OCR-Enhanced Digital Asset Management System: Prototype Design and Construction

Janny S. Surmieda
Assistant Professor
University of the Philippines School of Library and Information Studies

Abstract

Digitization initiatives have become increasingly prominent for libraries and information centers. These initiatives have, in turn, also led to an increase in the use of digital management systems to store digitization outputs. Since these are large volumes of documents, potential alternatives to providing access to these materials aside from manual data entry processes such as indexing and cataloging are explored. In this area, the potential of Optical Character Recognition (OCR) mechanisms is gaining prominence as the tool that can process a large number of documents and provide automated text recognition at high speed. Leveraging the potential of OCR, this paper describes the design of a digital asset management system and the application of Tesseract, an open-source OCR engine, to create a functional and adaptive system to manage digital assets and retrieve information from these contents.

Keywords: Optical Character Recognition, Document Retrieval, System Analysis and Design, Digital Asset Management Systems

INTRODUCTION

A fundamental purpose of the information profession is to provide and give access to quality information. Significantly for information centers such as libraries, archives, or record centers, information is contained in formats primarily in print. Fortunately, with decades of technological advancements, ways to facilitate information retrieval from these materials have extensively progressed. Digitization initiatives have become a growing effort today to preserve physical materials and to make information available and accessible. Parallel to this is the effort to adopt mechanisms that enhance the chance for access and retrieval of information from these sources, hence the use of digital asset management systems or information retrieval systems. These systems use indexing methodologies to categorize and lift information from sources, and human intervention may be necessary for this process. But opportunities to aid in information retrieval have grown as well, particularly with machine use and systems that enable fast retrieval of information that are raw and plain keywords aside from controlled and limited descriptors has become pivotal. Among these is optical character recognition (OCR).

To further describe the process of using an OCR, this paper presents its brief history and its applications. It proceeds with the rationale for developing a prototype system that explores the use of OCR and its design, features, and functionality. The result of the prototype testing and the potential future of OCR for libraries will also be discussed.

DEVELOPMENT OF OCR TECHNOLOGY

The ability to retrieve data extensively from full-text sources is a crucial process that the OCR mechanism can enable. OCR is a technology that enables computers to recognize text or characters from an image automatically (Salimah et al., 2021) and a process that converts letters on a digitally scanned printed document to digital characters (Reitz, 2013).

OCR technology has been consistently improved up to the point that it can produce exploitable results on mainstream documents (Chiron et al., 2017).

Earlier development of the OCR traces back to the early 19th Century, as detailed by Raj and Kos in their 2022 paper. Notably cited was Edmund Edward Fournier d'Albe's machine, which he called "exploring optophone" (Raj & Kos, 2022; Thomas, 2021). This device converts light to sound and enables blind individuals to hear it. According to Thomas, in September 1913, d'Albe demonstrated a prototype of the reading optophone that later on would not be used commercially but influential in the further development of OCR. Furthermore, at the height of the First World War, E. Goldberg invented a machine that could read a text and convert it into telegraph code, which he further developed to become the first electronic document retrieval system (Raj & Kos, 2022).

In the early '90s, OCR gained prominence and has consistently increased in usage in different ways. For instance, OCR was used to recognize handwritten text in the study of Gillies et al. in 1995. Their study recognized handwritten text from census forms, which aimed to aid the encoding of census data to a system that will convert it to a computer-readable form. The handwritten text came from the 1990 census, totaling 9,000, and yielded a field error rate of below 4%. Furthermore, OCR was also used to recognize language-specific text in the study of Robby et al. (2019), where they collected a dataset for Javanese characters from digital and handwritten sources with a total of 5880 characters. Image enhancing methods, such as "rotation, filling the missing of Javanese characters, noise removal and clarify the stroke of handwritten Javanese characters" (Robby et al., 2019, p. 501) were performed for handwritten characters that exhibit blurriness, noise, misalignment, and missing of font component. Neural-Network API from Tesseract OCR was used to train the dataset they collected. Upon completing their study, they infer that the characters "i" and "e" of Javanese characters are hard to distinguish from one another, thus causing lower accuracy on the result.

Application with OCR functionality

OCR can be used to power different systems, such as mobile applications or computer-based applications. Watanabe et al. (2003) developed a mobile application that extracts Japanese characters from a scene and translates them into English. The application enables the user to manually select the text region to be translated, which makes the translation available to users at the soonest possible time. One hundred forty-one scenes where Japanese characters were extracted were used to test the application. Findings indicated that the mobile application has an 85% performance rate.

Another study conducted in 2012 by Chammas et al. utilized OCR to enable a mobile application to translate Arabic text to the user's preferred language. The study focused on translating the names of villages and towns and restaurant menu entries. The mobile application was also able to recognize both handwritten and printed text resulting in a 2% error rate.

In 2014, Nurhayati et al. conducted research to integrate OCR in a mobile application that scanned the cooking recipe. They utilized Tesseract to convert scanned images to text and enable the users to search for recipes by their ingredients. They implemented this using a mobile application that a user can bring with them anytime, thus freeing the user from the need to bring a recipe book with them.

In a study by de Luna (2020), the researcher created a microcontroller-based braille platform utilizing a raspberry pi and Tesseract OCR engine. The platform utilized OCR to generate braille characters from the scanned text in a tactile display. They determined that a font size of 12 and below for Arial font results in an accuracy of 0%. de Luna further stated that 85% accuracy could be attained if the font size is increased to 18, thus concluding that an increase in accuracy could be attained if the font size of the material being processed is also increased.

Another notable use of OCR is on plate number recognition. Patel et al. (2012) conducted a study to extract text from vehicle plate numbers using Tesseract. They noticed that converting images to grayscale yielded an increase in the accuracy of text extraction compared to colored images. They also compared Tesseract to a proprietary OCR tool Transym OCR. The researchers observed that Transym OCR converts an image to grayscale before it performs character recognition. To compare correctly, the group of Patel also converted the vehicle plate number to grayscale using an algorithm they developed before feeding it to Tesseract OCR. Results of the test they conducted revealed that Tesseract OCR, although a command-line-based tool, was able to extract text from vehicle plate numbers more accurately compared to Transym OCR.

