A Text Extraction Approach towards Document Image Organisation for the Android Platform

M. Madhuram[1], Aruna Parameswaran[2]

[1]Assistant Professor, Department of Computer Science and Engineering SRM Institute of Science and Technology, Chennai, India
[2]Student, Department of Computer Science and Engineering SRM Institute of Science and Technology, Chennai, India

---***---

Abstract – *This paper demonstrates the implementation of a comprehensive gallery application based on the Android platform to handle document images. The fundamental objective of this app is to provide an automated interface for performing document image processing, text extraction, and organisation of relevant information. Over time, the widespread use of smartphones has resulted in the exponential growth of data. Despite the recent advancements in computing, storage techniques, and image processing, there is still a demand for highly optimised information retrieval methods, particularly for smartphones. Manual organisation of document images is an extremely time-consuming and tedious task. The proposed system presents an alternative approach by using Optical Character Recognition (OCR). The app employs the Firebase platform for detecting and extracting text. It also incorporates a Keyword Search feature, based on the KMP Algorithm, hence facilitating faster access to information. Additionally, this paper proposes a simple method for content-based categorisation of document images using a Bag-of-Words model.*

Key Words: Android Application, Document Image Analysis, Firebase, Image Processing, Optical Character Recognition software (OCR), Text Extraction

1. INTRODUCTION

Over the recent years, there has been a massive influx of data, particularly in the digitised form. As of 2017, it was reported that 90% of the world's data had been created in the span of just two years, at a rapid rate of 2.5 quintillion bytes a day. Consequently, data storage and information retrieval techniques have become more significant. Traditionally, data was stored physically in the paper format. However, recent advancements in computing, storage techniques (such as Big Data), and image processing have subsequently resulted in a shift towards digitised content.

Concentrated studies and research work on document images are being conducted presently. Document images vary from regular images in terms of their structural and spatial properties. Hence, in order to effectively process them, their text contents must be utilised. The term 'Document Image Analysis' (DIA) is used to refer to the set of techniques which perform digital conversion, text or image detection, and meaningful organisation of information. One of the most commonly used techniques in DIA is Optical Character Recognition (OCR). OCR is used to convert the printed or handwritten text present physically in a scanned document, image or scene-photo to an appropriate electronic format.

In some way or the other, every individual handles document images on a daily basis. Commonly used platforms for communication include messaging apps, e-mail and cloud storage services (Google Drive, Dropbox etc.). An example of commonly shared document images are the scanned class notes which are distributed to students and teachers. Other common examples include bills, receipts and even identity documents (such as passport, PAN card). Improvements in the standard of smartphone cameras, higher on-board storage, and the widespread availability of mobile scanner apps have collectively facilitated ease of information exchange.

The problem arises, however, when there is no singular method to organise large volumes of information. Eventually, this results in data misplacement and loss. As there may be multiple sources for these images, an automated system for syncing dynamically with the storage and further managing the information is required. The proposed work overcomes these issues by implementing features such as Keyword Search and Content-based Categorisation of Images.

The rest of this paper is organised as follows: Section II presents an overview of the literature survey done. Section III introduces the proposed system and proceeds to elaborate on its architecture and the purpose of each individual module. Section IV explains the algorithms used in the implementation of the system and the results are expanded upon in Section V. Section VI summarises the observations along with final concluding remarks and addresses the future scope of research.

2. LITERATURE REVIEW

Through the years, extensive research work has been done on Optical Character Recognition (OCR) and text extraction techniques. Increase in the volume of digitised information has further escalated these efforts. In this section, we shall review some of these systems and discuss about them in detail.

Neural Networks have been used extensively in several studies to perform Document Image Analysis. [3] Used a CNN to classify document images. The size of the image was initially reduced to prevent excess use of limited resources, and the system achieved state-of-art performance levels. In [4] a Kohonen Artificial Neural Network was implemented to extract text, and the paper also discussed about the various real-world applications. [2] implemented a DCNN along with an SVM to combine the outputs. It received an accuracy rate of 72.1% (median of 10 trials) using the Tobacco 3842 dataset.

