Design and implementation of android application to extract text from images by using tesseract for English and Hindi

Brijeshkumar Y. Panchal¹, Gaurang Chauhan²

¹Teaching Cum Research assistant, Department of Computer Science and Engineering, DEPSTAR, CHARUSAT, Charotar University of Science and Technology, (CHARUSAT), CHARUSAT campus, Changa, 388421, India.
² ITsouls, Vadodara.

panchalbrijesh02@gmail.com
gaurang.chauhan02@gmail.com

Abstract. The proposed Implementation is on the Android Application to extract using Tesseract OCR in which the following concepts will be used, which are Adaptive Thresholding, Connected Component, Fine Lines, and Recognize Word. Using this Optical Character Recognition (OCR) Technology, an Application generated text which is printed on a clean, B/W or colourful background can be converted into a computer readable form ASCII. With the help of this Android Application using Tesseract OCR, the system has two ways for Text Extraction. The first one is to capture a photo while the second one uploads an image from the gallery after that system can proceed for as per the user requirement which portion of the image they want to crop or edit. After editing the picture, it converts into the text. This Android Application is for two languages, English and Hindi.

Keywords— Text Extraction, Android Application, OCR, English and Hindi Language, Image Processing, Tesseract 4, Tesseract 3 Tesseract.

1. Introduction

In this paper, we are trying to develop an Android Application to extract text from the image. Plain white, B/W or colourful background images are taken as an input by the system. The main objective of this research is to utilize this feature of the computer through an Android Application. This visual capability will be incorporated using an Android Mobile phone working on the Tesseract OCR Engine.

The 21st century is all about mobile phones. Statistics even suggest that there are almost 10 billion mobile devices currently in use. Developing new Application based on the current scenario makes it easy for us to grab the attention. Optical character recognition is designed for scanned documents. It is a potent tool. This Application is for all types of people. Android mobile phones are more widely used in today’s generation.
For the Application side, there is a number of the application used for the image to text converter. The system can convert our old manuscripts as well as in the banking side system should use to read our cheque manually. In the case of this implementation, the focus on manuscripts or typed pages. The user will use this Application for converted their images into text format so; they can work fast. The image to text converters is mobile and has high compatibility with most file formats and operating systems currently available in the market. Web developers often require the conversion of files from one format to another. It can be solved by image to text converters, which can convert several files simultaneously, thus saving time. Archiving and retrieval of data are necessary for creative fields like print-media, law, government records, libraries insurance firms, and banks. It becomes more accessible when the conversion process can take place effortlessly. Some image to text converters also supports a ton of languages, which makes the job more versatile, more comfortable, and accurate. Simple user interfaces and high accuracy of character recognition can get many things done quickly and easily.

![Block diagram of text extraction](image)

**Figure 1:** Block diagram of text extraction

This is a block diagram of Text Extrication, where whole process is going to perform under five steps, Input Images, then after pre-processing on given images, Text Detection Text Extraction and lastly Text Recognition. This step may be reduced, when we use API like Tesseract OCR. Because, API has own Process [1][2].

### 2. Android

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as
smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It was unveiled in 2007, with the first commercial Android device launched in September 2008[3].

So android is an operating system designed with mobile in mind, the place where your phone’s functions and applications live. Everything you see on the display of your device is a part of the operating system. The Android OS is divided into various version numbers, implying significant jumps in features, operation, and stability, which usually have codenames. So, if you hear someone say Android Marshmallow, Nougat, Oreo, Pie, or the latest Android 10, that is just the name of the version of Android you might have on your device[4].

2.1 Advantages of Android [5]

(1.) Universal Chargers
(2.) More Phone Choices Are a Clear Advantage of Android
(3.) Removable Storage and Battery.
(4.) Access to the Best Android Widgets
(5.) Better Hardware
(6.) Infrared
(7.) Custom Keyboards
(8.) Google Play Is More User-Friendly

3. TESSARECT

There are number benefits of tesseract OCR, but main focus on improve the productivity and efficacy of work. The ability to instantly search through content is extremely useful, especially in an office setting that has to deal with high volume scanning or high document inflow. Now we can also use the copy and paste software on the paper instead of rewriting anything to fix it. Tesseract OCR is fast and reliable, meaning that the integrity of the text stays unchanged thus saving time. Once paired with other tools such as scanning and file compression, the benefits of tesseract OCR really shine. Workflow improves as workers no longer have to spend time on manual labour and can work quicker and more effectively.

This Tesseract OCR engine - libtesseract and command line program - tesseract. Tesseract 4 adds a brand new neural net (LSTM) based mostly OCR engine that is concentrated on line recognition, however conjointly still supports the heritage Tesseract OCR engine of Tesseract 3 that works by recognizing character patterns. Compatibility with Tesseract 3 is enabled by mistreatment the inheritance OCR Engine mode (--oem 0). It also needs trained data files which support the legacy engine, for example those from the tessdata repository [5].