Singh and Bhushan (2019) created an automatic number plate recognition (ANPR) system using Tesseract to identify and extract text present in plate numbers of vehicles flying through India's roads and highways. Employing deep neural networks to identify text within a vehicle plate number, they were able to get a 99% accuracy rate.

In a similar but more recent study, Adedayo and Agunloye (2021) created a real-time plate number recognition and detection system using Tesseract in Nigeria. They detect plate numbers from stationary vehicles using any input source. The acquired image is then processed using OpenCV, a python library for image processing. Their study resulted in 75% accuracy in recognizing license plate numbers. Researchers also recorded an average of 90 milliseconds for the OCR engine to process the license plate numbers properly.

It is prevalent in the literature that the result of OCR is not 100% accurate for the time being. Good results can be obtained by subjecting source images to some form of preprocessing, such as converting images to black and white. Several studies mentioned above also provided insight into how fast OCR engines can extract text from a source image.

OCR Applications in the Philippines

Regarding research conducted abroad, it is safe to say that OCR technology is getting much attention, which is also true in the Philippines. Researchers in the Philippines also conducted studies that utilized OCR in digital document classification, counterfeit currency detection, and transliteration of Baybayin text.

In 2020, Jayoma et al. created an OCR-enhanced application for the Department of Social Welfare and Development (DSWD) Caraga. The application accepts digital document files, extracts texts, classifies the documents, stores and makes them available for searching. The researchers aim to automate the classification process for documents of the DSWD Caraga. They used Python-Tesseract (PyTesseract), a Python-based wrapper for the Tesseract OCR engine. According to the researchers, the developed system was hosted on a local server within the ICT Center of DSWD Caraga.

Apoloni et al. (2022) developed a Philippine counterfeit currency detector equipped with OCR. The prototype was developed using Raspberry-pi and could detect the watermark, asymmetric serial number, see-through print, and security tread, all of which fall under the level-1 security features of Philippine currency. Their research utilized Tesseract OCR to identify data within the captured image and transform it into a string. The researchers subjected a combination of authentic and fake Philippine currency denominations from 20 to 1,000-peso bills to an experiment using the counterfeit currency detector. In their experiment, they observed that the detector worked best with the 200-peso bill as it shows an accuracy of 100%. Furthermore, the overall result of their experiment shows an average accuracy rating of 95.86% for all the banknotes. They concluded that the detector could distinguish between counterfeit and authentic Philippine currency.

Pino et al. (2021a, 2021b, 2022) conducted different studies on recognizing Baybayin characters. Their first study, published in 2021, utilized a support vector machine in classifying a Baybayin script. The OCR system they developed identified whether an image contains Baybayin or Latin script with a 98.41% accuracy rating. In a succeeding study published in the same year, they proposed a system that would perform transliteration of Baybayin text to its Latin equivalent at a word level. The researchers employed the OCR system, which utilized a support vector machine that they developed and implemented in an earlier study. Experimenting with 1000 publicly available images containing Baybayin written word, the system recorded a 97.9% accuracy rating. A more recent study by the authors published in 2022 implemented an OCR system that distinguishes between Latin and Baybayin text on a block level. Their system converts the identified Baybayin words to their corresponding Latin equivalent while the Latin text is displayed as it is. Furthermore, the system they developed identifies the Baybayin transliterated word by putting a red rectangular indicator to the word. They subjected the system to an experiment using 110 Baybayin and Latin block text images, among which 103 were correctly identified.

MOTIVATION

Adding OCR capability to a digital asset management system is a promising tool that will not only lessen the workload of information professionals in indexing and processing data but also make the digitized materials readily available and content searchable to its users.

Leveraging OCR leads to developing a simple document retrieval tool using available open-source

software. The aim is to exploit the potential of OCR and utilize it to query the text content of digitized materials.

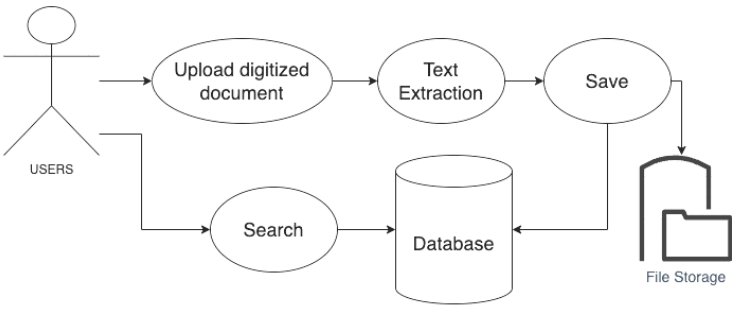
Several OCR software is available on the market, both proprietary and open-source. Well-known proprietary OCR software are Google Cloud Vision, Asprise OCR SDK, OmniPage, and Abbyy FineReader. Free OCR software, such as OCRFeeder, GOCR, OCRopus, and Tesseract, can be used, modified, and shared with others. For this prototype, Tesseract, one of the most advanced, free OCR software supporting more than 100 languages listed in its GitHub repository (Tesseract OCR, 2021), was utilized. Although lacking a graphical user interface, this poses an advantage for developers as it can easily be integrated into the information retrieval system being developed.

SYSTEM DESIGN

System Business Process

The prototype follows the system process as illustrated in Figure 1. The process starts by uploading digitized documents from which text will be extracted. The extracted text, together with other document characteristics like title and filename, will be saved to the database while the file will be saved on file storage within the server. Users of the prototype will be able to search data stored in the database.

