

Design of An Electronic Narrator on Assistant Robot for Blind People

Rizqi Andry Ardiansyah

Research Center of Electrical Power and Mechatronics, Indonesian Institute of Sciences, Cisitu 21/154 D, Bandung 40135, Indonesia

Abstract. Many personal service robot is developed to help blind people in daily life, such as room cleaning, for navigating, object finding, reading and other activities. In this context, the present work focuses the development of an image-to-speech application for the blind. The project is called Design of An Electronic Narrator on Assistant Robot for Blind People, and the final purpose is the design of an electronic narrator application on personal service robot that helps to narrate a text on a book, magazine, a sheet of paper etc to a blind person. To achieve that, a Raspberry pi board, a light sensor, OpenCV computer vision library, Tesseract OCR (Optical Character Recognition) library, eSpeak Text-to-Speech Synthesizer (TTS) library are integrated, which is enables the blind person to hear a narration from text on a book, magazine, a sheet etc.

1 Introduction

In the future, personal service robot is going to be an important thing in people life, playing an important role as appliances, servants and assistants [1]. Many personal service robot is developed to help blind people in daily life, such as room cleaning, for navigating, object finding etc [2-4]. One of the difficult activity for blind people is reading. They usually use Braille book to improve their knowledge [5]. Generally, books, magazines, newspapers are not printed in Braille, so it's must be printed in Braille. This process needs more time and costs, so an electronic narrator on assistant robot was developed with the aim to help a blind person to read general books.

There are some projects being done in the image to speech research area with various of method : The paper *Camera Reading For Blind People* [6], describes the process of developing a prototype of a mobile application (iOS), which can be used on an iPhone/iPod, Touch/iPad. The system uses already existing Tesseract OCR and TTS frameworks, it implement the combination of filters CIColorControls and CIColorMonochrome. In this paper also make comparison of three OCR tools Tesseract, Abbyy and Leadtools; *The Phone Reader* thesis [7], presents an Android mobile phone application system that allows recognizing the text in an image using the Tesseract OCR, the reading of the text is made through the TTS in the selected language. The OCR processing and translation are made on an external server, which requires an internet connection for communication between the server and the device; The paper *Open Source OCR Framework Using Mobile Device* [8] describes a project in which the image is captured with the camera of iPAQ PDA, and it is converted to 1-

channel MS-Windows bitmap format then the image is converted by an OCR tools into ASCII text files. In the postprocessing stage, all alphabetical and numerical characters are removed, no filter being applied. Flite TTS engine is applied for speech synthesis. Research in this area still needs to achieve better result and to overcome the limitations of real life situations which light conditions, the focus, alignment or the existence of shadows do not allow the effective recognition by OCR.

This paper describes the design prototype of an electronic narrator on assistant robot to help blind people. This prototype is built using Logitech webcam c250m, a Raspberry Pi board, a BH1750FVI light sensor, OpenCV computer vision library [9] [10], Tesseract OCR (Optical Character Recognition) library, eSpeak Text-to-Speech (TTS) library, and the source code is written in Python Programming Language [11]. The complete overview of all the components used in this prototype is on section 3. The rest of this paper is organized according to the following structure: In Section 2, the overall system architecture is described; In section 3, All of hardware and software components is presented in details; Some experiments result is presented in section 4; and section 5 presents some conclusions.

2 Overall system architecture

Building an electronic narrator requires integration of many hardware and software components. Figure 1 shows the overall system architecture of an electronic narrator prototype that i have developed.

The prototype includes a processor unit, a webcam, a light sensor, a computer vision library, a text to speech library, and an optical character recognition (OCR) library. A Raspberry pi board is configured as a processor

^a Corresponding author: rizqinhood@gmail.com

unit. It handles all the instructions. A Logitech webcam c250m is configured as an image capturer. Opencv library is configured as an image processor. First, image that contain text is captured by Logitech webcam C250m with display screen resolution of 640 x 480 pixels. The camera has a resolution of 1.3 megapixels with VGA sensor. In the Preprocessing stage, light conditions is sensed using BH1750FVI light sensor, then it is used as a reference value to adjust the contrast and the brightness of image. Then the process is continued to image thresholding. In this case, i use Binary Thresholding. Image thresholding, brightness and contrast adjustment is used to improve the optical character recognition result. After preprocessing stage, then optical recognition is begin to convert image to digital text. In the Postprocessing stage, Levenstein distance algorithm is used to compare the digital text with original text. Then the result of the comparison process is used as reference value in image thresholding process in the Preprocessing stage. The last is, the conversion process of the digital text into a human voice synthesis using eSpeak TTS library. Python programming language is used make all of the library and the hardware works together. Figure 2 shows the chart diagram of this system.



