An Accuracy Examination of OCR Tools

Jayesh Majumdar, Richa Gupta

Abstract*—In this research paper, the authors have aimed to do a comparative study of optical character recognition using different open source OCR tools. Optical character recognition (OCR) method has been used in extracting the text from images. OCR has various applications which include extracting text from any document or image or involves just for reading and processing the text available in digital form. The accuracy of OCR can be dependent on text segmentation and pre-processing algorithms. Sometimes it is difficult to retrieve text from the image because of different size, style, orientation, a complex background of image etc. From vehicle number plate the authors tried to extract vehicle number by using various OCR tools like Tesseract, GOCR, Ocrad and Tensor flow. The authors in this research paper have tried to diagnose the best possible method for optical character recognition and have provided with a comparative analysis of their accuracy.

***Keywords*— OCR tools; Orcad; GOCR; Tensorflow; Tesseract;**

I. INTRODUCTION

Optical character recognition is a method with which text in images of handwritten documents, scripts, passport documents, invoices, vehicle number plate, bank statements, computerized receipts, business cards, mail, printouts of static-data, any appropriate documentation or any computerized receipts, business cards, mail, printouts of static-data, any appropriate documentation or any picture with text in it gets processed and the text in the picture is extracted. This is very helpful in assessing various documents and pictures, reading vehicle number plates and various tags and remarks that can be used in identification [1]. It can be used to regenerate text from old documents and manuscripts which can degenerate in the printed form but can be preserved forever and made multiple copies when converted into the textual format by using OCR. Optical character recognition is basically the analysis of the scanned or captured image and then translating these character modules into character codes from where these codes can be used to store the text, edit the text, search and store the text more efficiently which can be used in the machine processes [1].

With OCR, number plates can be recognized automatically. It is used for data entry for various business documents exemplarily passport, cheque, bank statement, invoice etc. It is used in airports for recognition and extracting information from passports [2]. With its use, the information extracted from business cards is used in the contact list. Its other usages include scanning of books i.e. convert printed documents into texts, pen computing, developing technologies for assisting the visually impaired, making electronic images searchable of hard copies, defeating or evaluating the robustness of CAPTCHA.

Revised Manuscript Received on July 10, 2019.

Jayesh Majumdar, Jaypee Institute of Information Technology, Noida, India.
Richa Gupta, Jaypee Institute of Information Technology, Noida, India.

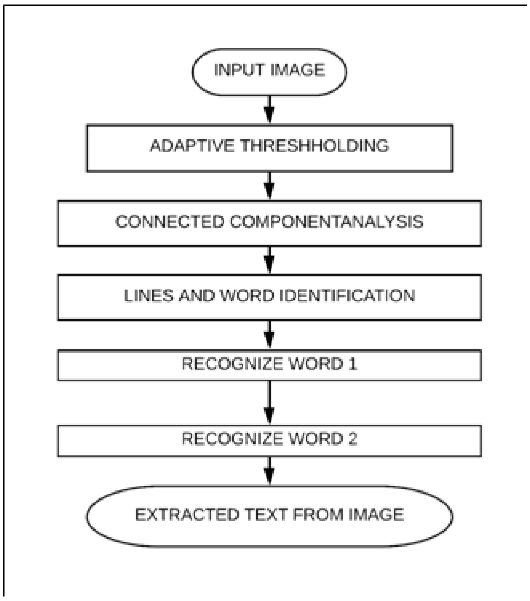

INPUT IMAGE
ADAPTIVE THRESHHOLDING
CONNECTED COMPONENTANALYSIS
LINES AND WORD IDENTIFICATION
RECOGNIZE WORD 1
RECOGNIZE WORD 2
EXTRACTED TEXT FROM IMAGE

Fig.1: Functioning of OCR [2]

II. OCR PROCDURE AND PROCESSING

To improve the probability of successful processing of an image, the input image is often 'pre-processed'; it may be de-skewed or despeckled. The functioning of the OCR involves the steps shown in fig 1. Adaptive thresholding and connected component analysis involves binarization of image that includes conversion of image into greyscale and Lines and Word Identification involves marking of the character outlines. Recognize word 1 and Recognize word 2 are the two phases in which recognition takes place in first phase the word is partially recognized whereas the second phase recognizes words more specifically[1][2].

The techniques can be explained as:

Despeckle: Removal of spots, granular noise or smoothing the edges to improve the image quality. After scanning, the document may not be aligned properly. Therefore, it is tilted anticlockwise or clockwise as per the misalignment of the document to make lines vertical or horizontal appropriately [2][3].