Figure 1
System Process



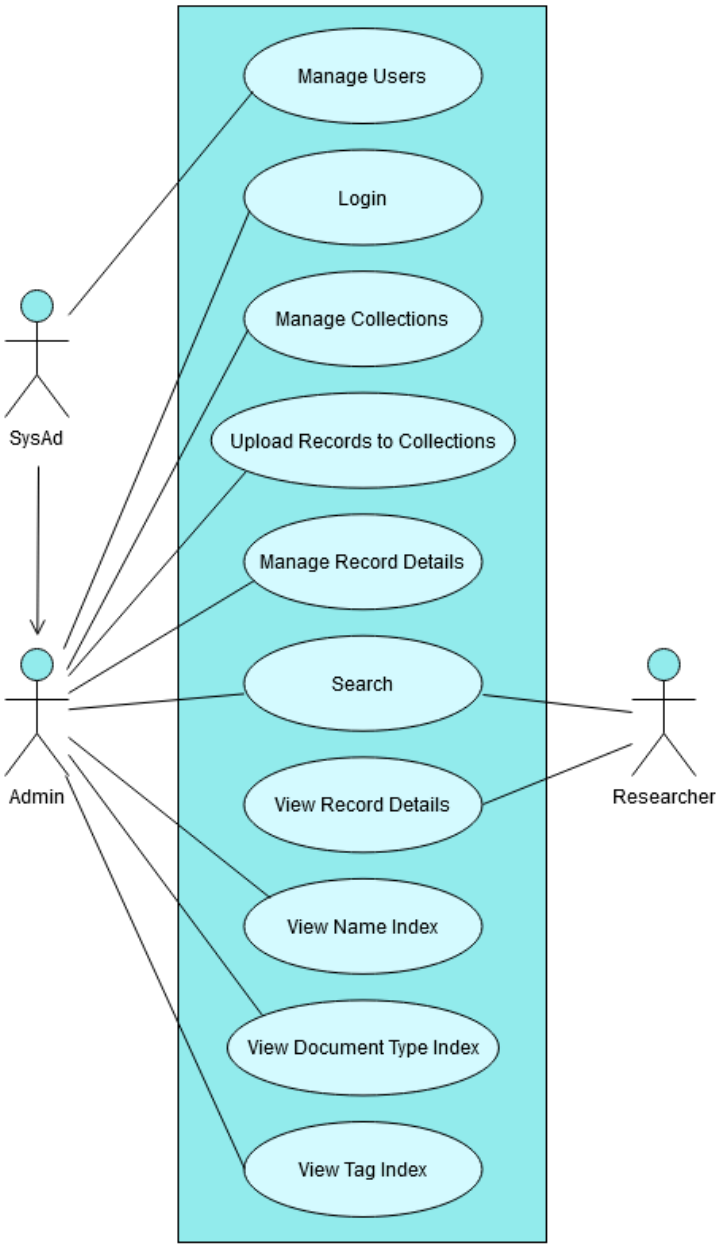
System Use Case

The prototype's design (Figure 2) is intended for three types of users: System Administrator (SysAd), Admin, and Researchers. Admin users can log in, manage the collection, upload records to collections, manage records details, search, view name index, view document type index, and view tag index within the prototype. A particular type of Admin user, the SysAd, has the added capability to manage users of the system. Researchers are non-authenticated users that can search and view record details.

Entity Relationship Diagram

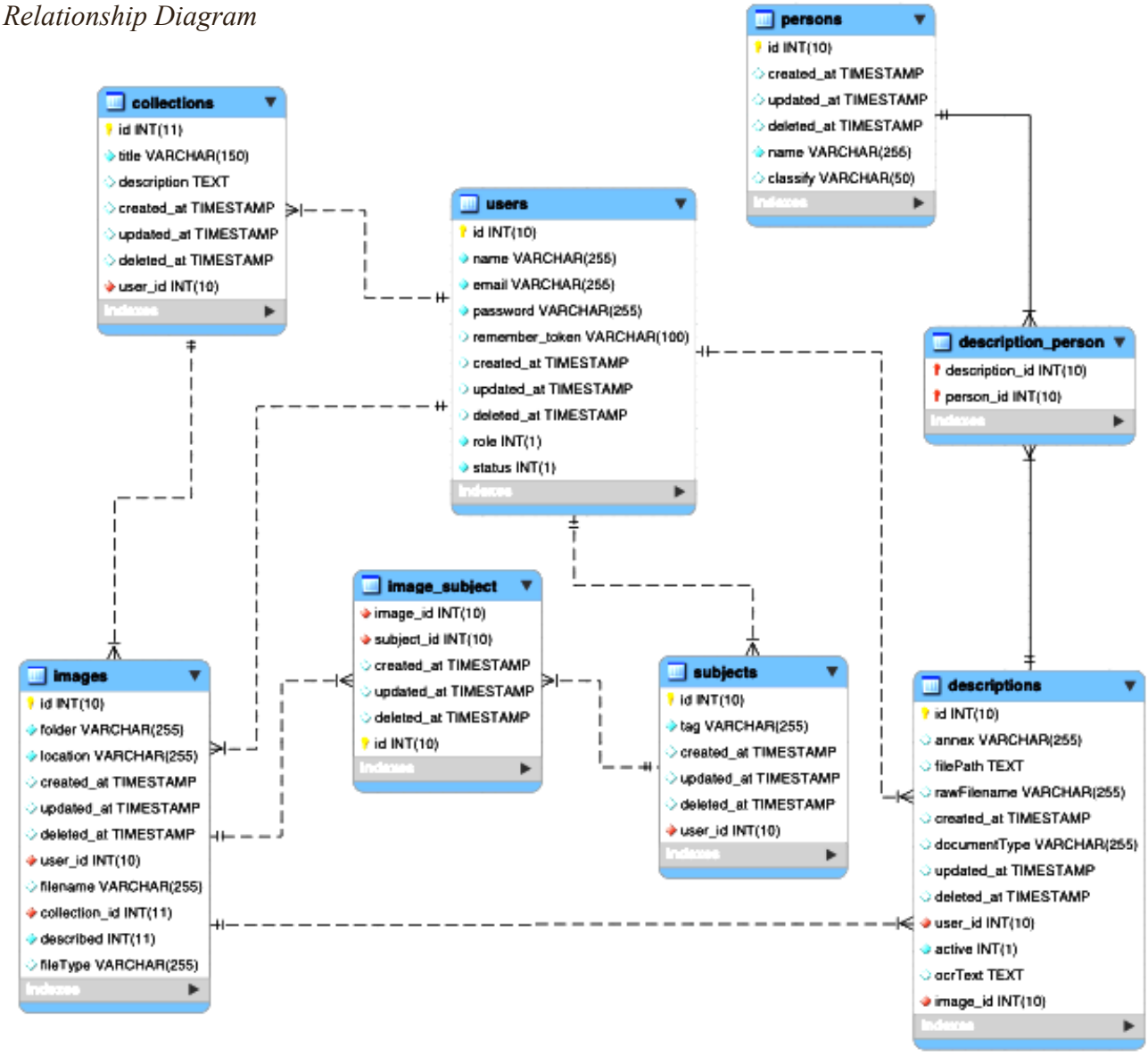
The entity-relationship diagram in Figure 3 represents the database system used in the prototype. The database comprises six main entities: users, collections, records, subjects, persons, and descriptions. Each entity stores related records using the assigned primary key, thus making it easy to retrieve linked entries.

