiler.

GOCR: GOCR claims it can handle single-column sans-serif fonts of 20–60 pixels in height. It reports trouble with serif fonts, overlapping characters, handwritten text, heterogeneous fonts, noisy images, large angles of skew, and text in anything other than a Latin alphabet. GOCR can also translate barcodes

OCRopus; OCRopus is a free document analysis and optical character recognition (OCR) system released under the Apache License, Version 2.0 with a very modular design through the use of plugins. These plugins allow OCRopus to swap out components easily. OCRopus is currently developed under the lead of Thomas Breuel from the German Research Centre for Artificial Intelligence in Kaiserslautern, Germany and is sponsored by Google. OCRopus is developed for Linux; however, users have reported success with OCRopus on Mac OS X and an application called TakOCR has been developed that installs OCRopus on Mac OS X and provides a simple droplet interface.

JAVAocr:

This is yet another OCR engine targeting mobile devices. Java OCR is a suite of pure java libraries for image processing and character recognition. Small memory footprint and lack of external dependencies makes it suitable for android development, provides modular structure for easier deployment. It has a modular structure for easier development and deployment on system. It is built to run on cross platform applications. There are some disadvantages in this OCR engine. First, it's not fast in processing. Some of the portions work really well but others are lousy. Second, it's not very well documented and supported.

Our research work optical character recognition as a cloud service in azure architecture attempts to study and develop a more efficient Optical Character Recognition as a Softwarea-Service in Microsoft Windows Azure as-Architecture.Design an efficient Optical Character Recognition System. Using ABBYY Cloud Standard Development Kit leveraging on Microsoft Windows Azure in a Cloud based Optical Character Recognition. This is implemented usingJava programming Language via Software-as-a-service (SaaS) cloud service model. In Optical Character Recognition System

3. THE EXISTING SYSTEM

Over the years Optical Character Recognition technology has improved speedily. However there are limitations with respect to the source materials and character formatting. Most documents formatting are lost during text scanning, except for paragraph marks and tab stops. Sometimes, the output from a finished text scan will be a single column editable computer file. In most cases this computer file will at all times require proofreading and spellchecking and in addition to reformatting to the required final layout.

Using one of the open source and free OCR software for instance, Tesseract OCR is an elegant engine with various layers. It works in step by step manner as shown in the block diagram in figure 1. The first step in the cycle is to sense the colour intensities of the image, named as adaptive thresholding, and converts the image into binary images. Second step is to do the connected component analysis of the image, which does the task of extracting character outlines [12].

After this the outlines extracted from image are converted into Blobs (Binary Long Objects). It is then organized as lines and regions and further analysis is for some fixed area. After extraction the extracted components are chopped into words and delimited with spaces. Recognition in text then starts which is a two pass process.

Tesseract was originally designed to recognize English text only and there are limits as to the range of languages it can handle. It can only handle left-to-right languages. While you can get something out with a right-to-left language, the output file will be ordered as if the text were left-to-right. Top-tobottom languages will currently be hopeless. It is unlikely to be able to handle connected scripts like Arabic. It will take some specialized algorithms to handle this case, and right now it doesn't have them. Moreover, it is likely to be so slow with large character set languages (like Chinese) that it is probably not going to be useful. There also still need to be some code changes to accommodate languages with more than 256 characters .

Input Image



Fig 1: System Architecture of the Existing SystemTesseract OCR Architecture (Patel, Patel, & Patel, 2012).

In Figure 1, the first part is when attempt to recognize each word is made. Each satisfactory word is accepted and second pass is started to gather remaining words. This brings in the role of adaptive classifier. The adaptive classifier then will classify text in more accurate manner. The adaptive classifier needs to be trained beforehand to work accurately. When the classifier receives some data, it has to resolve the issues and assign the proper place of the text [12].

3.1. The Proposed System

In this paper the proposed system is written in Java. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that integrates with ABBYY CLOUD Optical Character Recognition Standard Development Kit via a web Application Programming Interface. The Web API enablescommunication with remote server via Hyper Text Transfer Protocol. Through this Web API images are loaded to the OCR server, there the image is processed with the necessary recognition and export parameters, after which the results of processing are obtained and stored in a file.

3.2 The Design of the Proposed System

The design model is composed of a client computer application that connects to Microsoft Azure Platform and Data Center. The client application (TONY_CLOUD_OCR) implements ABBYY Cloud Standard Development Kit which runs on Microsoft Windows Azure.