Recently, more focus has been given to Scene Text Localisation, which refers to the detection of text in images and videos of natural scenes. Despite the various challenges involved, such as interference of background with the text, and variations in fonts, sizes and colours, several systems have come up with unique solutions. [6] proposed a system of scene text recognition using information about detected text regions and implemented it into an application. It introduced a novel feature of 'Stroke Configuration Map'. [8] also introduced an innovative feature called 'Stroke Support Pixels'. This follows the observation that text consists of strokes, which can be further segregated into regions consisting of a part of a character, a whole character or multiple characters. The system achieved a recall of 72.4%, precision of 81.8%, and f-measure of 77.1% in text localization as per the ICDAR competition evaluation scheme. Similarly, [7] proposed a stroke detector which was 4 times faster and produced 2 times lesser region segmentation. It also offered a wide variety of script support (Chinese, Hebrew etc.).

OCR finds itself several real-world application. In [10], OpenCV and Tesseract libraries of Python were used to detect the license plate of vehicles. Similarly, [9] performed both word extraction and matching with high performance despite the presence of noise and degradation in the image. This was achieved through rigorous pre-processing and contouring followed by indexing of characters into a database, which facilitated image retrieval using keywords.

The importance of extracting and utilising text has been previously studied in [1] for identifying the information present in Identity Documents. Rather than adopting a position-based approach, the system made use of semantic analysis to infer the semantics between extracted sentences. Relations were made using the overall distance between text fields. Overall, the system was found to be easily scalable and highly efficient.

An alternative approach towards OCR was the subject of research in [12] that combined both focus and sharpness measures of images and made use of a Support Vector Machine to classify the images into three different classes ('very good', 'unreadable' and 'normal'). The proposed method was found to be both simple and efficient, therefore optimal to run on mobile devices. Results indicated that the system was successful in reducing the pre-processing computation cost.

The aim of [11] was to test the accuracy of state-of-art techniques (such as image processing using OpenCV and Tesseract OCR modules) on extracting text from screenshots of smartphones. The analysis was performed on a set of 13,000 and odd images. It was able to determine that employing image pre-processing along with the OCR engine increased character level accuracy from 66% to 74.8%. It concluded that factors such as irregular fonts, blending of text and background colours, a mix of icons and text etc. were responsible for reducing the rate of accuracy.

Finally, in [5] document image analysis was used to build a Human-Document Interaction (HDI) system using an augmented interface and addresses the need for further improvement in the field of HDI.

3. PROPOSED SOLUTION

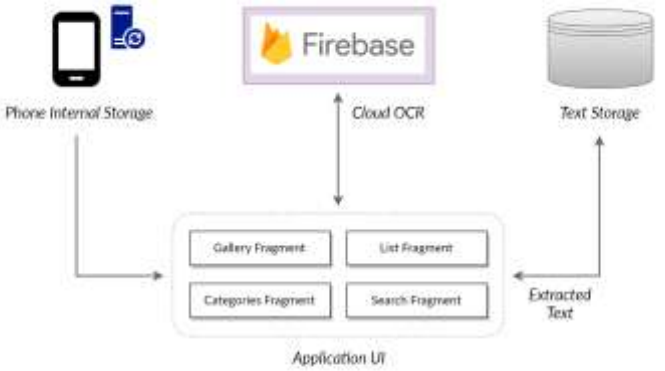
Before introducing the improved app design, it is crucial to first analyse what is lacking in similar existing applications. This will help us fully distinguish the unique features of the proposed work.

Mentioned below are some of the key observations made that heavily influenced the design of the app.