Figure 2
System Use Case



Surmieda, J. S. (2022). OCR-enhanced digital asset management system: Prototype design and construction. *Philippine Journal of Librarianship and Information Studies, 42*(2), 33–43.

Figure 3
Entity Relationship Diagram



Software

The prototype was developed using Laravel, Tesseract, Ghostscript, MySQL, PHP, and Apache HTTP Server, which was implemented in local development. These kinds of software are open-sourced and readily available in different operating systems.

PHP: Hypertext Preprocessor or PHP is an open-sourced, general-purpose scripting language used for web development. PHP started in the mid-1990s and grew as one of the leading languages powering the web (PHP Group, 2022).

Apache HTTP Server is a mature software developed in February 1995 (Apache HTTP Server Project, n.d.). Developed using collaborative development, it provided a well-documented, commercial-grade, and robust HTTP (Web) server.

Laravel is a PHP-based framework that makes prototype development a lot quicker. This framework utilizes a model, view, and controller architecture. In this development architecture, the system logic is separated from the view.

Ghostscript is a free interpreter with 30 years of active development for PostScript language and PDF files (Artifex Software, 2022). Ghostscript can convert a PDF document to an image in TIFF format.

Tesseract is an open-source software currently hosted by Google and can recognize more than 100 languages (Tesseract OCR, 2022). Tesseract was developed originally at Hewlett-Packard between 1985 and 1994 before the same company open-sourced it in 2005. Tesseract currently supports different output formats such as plain text, hOCR (HTML), PDF, and invisible-text-only PDF.

MySQL, currently owned by Oracle Corporation, can be used as an open-source product or with a standard commercial license from Oracle (MySQL 5.7 Reference Manual, 2023). MySQL servers can be used in mission-critical and heavy-load production servers.

PROTOTYPE DEVELOPMENT

Prototype development was done using Ubuntu 20.04.3 desktop edition, an open-source operating system (The Story of Ubuntu, n.d.). Ubuntu desktop can be downloaded from https://ubuntu.com/desktop.

The application has five main functions; staff management, collection management, search, name index page, document type index page, and subject index page.

Interface

Figure 4 shows the staff page of the OCR-enhanced information retrieval system. A list of system users with their roles is displayed. System administrators can also edit the details of each system user using this page. The functionality to add a new system user can also be found on this page.

Figure 4
Staff Page

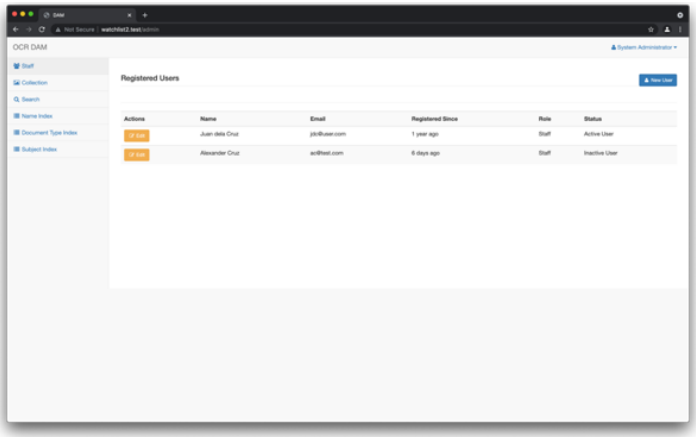


Figure 5
Collections Page

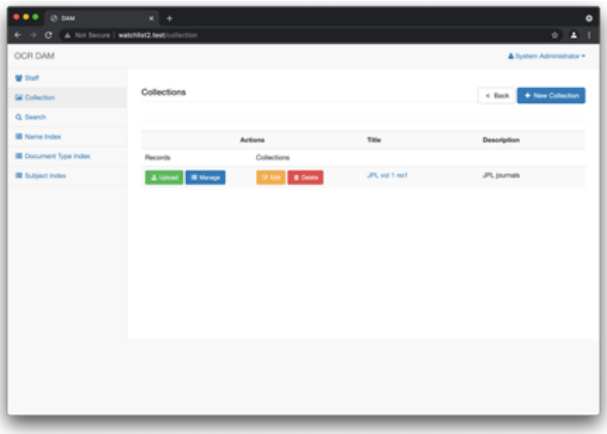


Figure 6
Uploading Record Interface

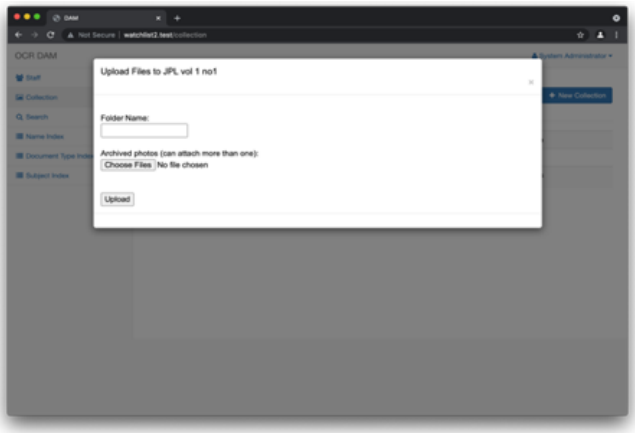


Figure 7
Manage Function Page

Surmieda, J. S. (2022). OCR-enhanced digital asset management system: Prototype design and construction. *Philippine Journal of Librarianship and Information Studies, 42*(2), 33–43.