Figure 3. Raspberry Pi Board

3 Overview of hardware and software components

All hardware and software components are used in this research are described here.

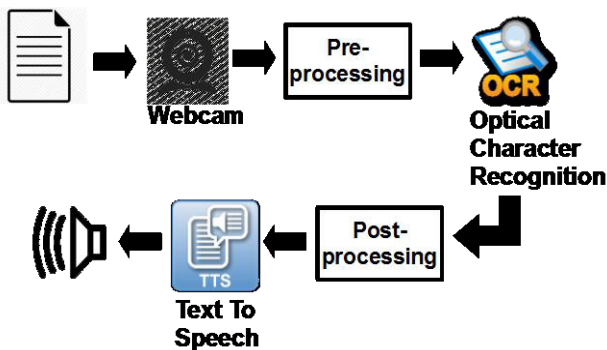


Figure 1. System Architecture

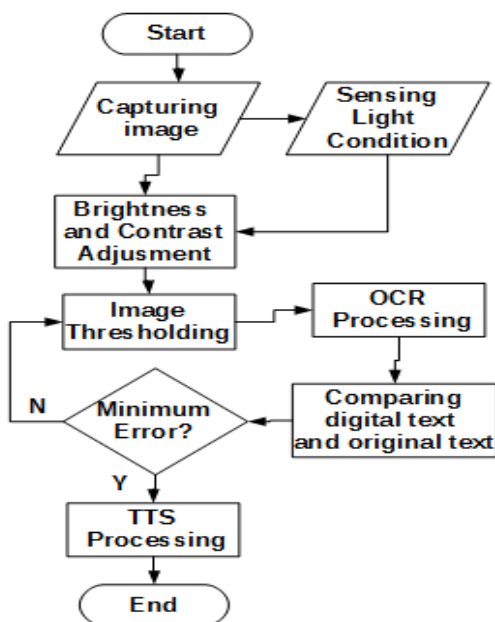


Figure 2. Chart Diagram of the System

3.1 Raspberry pi overview

Raspberry pi is a credit card - sized single board computer (Single Board Circuit / SBC). In this research used the Raspberry Pi board model B with Raspbian Linux inside. Raspberry Pi Model B has 512Mb RAM, 2 USB ports, Analog Audio output port, and an 100mb Ethernet port. It has a Broadcom BCM2835 system on a chip which includes an ARM1176JZF-S 700 MHz processor [12]. It supports a number of operating systems including a Debian-based Linux distro, Raspbian, which is used in this research [13]. The Raspberry Pi model B board can be shown on Figure 3.

3.2 OCR (Open character recognition) overview

Optical character recognition (OCR) is the process of recognition and automatic conversion of existing characters in an image into the digital text format, which can be used in various applications. The OCR has been widely studied and has displayed considerable advances regarding the performance and accuracy of the obtained results [14]. The process of optical character recognition can be summarized as a process that follows a set of steps [15]:

- Optical image acquisition
- Location and segmentation
- Preprocessing
- Feature extraction
- Classification
- Postprocessing

3.3 Text to speech synthesizer (TTS) overview

The goal of speech synthesis is to automatically generate artificial speech waveforms from another data format. Especially, a TTS synthesizer is a computer-based system that should be able to read any text aloud. The main components of a TTS are [16] :

- An NLP (Natural Language Processing) module responsible for the production of the phonetic transcription of an input text and, in some cases, the generation of prosody (intonation, duration, intensity or power);
- A DSP (Digital Signal Processing) module, which converts the data output from the NLP module into speech.