Fig .2: Architecture of the Proposed System

The proposed system as shown in figure 2 above is an interaction of the various components that ensure that texts are efficiently extracted form images. Firstly, files are loaded from disk or memory and Multi-page files are split into single pages to increase speed and scalability on multiple cores machines. The Document/Layout analysis component analyses the text block and detects the lines and identifies various individual characters. The OCR voting component applies optical characters. Afterwards texts are rebuilt on word level by using language information and morphology dictionaries. The recognized text is then exported in proper

Unicode format and reconstructed and synthesized to suit different output formats. The Optical Character Recognition output is then downloaded for the OCR cloud server to be saved in the disk.

3.2. Algorithm of the Proposed System

The proposed system preform optical character recognition in steps using ABBYY cloud SDK. The steps form image to text is as follows:

Opening images and PDFs

- Files are loaded from disk or memory.
- A large variety of images formats are supported
- Preparing the images
 - Multi-page files are split into single pages to increase speed and scalability on multiple cores machines
 - Rotate images so the text in the images can be read
 - Clean the images(removal and scanning dust from digital cameras)
- Analyze the layout, detected text, images, and barcodes tables' areas.
 - Detect the reading order of the texts
 - Analyze the text blocks and detect the lines and find/identify the individual characters"
- Read the "individual" characters. and apply optical character recognition
 - Vote different hypothesis of single characters, e.g. is it
 - "0" or "O" or "o" or a "Q" or "Ö"
 - "I" or "1" or "!" or "!"
- **Rebuild the text on a word level** by using language information
 - What characters are used and allowed in the language
 - What recognition settings are set-up internally
 - Are defined word lists available or can some details be looked up in a database
 - Use linguistic and morphology dictionaries
- **Export** the recognized:
- text in the proper "Unicode"
 - Provide all the details "found" in XML, e.g.
 - Character Positions (original and after de-skewing of the page)
 - Fonts, Formats (normal/bold), color
 - Hypothesis
 - Word in dictionaries
- **Reconstruct and synthesize the original layout** for different output formats:
 - Office formats
 - HTML
 - e-book
 - PDFs with an image and text layer, text only, PDF/A
 - Reconstruct the logical structure of a document (headers, footers, etc.)
 - Export/Save the different formats to RAM or disk so that they comply with the format standards."

3.3 Methodology

In this paper, the proposed system uses the object oriented approaches to implement the system requirements. It elaborates the analysis models to produce the implementation specifications of the system. In the design of the proposed system we use the following: object oriented Methodology: case diagram, the Unified Modelling Language (UML), class diagram, relation diagram, the activity diagram and the architectural design.

4. IMPLEMENTATION AND RESULT DISCUSSIONS

Implementation and testing of the system was carried out using theJava Runtime Environment (jre 7), Java Development Kit (jdk 1.6 or higher), Netbeans Integrated Development Environment 8.0 or higher.



Fig 3:the default graphical user interface of TONY_CLOUD_OCR.

Figure 3 is The user interface form where users are permitted to select a file to recognize, input output name, indicate recognition language, choose conversion option and the submit.



Fig 4.Graphical User Interface Open Button navigation window.

Figure 4 this allows the user to navigate to a desired file location in order to specify the correct file path.



Fig 5: the Graphical User Interface form with selected file path, output name and conversion options.

In figure 5 the output name text field allows the user to specify the output name to be used for the Optical Character Recognition output file name.in Conversion Options The *JComboBox selectRecognition Option* provides a variety of Recognition options such as "RECOGNIZE", "BUSCARD", "TEXTFIELD", "BARCODE", "PROCESSFIELDS",

"MRZ".The RECOGNIZE option is the default mode for general Optical Character Recogniton. BUSCARD mode allows for efficient recogniton of Business Cards, TEXTFIELD mode facilities recognitions in textfileds.

TONY_CLOUD_OCR		-		Х
File Path		jo	pen File	
Output Name]		
Language	ENGLISH			
Conversion Options	RECOGNIZE			
	submit			
	uploading			
	downloading			
	done			

Fig. 6 Graphical User Interface after Optical Character Recognition.

4.1 Discussion of Results

This application is adept of Using ABBYY cloud Optical Character Recognition Standard Development Kit service which leverages Microsoft Windows Azure to host OCR processing functionality. TONY_CLOUD_OCR was developed and integrated via a web Application Programming Interface. Result shows an application running without significant up-front infrastructure or maintenance cost. We outsource computing power for Optical character recognition to develop a scalable platform, independent application. Leveraging ABBYY Cloud OCR SDK this application has an exceptional language support and professional Recognition capabilities, which recognizes 198 languages in printed text, including all major European languages, Arabic, Chinese, Japanese, and Korean; intelligent character recognition (ICR), plus barcode, checkmark (OMR) and field-level/zonal recognition. Experimental test show effective transformation of scanned documents and images into searchable and editable text files. In addition, predefined profiles facilitate recognition tasks that are performed most frequently. Processing parameters included enable both easy startup and optimal OCR quality - with no need for manual system tuning. The application recognizes and retains document elements with remarkable accuracy, enabling precise reconstruction of structures and layouts. Furthermore, results shows an improved conversion and higher accuracy, providing optimum recognition accuracy.

