ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20220408162739

Professional paper
Received: 2022-04-08, Accepted: 2022-04-09

A Personalized and Scalable Machine Learning-Based File Management System

Bansal Veena*, Sati Dhiraj

Abstract: In this work, we present a hybrid image and document filing system that we have built. When a user wants to store a file in the system, it is processed to generate tags using an appropriate open-source machine learning system. Presently, we use OpenCV and Tesseract OCR for tagging files. OpenCV recognizes objects in the images and TesserAct recognizes text in the image. An image file is processed for object recognition using OpenCV as well for text/captions process using TesserAct, which are used for tagging the file. All other files are processed using Tesseract only for generating tags. The user can also enter their own tags. A database system has been built that stores tags and the image path. Every file is stored with its owner identification and it is time-stamped. The system has a client-server architecture and can be used for storing and retrieving a large number of files. This is a highly scalable system.

Keywords: document database; file tags; image tags; object detection; personalized filing system; scalable; tags database

1 INTRODUCTION

There are many organizations and individuals handling large volumes of documents. These organizations need to manage them efficiently without having to deploy expensive and complicated solutions available. Such an organization has documents containing text, images, videos and links in multiple formats in large numbers. Many of the documents are in paper format, physical media and others are in digital formats. These organizations need to scan and digitize the documents fast and process the documents in parallel on many computers. The documents must be stored in a way that facilitates retrieval based on content or on the purpose of the search. A system that retrieves desired documents with a near 100% recall rate and decent precision is desired. For micro and small organization, the cost of the system is also important, as they are usually tight on the budget. A massive parallel storage system does not fit in their budget. In this paper, a personalized file management system is presented that has been implemented and tested.

The activities that are part of a file management system include creating folders, uploading files and folders, naming, saving, searching and deleting files and folders [1]. There are many algorithms and techniques to retrieve information that focus on different aspects of performance. Recall rate, precision and speed are three main parameters for evaluating an information retrieval algorithm. Every operating system provides a directory structure for organizing the data and a search facility for finding a file by its name. A content-based image retrieval (CBIR) system focuses on retrieving images similar to a query image [2]. A text-based image retrieval (TBIR) system retrieves images that contain the query text. Images may also be retrieved by querying the objects contained in the images provided images have been tags associated with them. Automatic tagging has performance issues and manual tagging is very expensive. More advanced systems such as google photos automatically tag images and provide options to the user to include additional tags.

The command *grep* on Linux and *search* on windows can retrieve all documents that contain the query word in text documents. It has been shown that people spend more time navigating than searching as the search results are often less than satisfactory and search tools are complex [3, 4]. Individuals maintain thousands of files and folders [5] and the file management system provided by an operating system is just about adequate [3, 6].

All these systems become inadequate when an organization is in a business that involves scanning, storing and processing millions of documents and costs, privacy and regulations are also issues [3, 7]. There are systems such as Google file system (GFS) and Apache Hadoop among many others, which are designed for managing and processing volume and variety of data [8]. These systems have their own sharing and scheduling mechanisms. These systems are built using a client-server architecture. These systems are resource hungry, complex and expensive.

Micro and small organizations need a lightweight system to organize the data and facilitate retrieval based on contents as well as metadata. The scanned documents or images need to be processed to obtain their contents for retrieval [9]. The file management system may evolve as newer techniques become available for extracting the contents of the scanned images. It is also possible to deploy multiple extraction processes for better performance. For instance, the system could deploy two optical character recognizers (OCRs) for recognizing the text and two image-processing methods to identify objects contained in a scanned document. Depending on the area of the strength of each of the processors, the union of the output will give a superior set of contents. The system and its components have been described in the next section followed by a discussion and conclusion in section 3.

2 THE SYSTEM

The system has been built using a client-server architecture that facilitates scaling and parallel processing. The client system consists of four major modules- login, upload, process and search. The server system consists of a user management module, an upload management module, a processing management module and a search management module. The user management module does as the name suggests and requires no elaboration. The upload management module manages storage locations, types of documents that can be uploaded, keeps track of the documents in the repository and the processes that are available for extracting contents. The processing module manages the technologies available for extracting information and contents of the files uploaded into the system. The server is also responsible for managing the repository where the files in the raw form are stored. The search module is responsible for facilitating the search and rendering the results.