Binarization: Converting an image to a binary format i.e. conversion of the normal image to gray scale consisting of only two colors (black and white). Binarization is performed simply by bifurcating the specific text or image component from the background.

Binarization plays a crucial role in image processing as many of the softwares or recognition algorithms used commercially operate on binary images only as it is easier to use. Additionally, binarization increases the quality of character recognition via OCR [3].

Line removal and zoning: Removes unnecessary boxes and lines and zoning identify independent paragraphs and columns.

Script recognition: Identifies the script of the text. It is necessary as the script might have changed in the process.

Segmentation: Due to picture confection some words or characters are identified together or a word is separated into its constituent characters. Segmentation is isolation of such words and characters [1].

III. OCR TOOLS

OCR tools are soft wares and programs that are open for the public to use and extract text from images and work on it. Images that are scanned and have text as major part are processed and converted easily by these tools with better accuracy. On few occasions, it is very difficult for a tool to recognize text because of the background, extra brightness, dark shadows, font size, font type and many other irregularities that occur while capturing or scanning an image [2][4].

Input image format can be JPG, PNG, BMP, GIF, TIFF and PPM depending upon the OCR tools used.In this paper, the authors have compared Tesseract, GOCR, OCRAD and Tensor flow[5][6].

A. Tesseract

Google sponsored the development of this OCR engine in 2006 and released it under APACHE License. Tesseract executes from the command line Interface. Previously Tesseract can be used only for the .tiff file formats but now with the use of Leptoncia library, monospacing and proportional spacing can be easily detected by Tesseract[2][4].

Initially,tesseract could recognize English language and six other western languages but over time it was developed for many languages now tesseract can be trained for any language. It can also process text from right to left like in Arabic and Hebrew.
If the input images are not pre-processed tesseract tends to give very poor output [2][4].

Architecture and Working

The architecture of tesseract involves the following steps:

Adaptive Thresholding: Binarization takes place and the image is converted into a grey scale.

Connected component analysis: Character outlines are extracted and converted into blobs which are further organized into text lines and region that are analysed for equivalent text size and fixed area.

Text recognition: In the first step, each word is passed through an adaptive classifier as training data and an attempt to recognize the word is made.

In the second step, various issues are resolved and final output text is obtained from the image.

Tesseract command includes two arguments, in the first argument there is input file name that has text and the second argument has output file in which the extracted text is present. File extension .txt is used by Tesseract [2].Working of Tesseract is illustrated in fig 2.

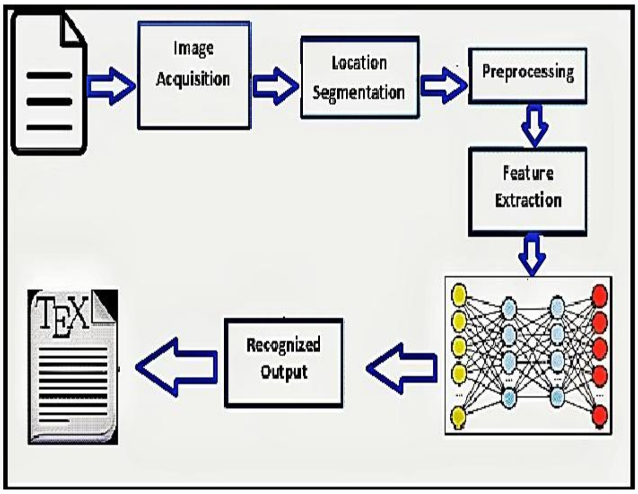


Fig.2: Working of Tesseract [2].

B. GOCR

GOCR or JOCR is a program for optical character recognition that converts scanned images, PCS and portable pixmap into text files. It uses gocr.tcl graphic interface and is processed as a single command line application. GOCR assures that it can treat single-column fonts which are 20–60 pixels in height [7]. It is not suited for cases which involve overlapping characters, cursive handwritten text, noisy images, heterogeneous fonts, largeangles of skew and text written in anything except the Latin alphabet.

Barcode translation can also be done using GOCR. It accepts various image file formats like ppm, pnb, pgm ,pbm, tga , pcx and png [2][7].