Figure 8
Search Page

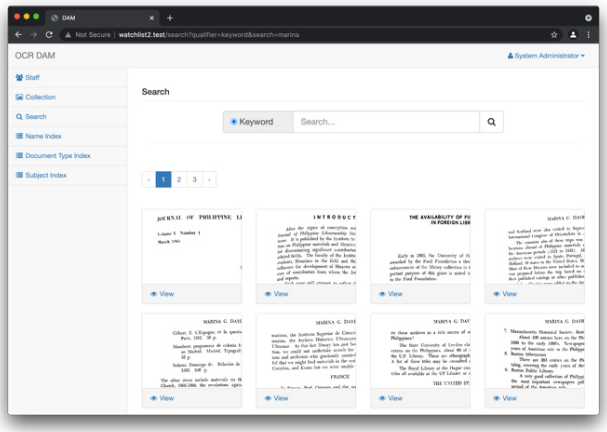
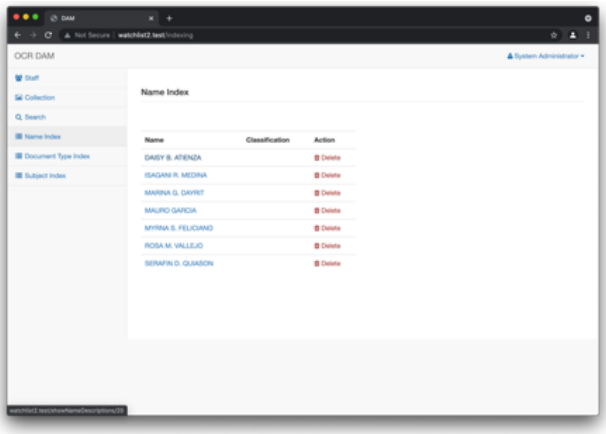


Figure 9
Name Index Page



Most system functionality is located on the Collection page (Figure 5). Within this page, system users can add a new collection, edit, and delete it. The functionality to upload records to a specific collection is accessible using the upload button. Figure 6 shows the interface where the system user indicates the folder and selects the images to upload to the system. A system user can utilize the "Manage" function to check, edit and tag an image. The Manage function page displays the extracted text from the image and some input fields for tagging individual names and adding a specific subject to the image. Figure 7 is an example interface where the extracted text from the document is presented alongside the document. This functionality allows Admin users to further improve the text acquired by OCR.

The search interface is found in Figure 8. On this page, a user provides a keyword to search the system. If that keyword is present in the system, the image containing the provided keyword is displayed on the screen. The images displayed to the user can be clicked, and a larger version of the image will be presented.

Figures 9, 10, and 11 are the different index pages: Name Index, Document Type Index, and Subject Index. On these pages, different terms are displayed, and once a term is clicked, images tagged with specific terms are displayed. The displayed images can also be clicked to enlarge.

Figure 10
Document Type Index Page

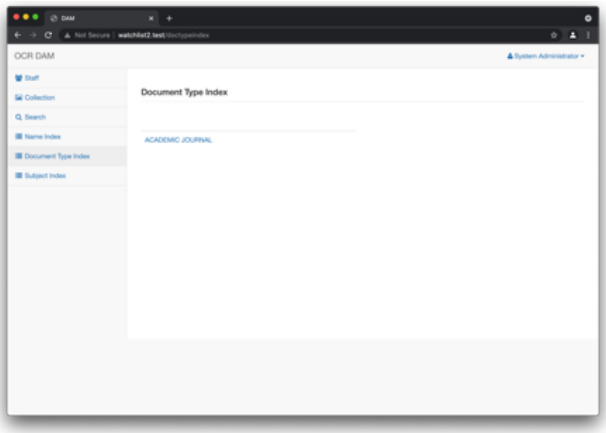
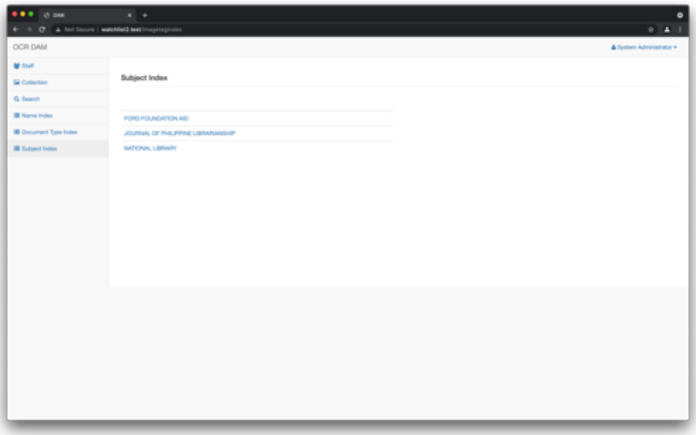