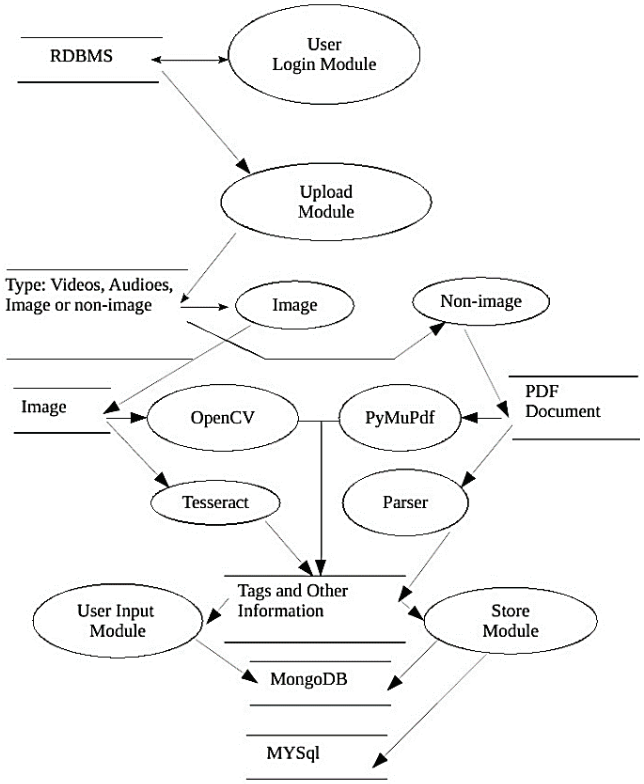


Figure 1 Data Flow Diagram(DFD) of the Client Module; an oval represents a functions, two parallel lines represent a data store and arrows indicate flow of data

The schematic diagram of the client system is shown in Fig. 1. We have used Data Flow Diagram (DFD) notation. This notation is a part of the unified modeling language. A function produces an output that goes to a temporary or permanent data store. The following function is executed when the required data becomes available in the data store. DFD naturally reveals processes that can progress in parallel.

2.1 The Upload Module

The system provides a GUI for uploading a file. The input to the system may be a file that has been generated by scanning a paper document that may contain text, images or both. A sample image is shown in Fig. 2.

The input can also be a document that is in digital form. The user can store a variety of documents in the system. The documents could be invoices, email, legal documents, mails, brochures or any other type of document originating from in-house systems, government agencies, business associates, legal agencies.

Every document has a Unique Identifier, Document type, date of upload, identification of the person who uploaded and document-specific details.

2.2 Processors

As discussed in the last subsection, there are three types of documents- images that are not editable, pdf files that may be edited using special editors and documents that are editable. Each type of document is processed differently.

A scanned file is processed using an appropriate machine learning tool to extract the contents of the file. Presently, a scanned file is processed using image detection function of OpenCV library which is a pre-trained image recognition system to obtain the objects contained in the image. It is also processed by Tesseract, an OCR, to extract the structure of the file and the text contained in the file. The extracted contents are further processed to obtain information that is useful for search operations. All stop words, punctuation marks and words of length two or less are removed. There are many open source software packages available for processing images and text images. An organization may choose other packages such as scikit-image for image processing and EasyOCR for text image processing.

If the document is in editable form and if the user has declared the type of the document, the document type-specific processing is done. For an email, all the fields are extracted such as to, from, subject, forwarded, date, time, attachments etc. Attachments if any are also stored in the repository.

Similarly, an invoice has an issuer, the date of issue, PAN number, invoice number, amount which are extracted from the invoice. If the type of the document is not known, only unique words present in the document are extracted. The system maintains a list of all unique words and objects. This list is available to the user at the time of search to pick keywords for search.

If a document is in pdf format, it is processed using PyMuPDF to extract the structure and contents of the document. A multipage file is divided into individual pages to facilitate search.

The processor outputs the location of each unit extracted. At the highest level, the location is the file and at the finest level, it is the line number and the page number. Notice that we have used pre-trained processors (OpenCV and Tesseract) and no training is required.