- Generally, when apps incorporate OCR, either the picture has to be selected individually from the gallery or snapped directly taken from the camera. There is a lack of automation on this front
- Most gallery apps are yet to implement an algorithm that separates document images from other image types
- Most apps do not use the extracted text in a meaningful manner, and generally just copy the text to the clipboard

3.1 Architecture

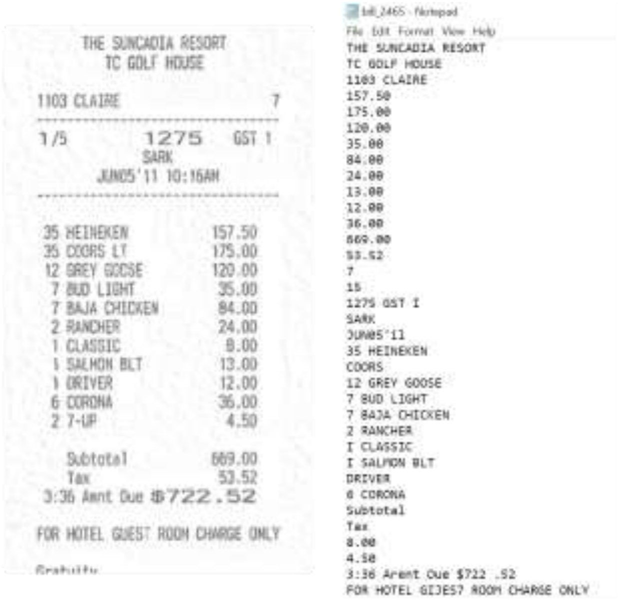
Fig. 1. depicts the system architecture of the proposed Android application. The app is responsible for connecting all the individual modules together. It passes the image to the Firebase Text Detection Engine in the cloud, retrieves the information and is also responsible for storing it into .txt files.



Fig—1: Architecture of Proposed System

3.2 Modules

1) Phone Internal Storage: The internal storage of the phone is the dynamic source of all images. It is subjected to image detection first, and only those which are identified to be document images have their contents extracted. The app aggregates both this information and collectively displays it in the UI.



```
THE SUNCADIA RESORT
TC GOLF HOUSE

1103 CLAIRE                    7
--------------------------------
1/5        1275       GST 1
             SARK
      JUN05'11 10:16AM
--------------------------------

35 HEINEKEN          157.50
35 COORS LT          175.00
12 GREY GOOSE        120.00
 7 BUD LIGHT          35.00
 7 BAJA CHICKEN       84.00
 2 RANCHER            24.00
 1 CLASSIC             8.00
 1 SALMON BLT         13.00
 1 DRIVER             12.00
 6 CORONA             36.00
 2 7-UP                4.50

  Subtotal           669.00
  Tax                 53.52
3:36 Amnt Due $722.52

FOR HOTEL GUEST ROOM CHARGE ONLY

Gratuity
```

```
bill_2465 - Notepad
File Edit Format View Help
THE SUNCADIA RESORT
TC GOLF HOUSE
1103 CLAIRE
157.50
175.00
120.00
35.00
84.00
24.00
13.00
12.00
36.00
669.00
53.52
7
15
1275 GST I
SARK
JUN05'11
35 HEINEKEN
COORS
12 GREY GOOSE
7 BUD LIGHT
7 BAJA CHICKEN
2 RANCHER
I CLASSIC
I SALMON BLT
DRIVER
6 CORONA
Subtotal
Tax
8.00
4.50
3:36 Arent Oue $722 .52
FOR HOTEL GIJES7 ROOM CHARGE ONLY
```

Fig—2: A document image of a bill (left) and its corresponding text contents extracted using Firebase and stored in a .txt file (right)

2) Firebase Text Recognition Platform: The text extraction is implemented using Google Firebase cloud Document Text Recognition platform. There are three major advantages of using Firebase:

- You don't require prior knowledge of Machine Learning in order to implement ML-based features
- Firebase Engine can be accessed from the cloud, hence the app avoids exceeding space constraints of the internal storage (unlike OCR SDKs)
- The entire platform is already optimised for Android development and doesn't need additional training data set in order to function

3) Text Storage: As and when Firebase extracts text from an image and returns it to the app, it gets stored into a .txt file having the same name as the image. This ensures that the information can be used for later purposes. Additionally, the app also stores all the existing file names in a string. This ensures that the same image isn't subjected to text extraction more than once, thereby enhancing the overall efficiency.

