Implementation of a Web Based Text Extraction Tool using Open Source Object Models

S Rajeswari *M Saibaba*

Abstract

The library in our institute is the repository of all reports and design documents. In our library, reports belonging to the past three decades are preserved. The present day scans, reports are all searchable, but the scan reports that are two decades old are not searchable. They are very important and constantly referred by Scientists and Engineers for the fast breeder design purpose. It was decided all these documents would be scanned and added to the collection. When scanning was done, it was found a skew getting introduced into these documents and OCRs were not directly de-skewing and required a pre-processing to be applied to the document. Optical character recognition (OCR) tool is a solution to extract the text from image and scanned documents. So it was decided an OCR would be developed using open source libraries and the de-skewing methods necessary would be added to make it useful to our task.

This paper discusses what is an OCR, the different steps necessary to extract text from image files using an OCR, the OCR tool development, evaluation of the OCR tool and the de-skewing method implemented in the tool.

Keywords: De-skewing, Image Files, OCR

1. Introduction

Optical Character Recognition, or OCR, is a technology that converts different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data. Early versions needed to be trained with images of each character, and worked on one font at a time. In this project, open source tools were used in the implementation of the OCR tool. This decision was taken since OCRs require a large collection of pixel library of different fonts along with their attributes like 'italics, bold, capital, size variation, etc.' Using an existing pixel library reduces the effort to be put into the development of an OCR and more efforts can be put into the development of pre-processing methods and fine tuning the OCR.

2. OCR – Steps to extract text from an image file

OCR extracts text from an image file. Image files are all binary files with only pixels present. OCR has to segment the file into pages which is the first step in segmentation. Before segmentation, most of the OCR does binarization.[1] Binarization can be broadly categorized as global [2] and local[3]. In this process, using algorithms the entire file is converted into 1s and 0s. Any colour or gray scale found in the file is removed. Converting the entire file into 1 or 0 based

on single threshold is called Otsu binarization [4]. Adaptive thresholding [5] and its variants are used to do localised thresholding.[6]

Page analysis is the next major routine used in the character recognition process. It is finding page elements like blocks of text, tables, lines, single characters [7]. Line segmentation consists of slicing a page of text into its different lines. This step also analyzes interline spacing, line skew and separates touching lines. The word segmentation isolates one word from another. It uses interword space of segmenting words [8]. The character segmentation separates the various letters of a word. If the characters have the same width (fixed pitch), character segmentation is easy. The problem is more when the width of the letters varies (proportional pitch) [9]. The actual character recognition extracts characteristics out of each isolated shape and assigns a symbol.

The pixels of a scanned image has to be organised into characters. To make this a library of characters in pixel form is necessary. This library should support different fonts, styles and sizes. At this point either a feature extraction routine or matrix matching routine compares the characters in the scanned image file to the characters in the library.

In open source object models, two such libraries were available. One was from Microsoft called the Microsoft Object Document Imaging (MODI) [10]. This library was available in C#. The other open source library was from Google called Tesseract. This library was available in C++. In this project, to develop the OCR tool, the MODI object model is used.

3. Development of in house OCR Tool

In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building software and applications. The MODI object model makes it possible to develop custom applications for managing document images (such as scanned and faxed documents) and the recognizable text that they contain. The complete overview of the library and details are available at [11].

The MODI object model was found to contain the complete pixel libraries for all the fonts supported by Microsoft Office. In addition the algorithms for the complete OCR process were also available. After the MODI OCR tool was developed and implemented in the intranet, few documents were tested to assess the accuracy of the tool. The documents and the test accuracy are recorded.

4. Evaluation Methodology:

It was decided to evaluate the trial versions commercial OCRs ABBY Fine Reader OCR [12], Acrobat Reader Pro OCR and open source Tesseract OCR [13] along with MODI OCR developed to test the accuracy in reading the documents. OCR tools generally have difficulty in checking if text is given in multiple columns, the text is colored, if the fonts are of different sizes, if the document is blurred or the document is an invoice.

To test the different abilities of OCR in segmenting, recognizing different size fonts, different color fonts, invoices, text embedded in images, distorted document, blur and bevel introduced to a document a test set of documents were prepared. They are available in Figure 1, Figure 2, Figure 3 and their reading accuracy decided by the number of words recognized correctly and tabulated in Table-1.