The output of the processors is inserted in a dynamic relational database MySQL and NoSql database MongoBD. Information about each file uploaded is also stored in both databases. The summarized content is stored in a MongoDB database. The advantage of storing contents in MongoDB is that search facilities provided by it become available. The front end assembles the input provided by the user into an SQL query that is translated into a MongoDB query using the MongoDB query transpiler.

Every document is saved in its original form as well as in pdf format. The advantage of saving documents in pdf format is that the rendering becomes easy across platforms. The files are organized into a repository.

2.3 The Search Module

The search module provides a graphical user interface to the user. It has been built to provide support for regular expressions. A user can use *and*, *or* and *negation* operations for constituting a search string. The user first specifies the type of the files and the system renders additional relevant fields for the user to fill in. For an invoice, the user can select the name of the issuer, the range of date of the issue, the amount range, the range of the upload of the invoices, and a set of keywords. The set of keywords may be selected from a dropdown list. The search results are then rendered on the user screen, which includes a link to the original file.

2.4 Re-processing

There are significant advances being made in the OCR, Handwriting recognition, object detection and natural language processing. The server can add more processors and update its existing processors. Since the system has a repository of the original documents, the server can process its entire repository or selected files regularly without impacting the operations. This is done using batch processing on idle desktops.

The discussion so far has been centered on uploading files. However, the system permits a user to upload a folder also. The structure of the folder is maintained. The objective of maintaining the folder structure is to take advantage of the information in folder names and hierarchy. Presently, the system relies on the security provided by the operating system and no additional security features have been built.

3 DISCUSSION AND CONCLUSION

There are two parameters on which search results of a content-based file management system are judged, precision and recall rate. The precision and recall rate are defined as follows.

$$precision = \frac{|(relevant\ documents) \cap (retrieved\ documents)|}{|(retrieved\ documents)|}$$

$$recall = \frac{|(relevant\ documents) \cap (retrieved\ documents)|}{|(relevant\ documents)|}$$

Let us say, there are 100 files (relevant documents) in the system that match a query and the system retrieves a total of 90 files (retrieved documents) out of which 60 (relevant document retrieved) are from the matching files and 30 are some random files, then the recall rate is 60/100 = 60% and precision is 60/90 = 66.66%. The system described in this paper uses OpenCV and Tesseract for processing files for extracting contents. The recall rate and precision depend on the efficacy of the processors used. The state-of-the-art for extracting embedded text is 96.5% precision and 92.3% recall rate [10]. The image processing techniques are also improving and state-of-the-art for identifying common objects is at 77% accuracy [11]. The performance of the processors that have been used by us is slightly less than the state-of-the-art. However, the choice of processors is driven by their maturity and availability.

Since we have used pre-trained and publicly available processors, the performance of our system depends on the performance of the processors used. The system doesnot require separate testing for its performance.

3.1 An Example

Let us say, a user logs into the system and wants to upload the image shown in Fig. 2. The user declares that it is an image. The upload module uploads the files and the processing module is invoked to process the file. First it is processed by OpenCV [12] that detects an *aeroplane* and then it is processed by Tesseract [13] that detects text *N254EK.*