Tesseract has Unicode (UTF8) support, and may recognize a hundred "out of the box" languages. Tesseract supports various us file formats: Plain text, HTML, XML, invisible-data only XML, TSV. Additionally, the master branch has experimental support for the processing of ALTO (XML)[6].
Figure 2: Phases of Tesseract

3.1 Adaptive Thresholding
Adaptive thresholding typically takes a grayscale or color image as input and, within the simplest implementation, outputs a binary image representing the segmentation. For every pixel within the image, a threshold must be calculated. If the pixel value is below the edge it's set to the background value, otherwise it assumes the foreground value [8].

3.2 Connected Component
For two pixels to be connected they have to be neighbours and their gray levels must specify a particular criterion of similarity. As an example, during a binary image with values 0 and 1, two pixels could also be neighbours but they're said to be connected providing they need same values. A pixel p with coordinates (x, y) has four horizontal and vertical neighbours known as 4-neighbors of p, given as: (x, y+1), (x, y-1), (x+1, y), (x-1, y) and 4 diagonal neighbours given as: (x+1, y-1), (x-1, y-1), (x+1, y+1), (x-1, y+1). Together these are known as 8-neighbors of p. If S represents subset of pixels in an image, two pixels p and q are said to be connected if there exists a path between them consisting entirely of pixels in S.

For any pixel p in S, the set of pixels that are connected thereto in S is termed a connected component of S [4].
3.3 Find Lines-Words
The line finding algorithm is intended in such the simplest way in order that a skewed page is recognized without skewing it, thus saving the loss of image quality. The first parts of the method are blob filtering and line construction. Assuming that page layout analysis has already provided text regions of roughly uniform text size, a straightforward percentile height filter removes drop-caps and vertically touching characters. The median height approximates the text size within the region, so it's safe to filter blobs that are smaller than some fraction of the median height, being presumably punctuation, diacritical marks, and noise. The filtered blobs are more likely to suit a model of non-overlapping, parallel, but sloping lines. Arranging and processing the blobs by x-coordinate makes it easier to assign blobs to a singular text line, while tracking the slope across the page, with the significantly curtailed danger of attributing to an incorrect text line within the presence of skew. Once the filtered blobs are assigned to lines, the smallest amount median of squares is employed to estimate the baselines, and therefore the filtered-out blobs are fitted into the acceptable lines. The concluding step of the road creation process merges blobs that overlap by a minimum of half horizontally, putting diacritical marks along with the proper base and properly associating parts of some broken characters [7].

3.4 Recognize word
Part of the popularity process for any character recognition engine is to spot how a word should be segmented into characters. The initial segmentation output from line finding is assessed first. The remainder of the word recognition step applies only to non-fixed pitch text. (1.) Chopping Joined Characters (2.) Associating Broken Characters [7]

4. Flow of android application
After opening the application, there will be a pop up of a splash screen that will appear for a while, and a page will appear. It will have a login form that requires an email address and a password to be filled. If the credentials are correct, you will be able to access the next page, and if you are a new user, you may have to sign up for the app. You will need to include all the necessary information; full name, email address, password, the reason for using the app, and contact number. After filling up the form, you'll have to submit the way, and it will redirect you to a login page. If you feed the wrong information on the login page, it will provide you a popup message of invalid login credentials. Otherwise, it will display a success message. The second screen will ask you to select an image and will provide two options - Hindi and English. Whichever option is selected, the interface of the Application will remain the same, but the process in the backend will be different.

Result and an image preview will be displayed on both the pages. The application will ask for two options. Either the user can select the image from the gallery or click a picture Using the camera. Additionally, cropping tools are also provided, which can help set the image straight. Once the image is selected or clicked, the user will be shown the final result under the 'result.'
Output 1: Select Image

Output 2: English Image Camera Capture

Output 3: English Output
Output 4: Hindi Image- Select from Gallery

Output 5: Hindi Output

Output 6: Text copy edit/copy/share
5. Evaluation of text extraction

For performance of recognition, we simulate the proposed character recognition system to various characters such as English and Hindi. We calculate the Recognition Rate (R)\[9\][10] as performance criterion of recognition.

\[
R = \frac{\text{Number of Correctly recognize characters}}{\text{Total Number of characters}}
\]

| Image Number | Image Type | Number of characters in Image | No of characters Extracted | Recognition Rate (R) (in Percentages) | Time taken for OCR Manually (in Seconds) |
|--------------|------------|-------------------------------|-----------------------------|----------------------------------------|----------------------------------------|
| Image 1      | color      | 15                            | 13                          | 88                                     | 0.5                                    |
| Image 2      | color      | 20                            | 18                          | 89                                     | 0.70                                   |
| Image 3      | B/W        | 80                            | 75                          | 95                                     | 0.52                                   |
| Image 4      | color      | 6                             | 5                           | 90                                     | 0.3                                    |
| Image 5      | color      | 9                             | 7                           | 92                                     | 0.5                                    |