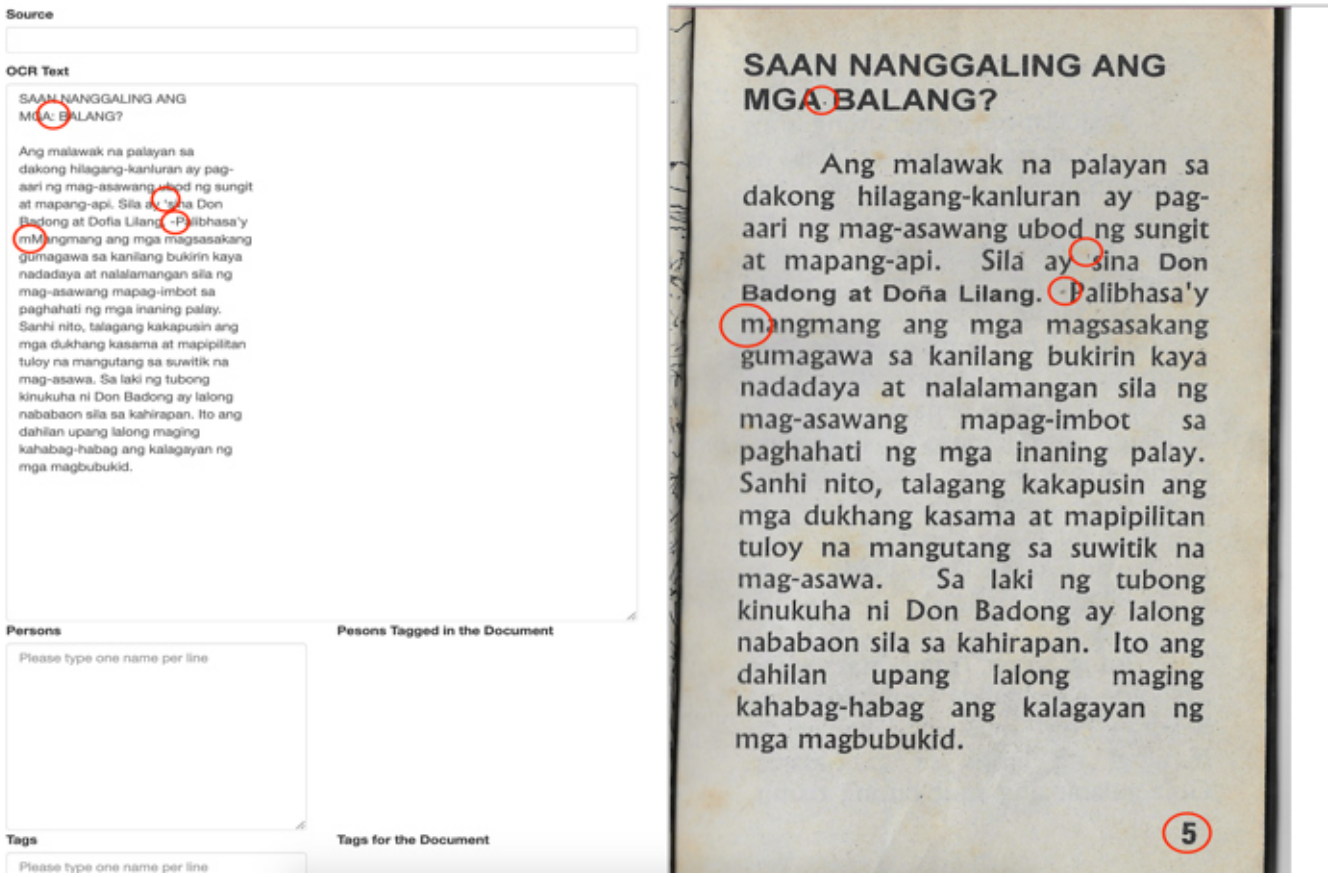
Figure 11
Subject Index Page



TESTING

To check if the system can extract words from an image and store it in the database, 15 images containing Filipino words were scanned. A flatbed scanner was used to scan the images, and they were saved in a portable network graphic (PNG) type.