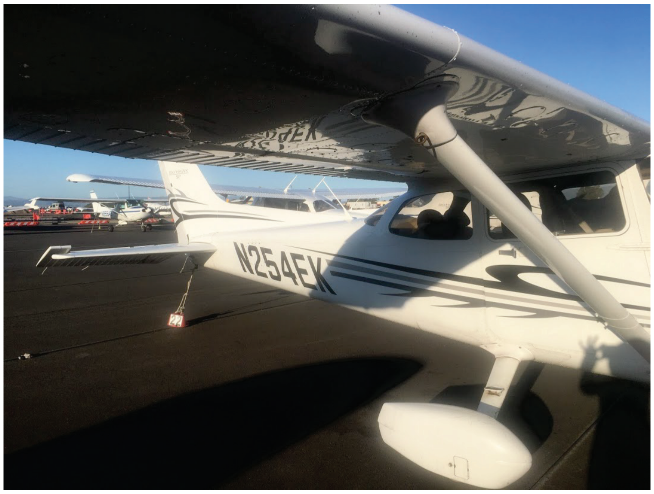
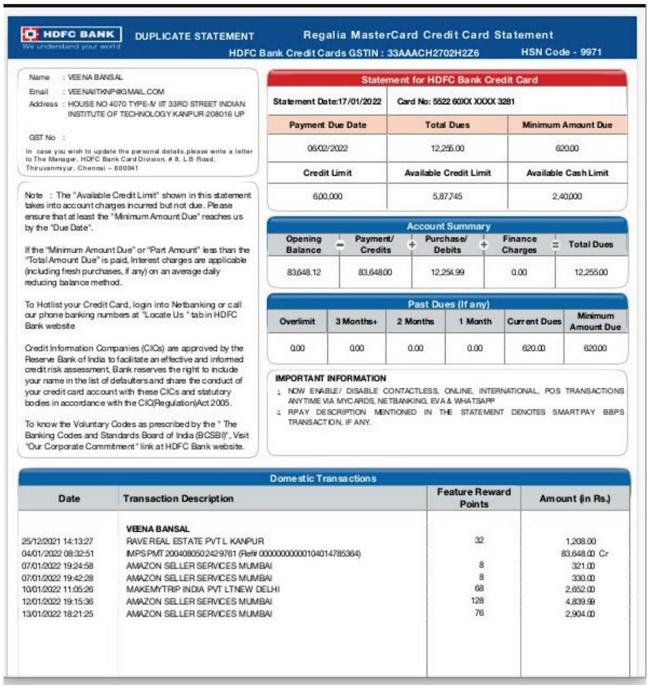


Figure 2 An example image that contains text and an object

The information about the file and metadata are stored in MySQL database. The output of the processers is stored in MongoDB. Metadata from the image such as Author, Title, Subject, Keywords, Category, Status and comments get stored in the MySQL database.

As another example, consider a pdf file shown in Fig. 3. This is credit card bill in pdf format. The processor is able to extract metadata about this file including author, creator, producer, header etc. and almost all the words contained in the pdf file. A subset of words extracted by PyMuPDF [14] as shown in Fig. 4. These words, metadata and file information make it possible to search for the file in multiple ways. One can search a file by name, its type, nature or by words and objects contained in the file. One can see that the output of the processor shown in Fig. 4 would require removal of stop words, punctuation marks and words that are less than 3 characters. This example clearly demonstrates the power of the file management system described in this paper. The output shown in Fig. 4 is stored in MongoDB to exploit the power of the search facility of MongoDB. The metadata is stored in dynamic MySql to enable entry of new metadata.



HDFC BANK
We understand your world
DUPLICATE STATEMENT
Regalia MasterCard Credit Card Statement
HDFC Bank Credit Cards GSTIN : 33AAACH2702H2Z6 HSN Code - 9971

Name : VEENA BANSAL
Email : VEENAIITKNP@GMAIL.COM
Address : HOUSE NO 4070 TYPE-IV IIT 33RD STREET INDIAN INSTITUTE OF TECHNOLOGY KANPUR-208016 UP

GST No :

In case you wish to update the personal details,please write a letter to The Manager, HDFC Bank Card Division, # 8, L B Road, Thiruvanmiyur, Chennai – 600041

Note : The "Available Credit Limit" shown in this statement takes into account charges incurred but not due. Please ensure that at least the "Minimum Amount Due" reaches us by the "Due Date".

If the "Minimum Amount Due" or "Part Amount" less than the "Total Amount Due" is paid, Interest charges are applicable (including fresh purchases, if any) on an average daily reducing balance method.

To Hotlist your Credit Card, login into Netbanking or call our phone banking numbers at "Locate Us " tab in HDFC Bank website

Credit Information Companies (CICs) are approved by the Reserve Bank of India to facilitate an effective and informed credit risk assessment, Bank reserves the right to include your name in the list of defaulters and share the conduct of your credit card account with these CICs and statutory bodies in accordance with the CIC(Regulation)Act 2005.

To know the Voluntary Codes as prescribed by the " The Banking Codes and Standards Board of India (BCSBI)", Visit "Our Corporate Commitment" link at HDFC Bank website.