> Frühere Schriftträger [Bearbeiten] Höhlenzeichnungen sind die ältesten Dokumente, die der Mensch mit Pigmenflarbe auf einen Untergrund gezeichnet hat. Die Sumerer, die Träger der ältesten bekannten Hochkultur, schrieben seit etwa 3200 v. Chr. mit Kellschrift auf weicher Tontafdin, die zum Teil bis heute gebrannt wurden. Auch aus Agrupten sind Schriftträger aus anorganischen Materialien bekannt, beispielsweise die Prunkpalette des Königs Nammer (3100 v. Chr.) aus Speckstein.



war ein Pinsel aus <u>Binsen</u>, Zwar gab es Papyrus im <u>antiken Griechenland</u>, jedoch war eine Verbreitung über Griechenland hinaus kaum bekannt. Im 3. Jahrhundert rszekzten die Griechen den Pinsel durch ein gespaltene <u>Rohrfeder</u>. Von dem griechinschen Wort *popyros* leitet sich das Wort Papier ab.

Im <u>Römischen Reich</u> wurde sowohl Papyrus benutzt als auch Wachstafein. In die Letzteren wurde der Text mitlet angespitzet Griffel gericht. Nach dem Schuber geglitzt und die Tatel komte erneut beschrieben werden. Öffentliche Vervalunzgespekulen angebracht. In <u>Inden</u> wurden Blätter von Palmengewichsen und in Ching Tafeln aus Knochen. Musseheln. Elfenbein benutzt. Später bestanden in Ching Tafeln aus Branze. Einz dan von Banbusstreffen und Scied. Palmaceblätter und Tierhäute wurden noch nicht als Schriftitteln aus Glätter, das Branze. Einz dan auch aus organischem Material, wie Hölz, Schriftitteln aus die Scied. Palmaceblätter und Tierhäute wurden noch nicht als Schriftitteln ernetzt. Orkeitenschen wurde ann Vermitt. Tierhäuse werden Tierhäuten benutzt. Die Fleibaus werden Tierhäuten benutzt. Die Verbause benutzt. Die Verbause benutzt.

Fig 7 image of a text image in German language for optical character recognition.



Fig. 8 output of a recognized scanned text image in German. The style and structure of the image is retained.



Fig 9 OCR input (Vehicle Plate Number Registration) and OCR output



Fig 10: a screen shot image of the University of Port Harcourt Website (www.uniport.edu.ng) to be subjected to Optical Character Recognition.



Fig 11an editable word document of the screenshot image University of Port Harcourt website subjected to Optical Character Recognition.

5. CONCLUSION

A scalable Optical Recognition tool for optical character recognition, barcode recognition and intelligent character recognition is designed and developed. The proposed system called TONY_CLOUD_OCR accepts image inputs and recognizes text as pixels in images via ABBYY cloud based OCR technology. It has a very rich and user friendly GUI developed in Java Swing Technology. Thus it is totally easy to use. With Java, the most considerable advantage is platform independence; therefore TONY_CLOUD_OCR is also platform independent. The installation file is only 63KB, so it is highly economical in terms of memory use and because it is based on object-oriented design, any further changes can be easily adaptable.

6. ACKNOWLEDGMENTS

Our thanks to God Almighty who made this research work a success and Microsoft Windows Azure