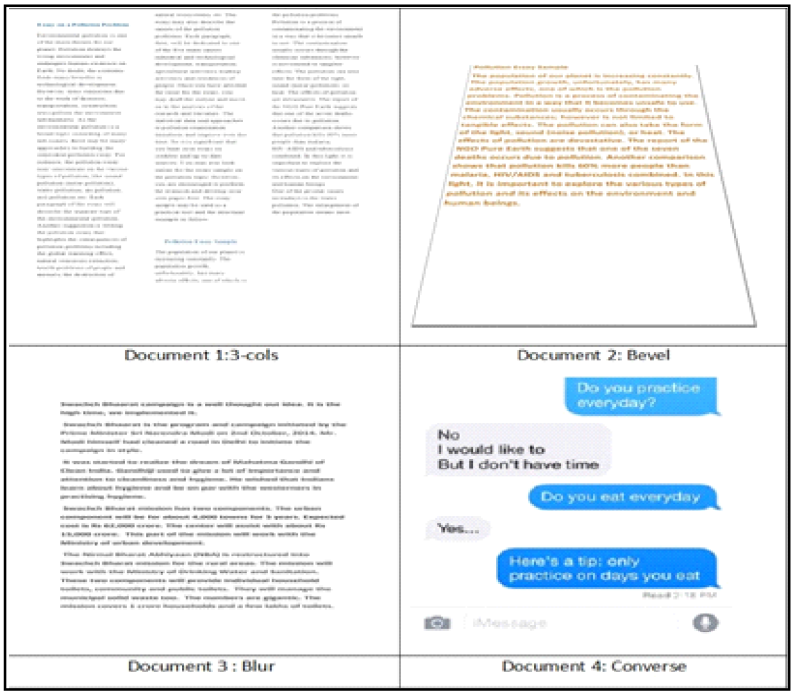

Figure 1

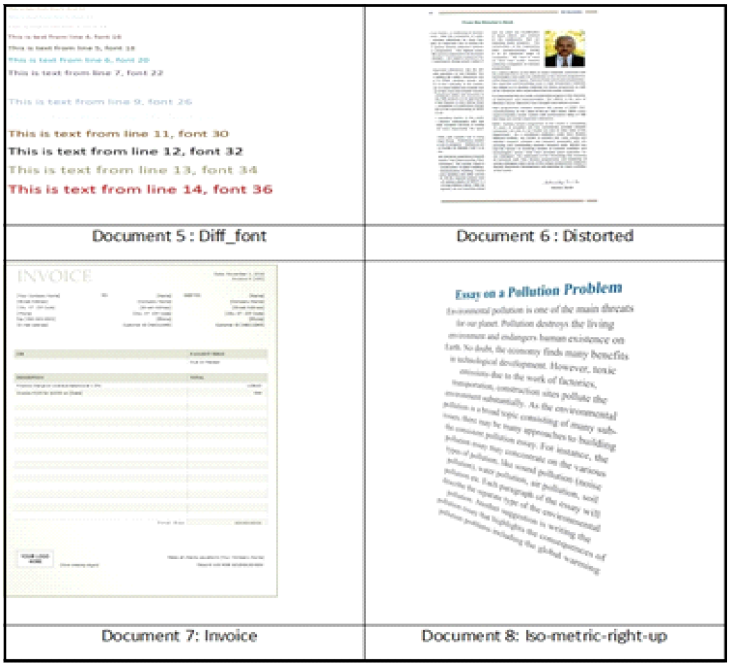

Figure 2

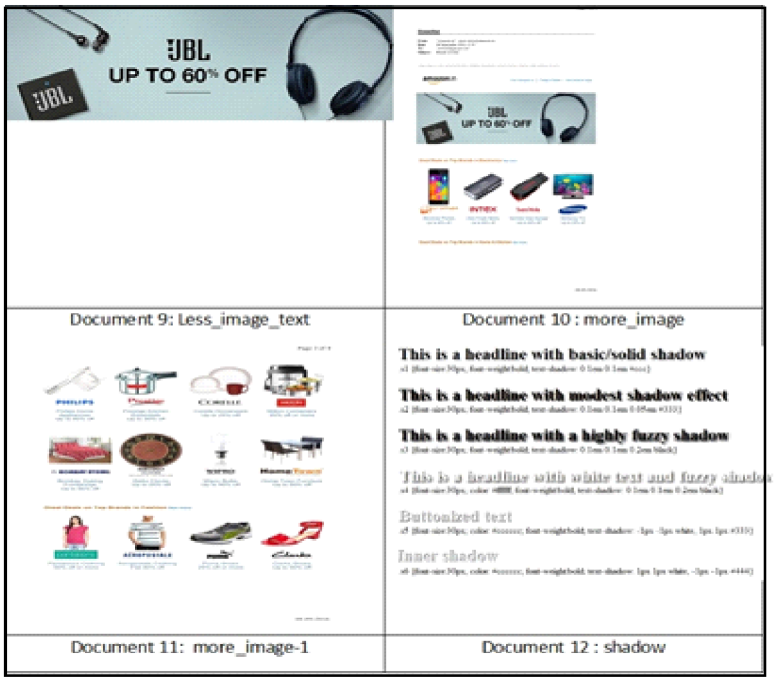

Figure 3

Table 1

SNo	Documents Name	Accuracy in Percentage			
		MODI	AcrobatPr	Abby	Tesser
1	3-cols	97.22	98.15	99.07	98.61
2	Bevel	98.50	97.76	100	83.58
3	Blur	85.14	0	83.16	81.18
4	Converse	74.28	91.42	85.71	82.85
5	Diff_font	100	96.39	100	100
6	Distorted	87.14	97.58	95.34	82.37
7	Invoice	82.79	89.24	100	83.87
8	Iso-metric-right-up	26.31	0	33.83	27.81
9	Less_image_text	100	100	100	100
10	More-image	93.33	93.33	98.09	88.57
11	More-image-1	90.67	89.83	93.22	94.06
12	Shadow	27.64	49.59	78.04	27.06

From the documents 1 to 12 found in Figures 1 to 3 and the table 1 clearly shows MODI OCR is able to read documents like invoice, mail screens, advertisements on the net, text combined with images accurately and is as good as trial versions of commercial OCRs.

5. Skew correction algorithms implemented

The older books and journal back volumes which are damaged and digital copy not available were scanned at the library. Such documents had some skew introduced in them when scanning. So, to correct the skew angle before using the OCR, a skew correction algorithm was necessary. The algorithms studied were Projection Profile algorithm, Hough Transform algorithm, Radon Transform, Fourier Transform and many variants [14]. In our tool, the Hough transform was used. The pseudo code used in development of the Hough deskewing is given below:

Read image and convert it into binary image.

Find edges.

Do the Hough transform to detect text lines of the image.

for each black pixel in a binary image

for theta (-angle to +angle) do

$\rho = x * \cos(\Theta) + y * \sin(\Theta)$

Find peaks over Hough transform.

Find and plot the lines.

Find theta Θ between lines and the x axis.

Compute accumulator array for theta

Skew angle = maximum (value on accumulator array).

It is computation intensive, accurate and simple.The skew introduced were in the range of 1 to 20 degrees and this is corrected efficiently by Hough transform and the algorithm is explained clearly in reference[15]. Refer to Figures 4 & 5 for skew , deskew using the tool.