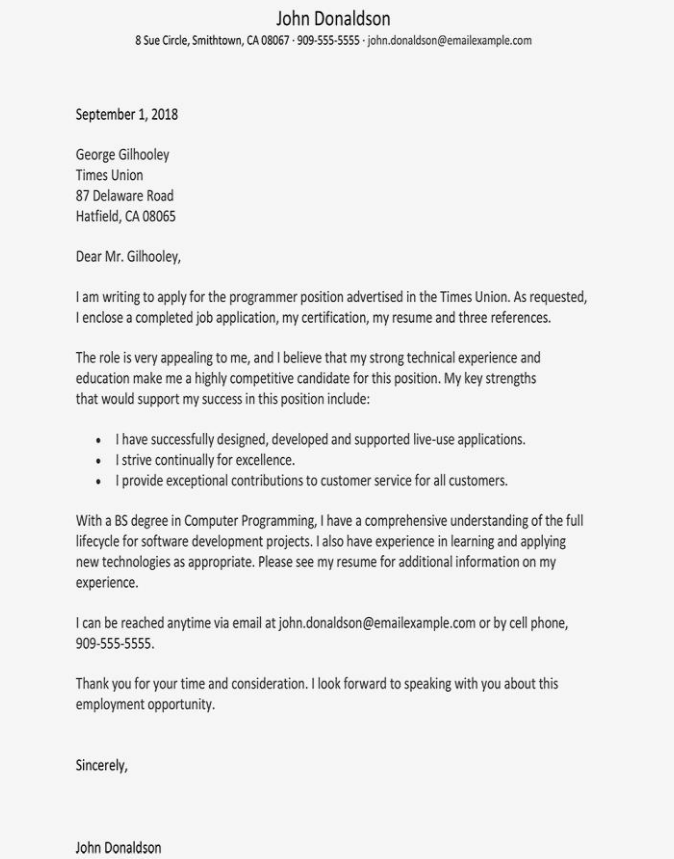
C. Ocrad

OCRAD is a program for optical character recognition which is a part of a GNU project. It has its license under GNU GPL. It rests on a feature extraction method, that processes pictures in portable pixmap formats known as Portable any map and produces text in byte (8-bit) or UTF-8 formats [8].

It also inculcates a layout analyser, which is able to separate the blocks or columns of text found in the printed state. Ocrad works as a stand-alone command-line application or as a back-end to other programs [8].

D. Tensor flow

Tensor flow is an open source symbolic maths library software which is used in the neural network and dataflow programming. The Tensor flow was developed under Google brain's second-generation system. Computation of Tensor flow is expressed via stateful dataflow of graphs [9]. TPU is a programmable AI accelerator designed to provide high throughput of low-precision arithmetic [10]. Tensor flow basically has two units first one is multi-dimensional arrays which are known as tensors they can be 1D vector, 2D or in form of matrix and the other is graphs which can break and save computations, facilitate distributed computation and help in visualizing many machine learning model [9][11].



John Donaldson
8 Sue Circle, Smithtown, CA 08067 · 909-555-5555 · john.donaldson@emailexample.com

September 1, 2018

George Gilhooley
Times Union
87 Delaware Road
Hatfield, CA 08065

Dear Mr. Gilhooley,

I am writing to apply for the programmer position advertised in the Times Union. As requested, I enclose a completed job application, my certification, my resume and three references.

The role is very appealing to me, and I believe that my strong technical experience and education make me a highly competitive candidate for this position. My key strengths that would support my success in this position include:

- I have successfully designed, developed and supported live-use applications.
- I strive continually for excellence.
- I provide exceptional contributions to customer service for all customers.

With a BS degree in Computer Programming, I have a comprehensive understanding of the full lifecycle for software development projects. I also have experience in learning and applying new technologies as appropriate. Please see my resume for additional information on my experience.

I can be reached anytime via email at john.donaldson@emailexample.com or by cell phone, 909-555-5555.

Thank you for your time and consideration. I look forward to speaking with you about this employment opportunity.

Sincerely,

John Donaldson

Fig.3: Input black and white text image

IV. ACCURACY MEASURE

To measure the accuracy of OCR done by respective tools, the authors have compared the output given by the tools with the text in the image. The original text in the image was saved in a text file manually from where it was compared to the output which was obtained after performing OCR. The comparison has been done on the basis of the two methods.

A. METHOD 1 – Character comparison:

In this method, the output was compared to the original text in the text file character by character. Each character of the output is compared by the respective character from the original text if it matches the count of matching character (MC) increases if it does not match the count of non-matching character (NMC) increases. To find the accuracy, the authors have divided the number of matching characters by a total number of characters.