Statement for HDFC Bank Credit Card

Statement Date:17/01/2022	Card No: 5522 60XX XXXX 3281	
Payment Due Date	**Total Dues**	**Minimum Amount Due**
06/02/2022	12,255.00	620.00
Credit Limit	**Available Credit Limit**	**Available Cash Limit**
6,00,000	5,87,745	2,40,000

Account Summary

Opening Balance	−	Payment/ Credits	+	Purchase/ Debits	+	Finance Charges	=	Total Dues
83,648.12		83,648.00		12,254.99		0.00		12,255.00

Past Dues (If any)

Overlimit	3 Months+	2 Months	1 Month	Current Dues	Minimum Amount Due
0.00	0.00	0.00	0.00	620.00	620.00

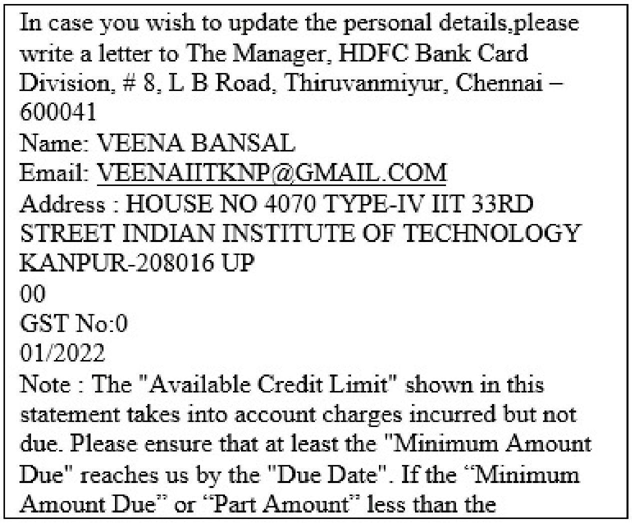
IMPORTANT INFORMATION

1. NOW ENABLE/ DISABLE CONTACTLESS, ONLINE, INTERNATIONAL, POS TRANSACTIONS ANYTIME VIA MYCARDS, NETBANKING, EVA & WHATSAPP
1. RPAY DESCRIPTION MENTIONED IN THE STATEMENT DENOTES SMARTPAY BBPS TRANSACTION, IF ANY.

Domestic Transactions

Date	Transaction Description	Feature Reward Points	Amount (in Rs.)
	VEENA BANSAL		
25/12/2021 14:13:27	RAVE REAL ESTATE PVT L KANPUR	32	1,208.00
04/01/2022 08:32:51	IMPS PMT 200408050242976 1 (Ref# 00000000000104014785364)		83,648.00 Cr
07/01/2022 19:24:58	AMAZON SELLER SERVICES MUMBAI	8	321.00
07/01/2022 19:42:28	AMAZON SELLER SERVICES MUMBAI	8	330.00
10/01/2022 11:05:26	MAKEMYTRIP INDIA PVT LTNEW DELHI	68	2,652.00
12/01/2022 19:15:36	AMAZON SELLER SERVICES MUMBAI	128	4,839.99
13/01/2022 18:21:25	AMAZON SELLER SERVICES MUMBAI	76	2,904.00

Figure 3 An example of a pdf file



In case you wish to update the personal details,please write a letter to The Manager, HDFC Bank Card Division, # 8, L B Road, Thiruvanmiyur, Chennai – 600041
Name: VEENA BANSAL
Email: VEENAIITKNP@GMAIL.COM
Address : HOUSE NO 4070 TYPE-IV IIT 33RD STREET INDIAN INSTITUTE OF TECHNOLOGY KANPUR-208016 UP
00
GST No:0
01/2022
Note : The "Available Credit Limit" shown in this statement takes into account charges incurred but not due. Please ensure that at least the "Minimum Amount Due" reaches us by the "Due Date". If the "Minimum Amount Due" or "Part Amount" less than the

Figure 4 A partial output produced by PyMuPDF for the pdf document shown in Figure 3

The present system has plenty of room for evolution in terms of automatic reorganization of the repository, databases and inclusion of graph databases for representing interlinking of documents and information. The system can also be improved by summarizing the contents and understanding the context of the search based on the user interaction with the system. Presently, the system can store audio and video files without any processing. The processor for audio and video files can be integrated. Additional information from tables, graphs, and tickers can be captured using web scraping which allows for searching beyond what is stored in the system. The demographic data may also be added to improve search operations. Encryption and masking of sensitive information in the documents are required and remain to be done.