4) Application UI: The application User Interface is the module with which the user interacts. Each key feature of the app has been translated into its own dedicated 'fragment'. There are four key fragments.

a) Gallery Fragment: Used to display the images in a GridView.

b) List Fragment: Used to display the images in a ListView along with information related to it, such as Date Modified, and file size. A RelativeLayout is used to design each individual list item.

c) Categories Fragment: This fragment implements the proposed Bag-of-Words model for achieving Content-based Categorisation of Images.

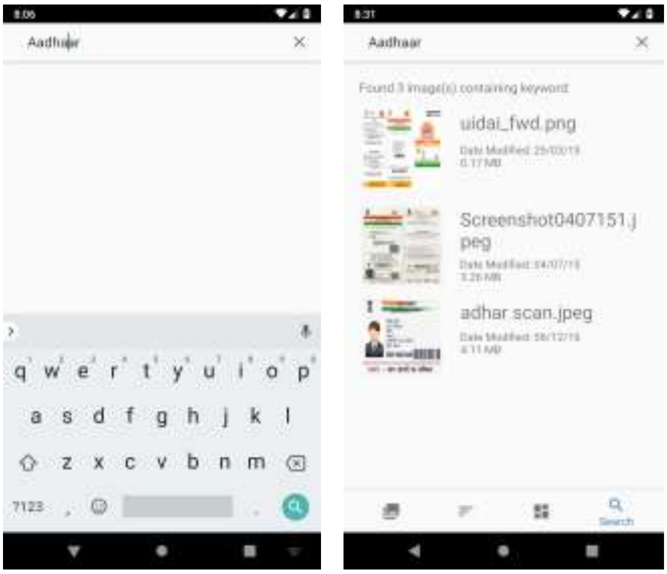
d) Search Fragment: This fragment allows the user to search for a certain keyword, based on which all the relevant images containing that keyword are displayed.

4. IMPLEMENTATION

4.1.1 Keyword Search Algorithm

The proposed app offers a keyword search feature. Since the text contents are extracted, string matching can be performed. This feature helps enhance usability and reduces the time spent by the user in manually searching for documents. The input will be the string that the user wants to search for and the output will be all the related document images containing that particular keyword.

Formally, given *m* text files, each consisting of *n* characters and a substring of length *k*, the goal of the algorithm is to find those files in which all *k* characters appear at least once.



Fig—3: Implementation of Keyword Search feature

A general algorithm can be given as follows:

Keyword Search Algorithm

```
Input: string accepted from the user
Output: set of text files containing the input string
for i = 1 to m do
    construct prefix table
    perform KMP
    if (keyword is present in text file) then
        select the corresponding image and display it
    end if
end for
```

A naive string searching algorithm has a complexity of *O(kmn).* However, by using a KMP Algorithm, the complexity is reduced to just *O(mn).* Space complexity is *O(km).* Given the scope and development scale of the app, KMP was chosen. However, for more advanced systems, hash tables with indexing can be used instead to increase efficiency further.

4.1.2 Proposed Content-based Categorisation

Content-based Categorisation for Document Images basically refers to the set of techniques which can be used to classify the document based on its text contents. In this proposed system, the classification will be based on the Bag-of-Words model.

The Bag-of-Words model refers to the vector representation of text, particularly used in Natural Language Processing (NLP). The vector consists of only the distinct words, disregarding both grammar and order of words in the sentences. For example, the sentences 'This is a cat. That is a dog' will have a corresponding vector as:

BoW[ ] = {"This", "is", "a", "cat", "That", "dog" }

The individual word frequencies of both the sentences are measured as:

sentence1[ ] = {1, 1, 1, 1, 0, 0}

sentence2[ ] = {0, 1, 1, 0, 1, 1}

The frequency of occurrence of specific words can act as a basis of classification.