*Table 1: English Language evolution*

| Image Number | Image Type | Number of characters in Image | No of characters Extracted | Recognition Rate (R) (in Percentages) | Time taken for OCR Manually (in Seconds) |
|--------------|------------|-------------------------------|-----------------------------|----------------------------------------|----------------------------------------|
| Image 1      | B/W        | 5                             | 2                           | 40                                     | 0.80                                   |
| Image 2      | color      | 12                            | 8                           | 58                                     | 0.32                                   |
| Image 3      | B/W        | 8                             | 4                           | 50                                     | 0.75                                   |
| Image 4      | color      | 9                             | 5                           | 55                                     | 0.6                                    |
| Image 5      | color      | 29                            | 17                          | 58.69                                  | 0.8                                    |

*Table 2: Hindi Language Evolution*
From this Tables data, we can predicate that the average accuracy of the English language is 91% and the average accuracy of the Hindi language is 52.4%.

![Bar Chart: ENGLISH AND HINDI RECOGNITION RATE IN PERCENTAGE](image)

**Figure 3:** Comparison between English and Hindi in the context of recognition rate (R)

6. Conclusion

In this implantation, we developed an Android app text extraction (English and Hindi) using Tesseract OCR. The process of text extraction contains input image, pre-processing it, text detection, text extraction, and last text recognition. In-text extraction phases of a tesseract, adaptive thresholding, connected component analysis, fine lines & words and recognize the name are utilized. With the help of this approach, the text from the captured image or saved image is recognized easily with less effort and more accuracy. Through this, we implemented many techniques which are mandatory to convert image to text. It is useful for other because it is an offline application, so there is no need for an internet connection. Old documents can also be scanned and corrected with the help of this application. Here we calculated Recognition Rate(R). Particular this implantation, from this Evolution we can conclude that, English Recognition Rate(R) is higher than Hindi.

7. Future work

In the future, the accuracy of the text extraction will improve, and the inclusion of other languages will be considered. There is an additional option to upload the images to collect the data. Even in uploaded images, there are various options to resize the portion of the text. In image section, noise removal by the system itself is possible. Also, this Application will implement extraction features for the handwritten characters. We can use other techniques that can be utilized, and that can help inspect the DPI of the images. We shall do whole implantation with the use of Machine learning Algorithms and Customized Dataset [11]. We can develop whole system in IOT base environment also [12].

8. Examples taken from published papers:

[1] M. Khodadadi, A. Behrad, *Text Localization, Extraction and Inpainting in color Images*,
Iranian Conference on Electrical Engineering (ICEE2012), May 2012, pp. 1035-1040.

[2] Sanjay Shah, Chintan Modi, Manisha Patel, International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), 2011 *Novel Approach for Text Extraction from Natural Images Using ISEF Edge Detection*. G. H. Patel College of Engineering & Technology V.V.Nagar, Gujarat, India

[3] https://en.wikipedia.org/wiki/Android_(operating_system)

[4] Mrs. Prachi Sasankar, Sadabai Raisoni, Mrs. Usha Kosarkar IOSR Journal of Computer Engineering (IOSR-JCE) *Research on Development of Android Applications* Women’s College, Nagpur SNDT Women’s University, Mumbai, India. BCA, G.H.R.I.T.Nagpur, R.T.M.Nagpur University, Nagpur, India. e-ISSN: 2278-0661, p-ISSN: 2278-8727 PP 28-32

[5] https://thisisglance.com/top-ten-advantages-of-android/

[6] https://mindorks.com/android/store/OCR-Engines/tesseract ocr/tesseract

[7] An Overview of the Tesseract OCR Engine Ray Smith Google Inc. theraysmith@gmail.com.

[8] http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm

[9] Chirag Patel, Atul Patel, PhD, Dharmendra Patel. International Journal of Computer Applications *Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study*. Smt. Chandaben Mohanbhai Patel Institute of Computer Applications (CMPICA) Charotar University of Science and Technology (CHARUSAT) (0975 – 8887) Volume 55– No.10, October 2012 50

[10] K.Karthick, K.B.Ravindrakumar, R.Francis, S.Ilankannan. International Journal of Recent Technology and Engineering (IJRTE) Published By: Blue Eyes Intelligence Engineering & Sciences Publication Retrieval Number A2670058119/19©BEIESP Steps Involved in *Text Recognition and Recent Research in OCR; A Study ISSN: 2277-3878, Volume-8, Issue-1, May 2019 3095

[11] Parilkumar Shiroya; Darshan Vaghasiya; Meet Soni; Vrajkumar Patel; Brijeshkumar Y. Panchal. “*Book Genre Categorization Using Machine Learning Algorithms (K-Nearest Neighbor, Support Vector Machine and Logistic Regression) using Customized Dataset*”, International Journal of Computer Science and Mobile Computing, Vol.10 Issue.3, March-2021, pg. 14-25.

[12] Brijesh Y. Panchal, Krunal Parmar, Mikisha Patel, Richa Panchal, “*Design and Implementation of Auto Irrigation System by Checking Soil Moisture According to the Crops.*” International Journal Computer Technology & Applications, Vol 11 (4), 61-65.