In this paper, a personalized filing system has been described. The system is scalable as it is based on the client-server architecture and is designed to use the idle computing resources in the organization without resorting to expensive hardware.

Notice

The paper will be presented at MOTSP 2022 – 13th International Conference Management of Technology – Step to Sustainable Production, which will take place in Primošten/Dalmatia (Croatia) on June 8–10, 2022. The paper will not be published anywhere else.

4 REFERENCES

[1] Dinneen, J. D. & Julien, C. A. (2020). The ubiquitous digital file: A review of file management research. *Journal of the Association for Information Science and Technology*, *71*(1), E1-E32. https://doi.org/10.1002/asi.24222

[2] Tiwari, A. & Bansal, V. (2004). PATSEEK: Content Based Image Retrieval System for Patent Database. In *ICEB*, 1167-1171.

[3] Ravasio, P., Schär, S. G., & Krueger, H. (2004). In pursuit of desktop evolution: User problems and practices with modern desktop systems. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *11*(2), 156-180. https://doi.org/10.1145/1005361.1005363

[4] Fitchett, S. & Cockburn, A. (2015). An empirical characterisation of file retrieval. *International Journal of Human-Computer Studies*, *74*, 1-13. https://doi.org/10.1016/j.ijhcs.2014.10.002

[5] Hicks, B. J., Dong, A., Palmer, R., & Mcalpine, H. C. (2008). Organizing and managing personal electronic files: A mechanical engineer's perspective. *ACM Transactions on Information Systems (TOIS)*, *26*(4), 1-40. https://doi.org/10.1145/1402256.1402262

[6] Jones, W., Dumais, S., & Bruce, H. (2002). Once found, what then? A study of "keeping" behaviors in the personal use of web information. *Proceedings of the American Society for Information Science and Technology*, *39*(1), 391-402. https://doi.org/10.1002/meet.1450390143

[7] Agrawal, N., Bolosky, W. J., Douceur, J. R., & Lorch, J. R. (2007). A five-year study of file-system metadata. *ACM Transactions on Storage (TOS)*, *3*(3), 9-es. https://doi.org/10.1145/1288783.1288788

[8] Yang, J. (2015). From Google file system to omega: a decade of advancement in big data management at Google. In *2015 IEEE First International Conference on Big Data Computing Service and Applications* (pp. 249-255). IEEE. https://doi.org/10.1109/BigDataService.2015.47

[9] Choras, R. S. (2007). Image feature extraction techniques and their applications for CBIR and biometrics systems. *International journal of biology and biomedical engineering*, *1*(1), 6-16.

[10] Ye, J., Chen, Z., Liu, J., & Du, B. (2020). TextFuseNet: Scene Text Detection with Richer Fused Features. In *IJCAI* (pp. 516-522). https://doi.org/10.24963/ijcai.2020/72

[11] Ayachi, R., Said, Y., & Atri, M. (2021). A convolutional neural network to perform object detection and identification in visual large-scale data. *Big Data*, *9*(1), 41-52. https://doi.org/10.1089/big.2019.0093

[12] See https://www.opencv.org/

[13] See https://www.tesseract-ocr.github.io
[14] See https://www.pymupdf.readthedocs.io/en/latest/

Authors' contacts:

Veena Bansal, Associate Professor
(Corresponding author)
Indian Institute of Technology Kanpur,
Kalyanpur, Kanpur, 208016, India
+918953139897, veena@iitk.ac.in

Dhiraj Kumar Sati, Head - Solution & System Development
Capital Business Systems Pvt. Ltd**.**
288 A, Phase IV, Udyog Vihar, Sector 19
Gurugram, 122016, India
+919810077159, dhiraj.sati@cbsl-india.com
www.cbslgroup.in