Figure 12
OCR Sample A

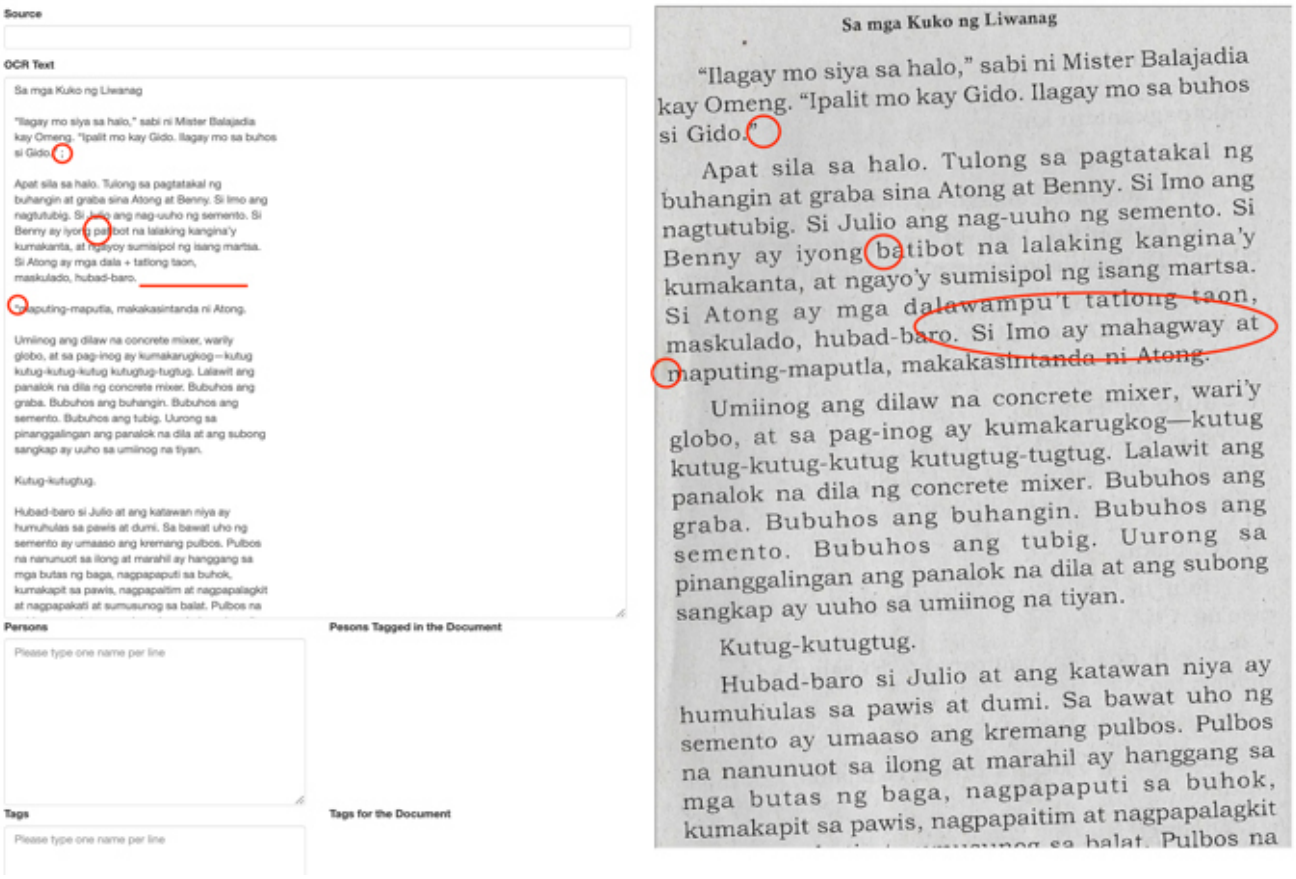


Note. Sample page from *Sari-saring alamat tungkol sa mga hayop, halaman at pook: pantulong sa mga guro at mag-aaral!* (p. 5), by T.C. Basadre, 2003, TCB Book Supply. Copyright 2003 by TCB Book Supply and Teresita C. Basadre.

Table 1
OCR Word Count

SCANNED IMAGE	WORD COUNT	OCR TEXT		PERCENTAGE
		Correct	*Error*	
T1	50	47	3	94.00%
T2	97	92	5	94.85%
T3	103	99	4	96.12%
T4	91	88	3	96.70%
T5	99	93	6	93.94%
T6	104	99	5	95.19%
T7	89	85	4	95.51%
T8	83	79	4	95.18%
T9	189	189	0	100.00%
T10	212	207	5	97.64%
T11	181	178	3	98.34%
T12	242	236	6	97.52%
T13	116	104	12	89.66%
T14	205	193	12	94.15%
T15	170	164	6	96.47%
TOTAL	2031	1953	78	**96.16%**
AVERAGE	135.4	130.2	5.2	**96.16%**

Figure 13
OCR Sample B



Note. Sample page from *Sa mga kuko ng liwanag* (p. 6), by E. Reyes, 2007, C&E Publishing. Copyright 2007 by C&E Publishing and Edgardo M. Reyes.

Words from each of the pages were counted and recorded in Table 1. Images were uploaded as-is, and no image enhancements were done. The result of the OCR was then checked using the manage function of the OCR-enhanced information retrieval. OCR text was evaluated against the scanned image text, and the number of errors and correct text was counted.

Table 1 shows the OCR results for the 15 images uploaded to the OCR-enhanced digital asset management system. In total, 2,031 words were counted from the 15 scanned images; each image contained 134.5 words on average. The OCR-enhanced information retrieval system was able to recognize and record 1,953 words correctly, and 78 errors occurred during the OCR process. On average, 130.2 words were recognized and recorded, and an error of 5.2 words per image was observed. The system shows a 96% accuracy rate in terms of words recognized.

Figures 12 and 13 are examples of images fed to the OCR-enhanced asset management system. The image in Figure 12 displays a light speckle on the page, which was interpreted as a period in the OCR text. The image in Figure 13 was not properly aligned, resulting in discrepancies in the OCR-extracted text. Looking closely at the errors produced by the OCR (see Figures 12 and 13), it can be observed that these were the results of poorly prepared images. The result shows that image preprocessing, such as image alignment, image cleaning, and conversion to black and white, affects the accuracy of OCR results.

CONCLUSION

A functional document retrieval tool using open-source software was successfully developed by integrating OCR functionality. Initial testing proved that it could help in information discovery. Notably, the use of Tesseract is not perfect, but its capability to

recognize more than 100 languages and embed them in a digital repository is valuable in information discovery and retrieval.

The prototype developed in this paper is very much in the early stages and needs more enhancement and development to further improve its capabilities and functionalities. The success of this prototype showed potential for the future of OCR for libraries and information centers. With the abundance of open-source technologies, designing and implementing an OCR-enhanced information system is attainable for system developers at a low cost.

Libraries, archives, and similar information centers can take advantage of the potential of an OCR-enhanced system. As its main thrust is to provide searchability and access to its collections, implementing an OCR-enhanced information system makes it possible for materials to be readily searchable and thus gives quick access to raw and plain contents on each material.

DECLARATION ON CONFLICTING INTERESTS

The author declared no potential conflicts of interest with respect to research, authorship, and/or publication of this article.

DECLARATION ON SOURCES OF FUNDING

The author received no financial support for the research, authorship, and/or publication of this article.

REFERENCES

Adedayo, K. D., & Agunloye, A. O. (2021). Real-time automated detection and recognition of Nigerian license plates via deep learning single shot detection and optical character recognition. *Computer and Information Science*, *14*(4), 11–19. https://doi.org/10.5539/cis.v14n4p11

Apache HTTP Server Project. (n.d.). Apache Software Foundation. Retrieved February 10, 2022, from https://httpd.apache.org/ABOUT_APACHE.html

Apoloni, J. M. R., Escueta, S. D. G., & Sese, J. T. (2022). Philippine currency counterfeit detector using image processing. *2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA)*, 436–441. https://doi.org/10.1109/CSPA55076.2022.9781980

Artifex Software. (2022). *Ghostscript.* Retrieved February 10, 2022, from https://www.ghostscript.com/

Basadre, T. C. (2003). *Sari-saring alamat tungkol sa mga hayop, halaman at pook: Pantulong sa mga guro at mag-aaral!* (C. S. Canonigo, Ed.; Unang Limbag [=1st edition]).). TCB Book Supply.