7. REFERENCES

- Baker, E., and Joint, A. (2011), Knowing the past to understand the present issues in the contracting for cloud based services, Computer Law and Security, Review 27, 407-415.
- [2] Bo. L, Solamayor B., R. Madduri, K. Chard, and I. Foster, "Deploying Bioinformatics workflow on cloud with Galaxy and Globus provision". In High performance computing Networking Storage and Analysis (SCC), 2012 SC Companion 2012, 1089-1095
- [3] Claudiucires, D., Ueli, M., Gambardella, M., and Schrudhuber, J. (2011), Convolutional Neural Network Committees for Handwritten Character Classification, Proceedings of the IEEE International Conference on Document Analysis and Recognition.
- [4] Hassen, H.; Khemakhem, M. (2014), "A secured distributed OCR system in a pervasive environment with authentication as a service in the Cloud," in Multimedia Computing and Systems (ICMCS), 2014 International Conference on, vol., no., 1200-1205, 14-16
- [5] Ittner, D.I, Lewis, D.Y, and Ahn, D.D (1995), Text categorization of low quality images, Proceedings of SDAIR-95 4th Annual Symposium on Document Analysis and Information Retrieval
- [6] Klein, D., Kamvar, S.D., and Manning, C.D. (2002) from the instance-level constraints to space-level constraints; making the most of prior knowledge in data clustering. In proceedings of Nineteenth international conference on machine learning ICML '02. San Francisco, CA, USA 307-314
- [7] Lais, S. (2002). Application Development. Retrieved August 26, 2015, from COMPUTERWORLD: http://www.computerworld.com/article/2577868/appdevelopment/optical-character-recognition.html
- [8] Malakar Samir (2012)."Text line extraction from handwritten document pages using spiral run length smearing algorithm". IEEE international conference on communications, Devices and intelligent system (CODIS)
- [9] Mohammed, C., Nawwaf, K., Cheng-Lin, and L., Ching, Y.S (2007), Character Recognition Systems, *A Guide for Students and Practictioners*, Wiley-Interscience Publications. 255-276.

- [10] Mutholib, A., Teddy, S.G, and Mira, K. (2012), Design and Implementation of automatic number plate recognition on android platform, Proceedings of the IEEE International Conference of Computer and Communication Engineering,
- [11] Oak, H. (2014, November). Java EE Applications on Oracle's Java Cloud Service. An Oracle Brief. 500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A: Oracle Press.
- [12] Petal, Petal, And Petal (2012) "Optical character Recognition, by OpenSource OCR Tool Tesseract: A case study. International journal of Computer Applications (0975-8887) volume 55 no.10, October.
- [13] Prodan, R. and Ostermann, S., A survey and taxonomy of Infrastructure as a Service and Web Hosting Cloud Providers, Proceedings from the 10th IEEE/ACM International Conference on Grid Computing,
- [14] Ravina, M., Supriya, I., and Nilam, D. (2013), Optical Character Recognition, Proceedings from the International Journal of Recent technology and Engineering, Vol. 2, Issue 1, 72-75.
- [15] Roger, T.H, and Kathleen, C. (1999), Quality of OCR for Degraded Text Images, Proceedings of the 4th ACM conference on Digital Libraries, New York, U.S.A.
- [16] Sgdev-blog's blog (2014, May 20). How to build Java based Cloud Application. Retrieved August 26, 2015, from Java.net:http://www.java.net/blog/sgdevblog/archive/201 4/05/20/how-build-java-based-cloud-application
- [17] Sukhpreet, S. (2013), Optical character recognition techniques: A survey, Journal of emerging trends in computing and information sciences, Vol. 4, Issue 6, 545-550.

- [18] Singh, S. (2013). Optical Character Recognition Techniques: A Survey. Journal of Emerging Trends in Computing and Information Sciences, 4(6).
- [19] Trojahn, M.; Lei Pan; Schmidt, F. (2013). "Developing a Cloud Computing Based Approach for Forensic Analysis Using OCR," in IT Security Incident Management and IT Forensics (IMF), 2013 Seventh International Conference on, vol., no., 59-68, 12-1
- [20] Tsai, W., Sun X., Balasooriya, (2012) Service-Oriented Cloud Computing Architecture, Proceedings from the 7th IEEE International Conference on Information Technology, 684-689.
- [21] Vaquero, L.M, Merino, L.R, Caceres, J., and Linder M. (2009). A break in the clouds: Towards a cloud definition, Proceedings from the ACM SIGCOMM Computer Communication Review, Vol. 39, No.1, 50-55.
- [22] Vincent L. 2007 "Google Book Search; Document Understanding on a Massive Scale". Proceeding of the ninth international conference on Document Analysis and Recognition (ICDAR), 819-823.
- [23] Wang, L., Laszewski G., Kunze M., Tao, J. (2010). Cloud Computing: A perspective study, J New Generation Computing, 1-11.
- [24] Youssef, A.E. (2012). Exploring Cloud Computing Services and Applications, Proceedings from the Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, Issue 6, 838-847.
- [25] Yu-Shuang Dong, Gao-Chao Xu, and Xiao-Dong Fu (2014)" A Distributed parallel Genetic Algorithm of placement strategy for virtual Machines Deployment on Cloud platform". Scientific world journal volume Article ID 259139, http://dx.doi.org/10.1155/2014/259139