Considering the specific use case of identifying bills and receipts. Any bill or receipt document will necessarily contain the following words:

billDoc[ ] = {"receipt", "bill", "tax", "total", "gst", "$"}

An identity document, on the other hand, will have the following:
idDoc[ ] = {"name", "id", "number", "DOB", "address"}

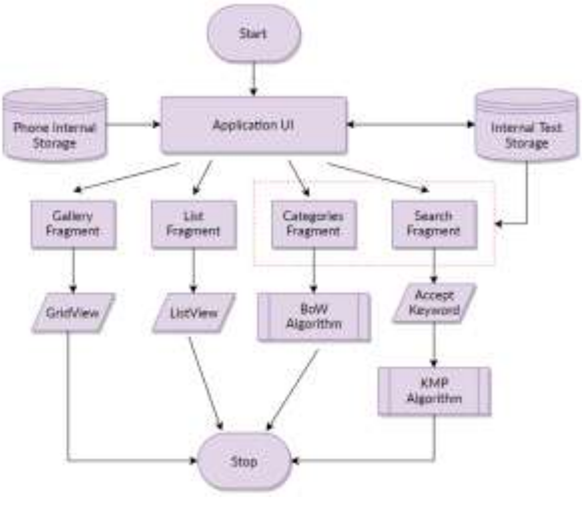
Consider the document shown in Fig. 2. Here, the words 'tax', 'total', 'GST' are all present. The vector representation of this document corresponding to billDoc will be {0, 0, 1, 1, 1, 1} and corresponding to idDoc will be {0, 0, 0, 0, 0}. Hence we infer that since the vector representation matches more with a bill rather than identity document, the image is a bill/receipt document image.

The proposed model is rather basic when compared to other approaches such as the use of Neural Networks. However, mobile systems are not capable of performing extremely complex computations. Hence, the Bag-of-Words model will be much more efficient than other specialized classifiers or CNNs.

4.1.3 Process Flow

Fig. 4. illustrates the flow of control of individual processes and transfer of data within the app. So firstly, we have two separate input sources for the UI; the internal phone storage and the extracted text storage. The contents of the internal phone storage and text storage are made available to the individual fragments through the *MainActivity.java*. While the Gallery and List fragments need only the images to function, the Search and Categories fragments additionally require the text storage to execute the algorithms.

To briefly describe the entire working of the proposed app, the *MainActivity.java* automatically scans the internal storage. It segregates the document images through image detection techniques and further performs OCR through Firebase. The extracted text is stored in text files. The app synchronises with the storage dynamically at run time. The categories fragment allows the user to see the various types of documents present in his mobile phone (for example, Bills, Receipts, Identity Documents etc.). The app meticulously keeps a track of all the images whose text contents have already been extracted. This directly impacts the performance and efficiency of the app.



Fig—4: Process Flow Diagram for the app

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

After successfully implementing the app, experimental results showed that the overall performance was satisfactory. The app design was made to be intuitive, minimalistic, yet functional. The system was tested for 50 document images and was successfully able to perform OCR and keyword search.

During the development and testing period, a few glitches and challenges were encountered, and are mentioned below.

***1) Unpredictable accuracy of Firebase Engine**:* This was the only drawback of implementing Google's Firebase for the text detection process. There were several factors found to have affected the accuracy of the OCR Engine such as:

- Size of the text
- Font used
- Colour of text with respect to the background
- Any tilt in the text

This was solely responsible for hindering the performance and accuracy of the application.

***2) Inefficient Storage Techniques**:* The app employs text file storage techniques to hold the extracted information. However, this is not sufficient for an app that may contain thousands of images. The access time increases and overall speed decreases considerably as the library size increases. This issue can be solved by implementing a hash table instead, as it has an access time of just *O(1).*

***3) Lack of support for external storage**:* The app needs to incorporate external storage (such as SD card) as well to be fully functional. The implementation of this has been omitted due to lack of time. The second challenge was the storage of information. While a .txt file storage system is practical for small-scale applications, there will definitely be a setback in the time taken to retrieve information. Even databases are notably inefficient in sub-string search. However, the use of hash-indexed table containing all commonly searched keywords and results might be able to make the storage more efficient.