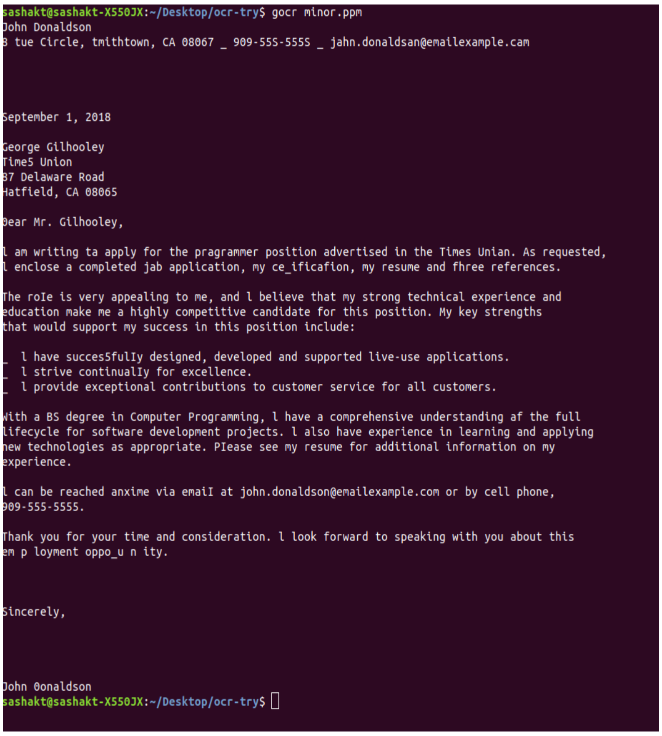
Accuracy percentage = (Number of matching characters/ Total characters) *100= {MC / (MC +NMC)}*100

Accuracy by this method is more as it compares each and every character.

B.METHOD 2- Word comparison:

In this method, the output was compared to the original text in the text file by word by word comparison. Each word in output is taken as a string and compared to the string provided with the original text and if the word matches (MW), the match count increases by 1 and if it doesn't, the count of the non-matching word (NMW) increases by one. To find the accuracy, the authors divide the matching word by total words.

Accuracy percentage = (Matching word/Total words)*100 = {MW / (MW +NMW)}*100



```
sashakt@sashakt-X550JX:~/Desktop/ocr-try$ gocr minor.ppm
John Donaldson
8 tue Circle, tmithtown, CA 08067 _ 909-555-5555 _ jahn.donaldsan@emailexample.cam



September 1, 2018

George Gilhooley
Time5 Union
87 Delaware Road
Hatfield, CA 08065

0ear Mr. Gilhooley,

l am writing ta apply for the pragrammer position advertised in the Times Unian. As requested,
l enclose a completed jab application, my ce_ificafion, my resume and fhree references.

The roIe is very appealing to me, and l believe that my strong technical experience and
education make me a highly competitive candidate for this position. My key strengths
that would support my success in this position include:

_  l have succes5fulIy designed, developed and supported live-use applications.
_  l strive continualIy for excellence.
_  l provide exceptional contributions to customer service for all customers.

With a BS degree in Computer Programming, l have a comprehensive understanding af the full
lifecycle for software development projects. l also have experience in learning and applying
new technologies as appropriate. PIease see my resume for additional information on my
experience.

l can be reached anxime via emaiI at john.donaldson@emailexample.com or by cell phone,
909-555-5555.

Thank you for your time and consideration. l look forward to speaking with you about this
em p loyment oppo_u n ity.



Sincerely,



John 0onaldson
sashakt@sashakt-X550JX:~/Desktop/ocr-try$ 
```

Fig.4: Output using GOCR when input image is text image

V. COMPARATIVE ANALYSIS

In this paper, accuracy percentage and error percentage have been used to do a comparative analysis of OCR tools. The number of total characters or words haven determined and used to calculate the accuracy based on the different output shown by different tools. Fig 3 shows the input of the image of text that was processed by OCR tools. Table 1 contains the output analysis of the Tesseract, GOCR, Ocrad and Tensor Flow.

It can be seen in Fig. 4 that the output is shown by GOCR when text is used as input shows very good accuracy compared to when vehicle number plate is given as the input image. When various images of vehicle number plate were given as input Tesseract gave most accurate output. Fig. 5 shows the input image which was processed by OCR tools. Table 2 contains the output analysis.

Table 1:- Accuracy and error when input image is text image

Tool used	Input image type	Number of words in input image	Number of words obtained after performing ocr	Number of characters not obtained after ocr	Accuracy percentage	Error percentage
Tesseract	BnW-text	190	183	7	96.31%	3.69%
Gocr	BnW-text	190	176	14	92.63%	7.37%
Ocrad	BnW-text	190	101	89	53.15%	46.85%
Tensor flow	BnW-text	190	168	22	88.42%	11.58%

Table 2 :- When input image is vehicle number plate.