Figure 4

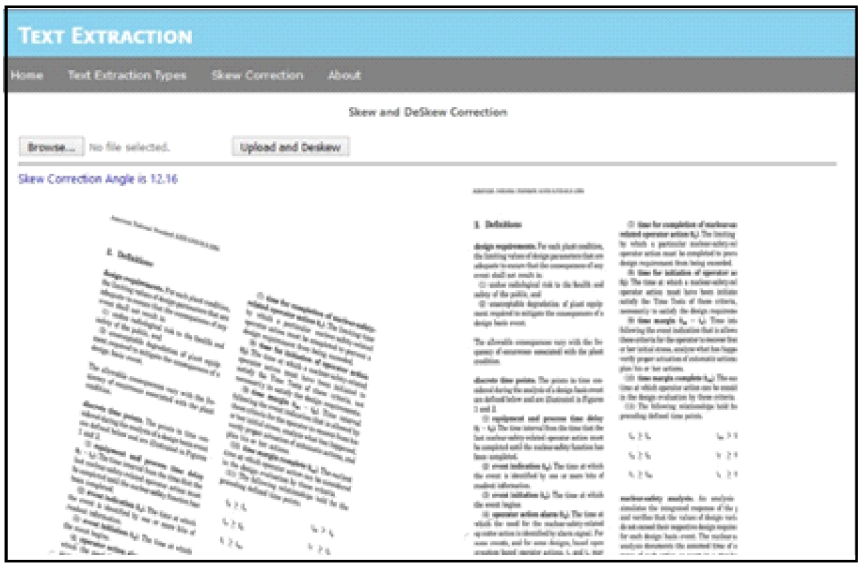

Figure 5

6. The architecture of the Tool

The text extraction tool was developed using Windows Server/ IIS web server/ dot NET technology. The screen shot of the tool is shown in Figure 6. In the figure, the text extracted from an image file is displayed.

Figure 6

7. Conclusion

It was found, the OCR tool developed in house using open source object model MODI gives more than 95% accuracy of documents that does not have any noise and not distorted due to age. The skewing tool de-skews all documents in the range from 0 to 30 degrees perfectly since this is the skew introduced during scanning. The text extraction tool implemented is very useful for scanned and image documents.

References

1. Sezgin M, Sankur B. (2004), Survey over image thresholding techniques and quantitative performance evaluation, (pp. 146-168). Journal of Electronic Imaging 13 (1).
2. Gonzalez, Rafael c. & Woods, Richard e. (2002). Thresholding. In Digital Image Processing, (pp. 595–611). Pearson Education ISBN 81-7808-629-8
3. Gllavata R, Ewerth, Freisleben B. (2003). Finding Text in Images via Local Thresholding, (pp. 539-542). Proceedings of IEEE Symposium on Signal Processing and Information Tech-nology, Siegen, 14-17 December 2003,
4. Otsu, (1979). A thresholding selection method from gray-scale histogram, (pp. 62-69). IEEE Transactions on System, Man, and Cybernetics.
5. Sauvola T, Seppanen T, Haapakoski S. Pietikanen M, (1997). Adaptive document binarization, (pp 147-152), Fourth International Conference Document Analysis and Recognition (ICDAR), Ulm, Germany, August 1997.
6. Nikolaou, Badekas E, Papamarkos N, and Strouthopoulos C, (2006), Text localization in color documents, (pp 181-188), International Conference on Computer Vision Theory and Applications, Setu´bal, Portugal, 2006.
7. Kaur Sukhvir , P.S.Mann , Khurana Shivani, (2013), Page Segmentation in OCR System- A Review, International Journal of Computer Science and Information Technologies, Vol. 4 (3) , 2013, 420-422- -ISSN-0975-9646
8. Fujisawa, Nankano Yasuaki, And Kurino Kiyomichi, (1992), Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis, Proceedings of The IEEE. Vol. 80. No. 7. July 1992
9. Zuva Tranos, Olugbara Oludayo O., Ojo Sunday O and Ngwira Seleman M. (2011), Image segmentation, Available Techniques, Developments and Open Issues, Canadian Journal on Image Processing and Computer Vision Vol. 2, No. 3, March 2011.
10. Microsoft, Microsoft Office Document Imaging Object Model, Available at web site URL https://msdn.microsoft.com/en-us/library/office/aa203311 (v=office.11).aspx (Accessed on 23/12/2016).
11. Wikipedia, Microsoft office shared tools, Available at web site URL https://en.wikipedia.org/wiki/Microsoft_Office_shared_tools (Accessed on 20/02/2017)
12. Abby, Abbyy Finerreader, Available at web site URL http://www.abbyy.com/en-apac/ (Accessed on 15/10/2016)

13. Google, Tessaract OCR, Available at web site URL http://code.google.com/p/tesseract-ocr/ (Accessed on 12/06/2016)

14. H.S Baird, (1987) The Skew Angle of Printed Documents, Proceedings of Conference Society of Photographic Scientists and Engineers, Rocherster, New York, 1987, pp 14-21.

15. V N Manjunath Aradhya, G Hemantha Kumar, and P Shivakumara (2007)- Skew Detection Technique for Binary Document Images based on Hough Transform - International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:8,.

About Authors

S Rajeswari, Head, Scientific Information Resource Division, IGCAR
Email: raj@igcar.gov.in

Mr. M Saibaba, Group Director, Resource Management Group, IGCAR