Chammas, E., Mokbel, C., Al Hajj Mohamad, R., Oprean, C., Sulem, L. L., & Chollet, G. (2012). Reducing language barriers for tourists using handwriting recognition enabled mobile application. *2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, 20–23. https://doi.org/10.1109/ICTEA.2012.6462868

Chiron, G., Doucet, A., Coustaty, M., Visani, M., & Moreux, J.-P. (2017). Impact of OCR errors on the use of digital libraries: Towards a better access to information. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL),* pp. 249–252. https://doi.org/10.1109/JCDL.2017.7991582

De Luna, R. G. (2020). A Tesseract-based optical character recognition for a text-to-braille code conversion. *International Journal on Advanced Science, Engineering and Information Technology*, *10*(1), 128–136. https://doi.org/10.18517/ijaseit.10.1.9956

Gillies, A., Hepp, D., Rovner, R., & Whalen, M. (1995). Handwritten text recognition system for processing census forms. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, *3*, pp. 2335–2340. https://doi.org/10.1109/ICSMC.1995.538130

Jayoma, J. M., Moyon, E. S., & Morales, E. M. O. (2020). OCR based document archiving and indexing using PyTesseract: A record management system for DSWD Caraga, Philippines. *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–6. https://doi.org/10.1109/HNICEM51456.2020.9400000

MySQL 5.7 Reference Manual. (2023). Oracle. https://dev.mysql.com/doc/refman/5.7/en/introduction.html

Nurhayati, Risda, B. C., & Masruroh, S. U. (2014). Optical character recognition feature implementation in cooking recipe application using tesseract Google project. *2014 International Conference on Cyber and IT Service Management (CITSM)*, 44–47. https://doi.org/10.1109/CITSM.2014.7042173

Patel, C., Patel, A., & Patel, D. (2012). Optical character recognition by open source OCR tool Tesseract: A case study. *International Journal of Computer Applications*, *55*(10), 50–56. https://doi.org/10.5120/8794-2784

PHP Group. (2022). *PHP: History of PHP and Related Projects—Manual.* Retrieved February 10, 2022, from https://www.php.net/manual/en/history.php

Pino, R., Mendoza, R., & Sambayan, R. (2021,

February 15). Optical character recognition system for Baybayin scripts using support vector machine. *PeerJ Computer Science*, *7*, e360. https://doi.org/10.7717/peerj-cs.360

Pino, R., Mendoza, R., & Sambayan, R. (2021, June 16). A Baybayin word recognition system. *PeerJ Computer Science*, *7*, e596. https://doi.org/10.7717/peerj-cs.596

Pino, R., Mendoza, R., & Sambayan, R. (2022). Block-level optical character recognition system for automatic transliterations of Baybayin texts using support vector machine. *Philippine Journal of Science*, *151*(1), 303–315. https://doi.org/10.56899/151.01.23

Raj, R., & Kos, A. (2022). A comprehensive study of optical character recognition. In *2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*, pp. 151–154. https://doi.org/10.23919/MIXDES55591.2022.9837974

Reitz, J. (2013). Optical character recognition. In *Online Dictionary for Library and Information Science.* ABC-CLIO LLC. Retrieved February 15, 2022, from https://products.abc-clio.com/ODLIS/odlis_o.aspx#opticalcharacter

Reyes, E. (2007). *Sa mga kuko ng liwanag* (Ikalawang edisyon [=2nd edition]). C & E Publishing.

Robby, G. A., Tandra, A., Susanto, I., Harefa, J., & Chowanda, A. (2019). Implementation of optical character recognition using Tesseract with the Javanese script target in Android application. *Procedia Computer Science*, *157*, 499–505. https://doi.org/10.1016/j.procs.2019.09.006

Salimah, U., Maharani, V., & Nursyanti, R. (2021). Automatic license plate recognition using optical character recognition. *IOP Conference Series: Materials Science and Engineering, 1115,* 012023. https://doi.org/10.1088/1757-899X/1115/1/012023

Singh, J., & Bhushan, B. (2019). Real-time Indian license plate detection using deep neural networks and optical character recognition using LSTM Tesseract. *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 347–352. https://doi.org/10.1109/ICCCIS48478.2019.8974469

Tesseract OCR. (2022). [C++]. Tesseract-ocr. https://github.com/tesseract-ocr/tesseract (Original work published 2014)

The story of Ubuntu (n.d.). Canonical. Retrieved February 15, 2022, from https://ubuntu.com/about

Thomas, E. (2021 July 05). Turning letters into tones: A century ago, the optophone allowed blind people to hear the printed word. *IEEE Spectrum*, *58*(7), 34–39. https://doi.org/10.1109/MSPEC.2021.9475413

Watanabe, Y., Sono, K., Yokomizo, K., & Okada, Y. (2003, August 18). Translation camera on mobile phone. *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, *2*, II-177–II-180. https://doi.org/10.1109/ICME.2003.1221582

AUTHOR BIOGRAPHY

Janny S. Surmieda is a full-time faculty member of the UP School of Library and Information Studies. He teaches classes on Indexing and Abstracting, Library Management, and ICT courses on programming, database structure and web applications in Library and Information Science. Before joining UPSLIS, he was a practicing Librarian for ten years and has worked on projects such as web application, website design and library system development. His research interests include knowledge organization, knowledge discovery, data mining, bibliometrics, natural-language processing and meta-analysis in information science. He completed both his Bachelor and Masters degree in Library and Information Science from UPSLIS.

Email: *janny@slis.upd.edu.ph*

This work is licensed under a Creative Commons Attribution 4.0 International License.

The PhJLIS is published by the School of Library and Information Studies, University of the Philippines Diliman.
ISSN 2719-0471 (Online)