Tool used	Input image type	Number of characters in input image	Number of characters obtained after performing ocr	Number of characters not obtained after ocr	Accuracy percentage	Error percentage
Tesseract	Colour-Vehicle number plate	11	10	1	90.95%	9.1%
Gocr	Colour-Vehicle number plate	11	6	5	54.54%	45.46%
Ocrad	Colour-Vehicle number plate	11	6	5	54.54%	45.46%
Tensor flow	Colour-Vehicle number plate	11	8	3	72.72%	27.28%

VI. Conclusion

When the accuracy of OCR is compared using input image as text image, the accuracy of all the tools is found to be better. Tesseract has the highest accuracy whereas GOCR also shows highly accurate output. The accuracy of Tensor flow depends on the size of the training dataset used and Ocrad shows decent results as well. When the input image provided is vehicle number plate which hasa combination of alphabetical characters and numeric values .

Tesseract shows outstanding results and proves to be most accurate whereas GOCR and OCRAD show decent results.

Tesseract in both the cases gives optimum results and proves very efficient in performing OCR.

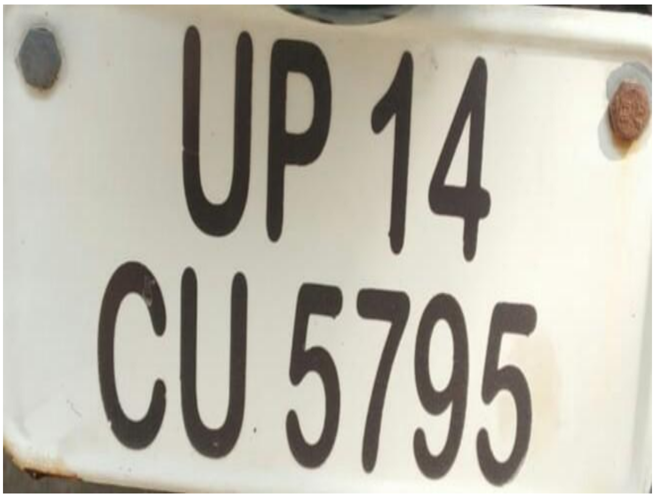


Fig.5 Input image of vehicle number plate

Though even Tesseract could not correctly identify the specials marks like -> and <>.

Error Analysis:-

Numbers of errors were encountered during this study, some of them are mentioned below:-

1. “5” is recognized as “8”.
2. “q” is recognized as “0”.
3. “I”,“L” is recognized as “1”.
4. “o” is recognized as “0”
5. “p” is recognized as “0”.
6. “d” is recognized as “c”, “1” or “0”.
7. “g” is recognized as “0”.
8. “!” is recognized as t.
9. “->” is recognized as “–” or “.”

10. Remarks of bolts are recognized as "." or ":"
11. The accuracy went low due to brightness difference because of shadow casted.
12. The background had to be cropped as the accuracy went very low.

REFERENCES

[1]OCR,Weblink: https://en.wikipedia.org/wiki/Optical_character_recognition[Accessed: 08- December-2018]

[2] S.Dhiman, A.J. Singh, "TesseractVsGocr A Comparative Study" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-4 , September 2013

[3]A. SHINDE , "Pre-processing and Text Segmentation for OCR", International Journal of Computer Science Engineering and Technology, pp. 810- 812 ,2012

[4] C.Patel, A.Patel, D.Patel, "Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study" International Journal of Computer Applications (0975 – 8887) Volume 55– No.10 , October 2012

[5] S. Vijayarani, Ms. A.Sakil, "PERFORMANCE COMPARISON OF OCR TOOLS" International Journal of UbiComp (IJU), Vol.6, No.3, July 2015

[6] Y. WEN, Y. L. An Algorithm for License Plate Recognition Applied to Intelligent Transportation System, IEEE Transactions on Intelligent Systems, pp. 1-16, 2011.

[7]]GOCR, http://jocr.sourceforge.net[Accessed: 11- December-2018]

[8] ORCAD, https://www.gnu.org/software/ocrad/ [Accessed: 11- December-2018]

[9]]Tensorflow, https://www.tensorflow.org [Accessed: 14- December-2018]

[10] Hui W. U., B. L., "License Plate Recognition system" International Conference on Multimedia Technology (ICMT). pp. 5425 - 5427. 2011.

[11] H. ErdincKocer, K. KursatCevik, "Artificial Neural Networks Based Vehicle License Plate Recognition", Procedia Computer Science, Volume 3, Pages 1033-1037. 2011.