6. CONCLUSION AND FUTURE WORK

This paper elaborated on the design and construction of a comprehensive system for Document Image retrieval and searching based on the mobile platform. The main advantages of this system were that it facilitated features such as keyword search and content categorisation of images, and also helped automate the entire organisation process for the user, while simultaneously occupying the least amount of space.

The system employed the KMP algorithm for keyword search, and proposed using a Bag-of-Words model for categorisation. The platform used for text detection is Firebase from Google's ML Kit. Experimental results showed that the overall performance of the app was satisfactory. There were a few challenges encountered during the development of this project. The first was the unpredictable accuracy of Firebase OCR. It was also observed that the use of a hash table instead of text file storage system could further optimise performance and information retrieval.

The limited availability of time during the period of app development was solely responsible for limiting the scope of the proposed system. However, future work will look into improving this system, particularly in terms of efficiency, and by adding several other useful features, such as integration with cloud storage, and even allowing cross-platform access of system. Additionally, a concentrated study has to be made comparing the various state-of-art OCR Engines available for the Android platform, and the proposed system should implement the most accurate one. Lastly, the proposed system can even be extended to include other types of images, and hence a complete solution can be developed and deployed.

REFERENCES

[1] Gutiérrez, José C., Rodolfo Valiente, Marcelo T. Sadaike, Daniel F. Soriano, Graça Bressan, and Wilson V. Ruggiero. "Mechanism for Structuring the Data from a Generic Identity Document Image using Semantic Analysis." In Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web, pp. 213-216. ACM, 2017.

[2] Roy, Saikat, Arindam Das, and Ujjwal Bhattacharya. "Generalized stacking of layerwise-trained deep convolutional neural networks for document image classification." In Pattern Recognition (ICPR), 2016 23rd International Conference on, pp. 1273-1278. IEEE, 2016.

[3] Kang, L., Kumar, J., Ye, P., Li, Y., & Doermann, D. (2014, August). Convolutional neural networks for document image classification. In Pattern Recognition (ICPR), 2014 22nd International Conference on (pp. 3168-3172). IEEE.

[4] Manwatkar, P. M., & Yadav, S. H. (2015, March). Text recognition from images. In Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on (pp. 1-6). IEEE.

[5] Karatzas, D., DAndecy, V. P., Rusinol, M., Chica, A., & Vazquez, P. P. (2016, April). Human-Document Interaction Systems--A New Frontier for Document Image Analysis. In Document Analysis Systems (DAS), 2016 12th IAPR Workshop on (pp. 369-374). IEEE.

[6] Yi, Chucai, and Yingli Tian. "Scene text recognition in mobile applications by character descriptor and structure configuration." IEEE transactions on image processing 23, no. 7 (2014): 2972-2982.

[7] Busta, Michal, Lukas Neumann, and Jiri Matas. "Fastext: Efficient unconstrained scene text detector."

Proceedings of the IEEE International Conference on Computer Vision. 2015.

[8] Neumann, Lukáš, and Jiří Matas. "Efficient scene text localization and recognition with local character refinement." Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015.

[9] Yadav, Seema, Priya Bhanushali, Saurabhkumar Jain, and Tejinder Kaur. "Word matching and retrieval from images." In Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of, vol. 1, pp. 318-323. IEEE, 2017.

[10] Palekar, R. R., Parab, S. U., Parikh, D. P., & Kamble, V. N. (2017, April). Real time license plate detection using openCV and tesseract. In Communication and Signal Processing (ICCSP), 2017 International Conference on (pp. 2111-2115). IEEE.

[11] Chiatti, A., Yang, X., Brinberg, M., Cho, M.J., Gagneja, A., Ram, N., Reeves, B. and Giles, C.L., 2017, December. Text extraction from smartphone screenshots to archive in situ media behavior. In Proceedings of the Knowledge Capture Conference (p. 40). ACM.

[12] Bui, Quang Anh, David Molard, and Salvatore Tabbone. "Predicting mobile-captured document images sharpness quality." 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). IEEE, 2018.