3.4 Levenshtein distance algorithm

The Levenshtein distance algorithm was created by the Russian scientist Vladimir Levenshtein. The basic principle of the algorithm is the measurement of similarity between two strings. This is done by calculating the number of basic operations or changes needed to make two strings equal. Changes may consist of a substitution, insertion or deletion of a character. The Levenshtein distance $d(p, t)$ between two strings (p and t represent the strings) is the minimum number of operations required to make $p = t$. For example, if $p = \text{"kitten"}$ and $t = \text{"sittin"}$, the Levenshtein distance is $d(p, t) = 2$ since the strings can be made using the same two substitutions (replacement of letter 'k' by letter 's', exchange of letter 'e' by letter 'i') [6]

3.5 Image thresholding

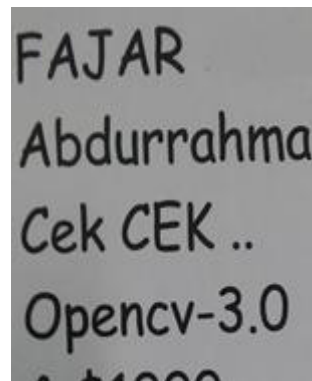
Thresholding is the simplest method of image segmentation. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. In this project this method is used to make the text in the image can be recognized better. The basic principle of image thresholding is the replacement of each pixel's colour which is has lower value than threshold value in an image with another colour. For example : we will replace the image pixels into black pixel if the image intensity is less than threshold value or a white pixel if the image intensity is greater than that threshold value.

4 Experimental results and analysis

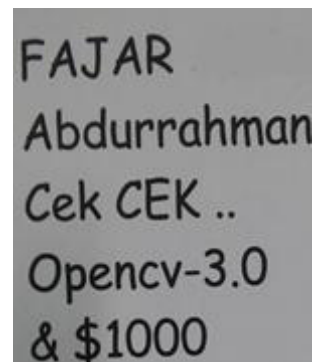
In this research, tests were made with 10 images on three difference lightning conditions : 6 LUX, 62.5 LUX, and 87.5 LUX. In this research, the tests results before using Preprocessing stage and after using Preprocessing stage are presented. Some images used in the tests are presented in Figure 4 (a)(b)(c) and the images result of thresholding process are presented in figure 4 (d)(e)(f).



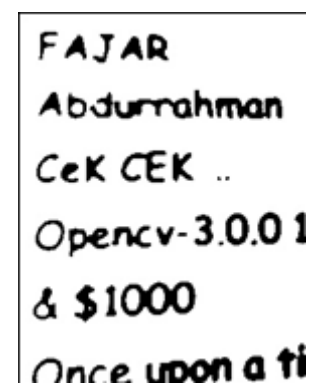
(a)



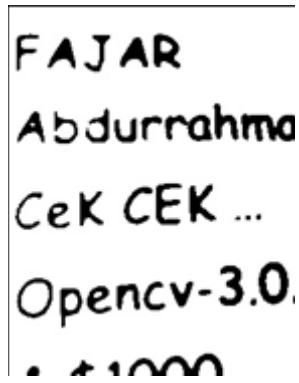
(b)



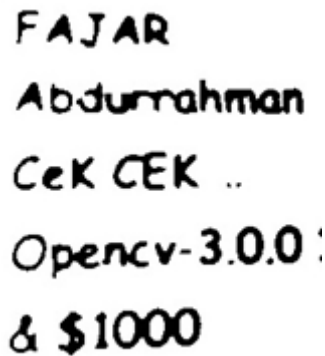
(c)



(d)



(e)



(f)

Figure 4. Example of images used in tests, on 6 LUX (a), on 62.5 LUX (b), on 87.5 LUX (c), images after thresholding process (d)(e)(f)

The results of the tests with three difference lightning conditions before Preprocessing is presented in Figure 5.

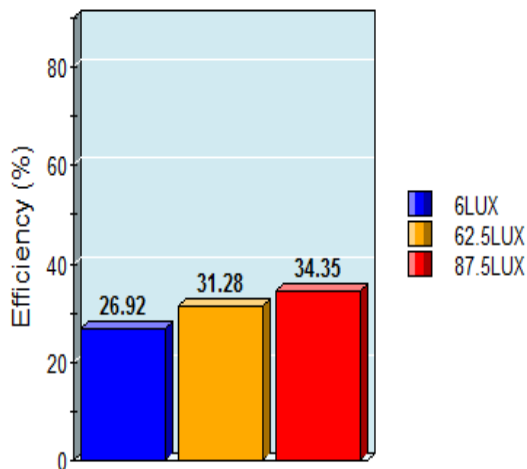


Figure 5. The results of the tests with three difference lightning conditions

The comparison between the results of the tests without Preprocessing and using Preprocessing is presented in Figure 6.

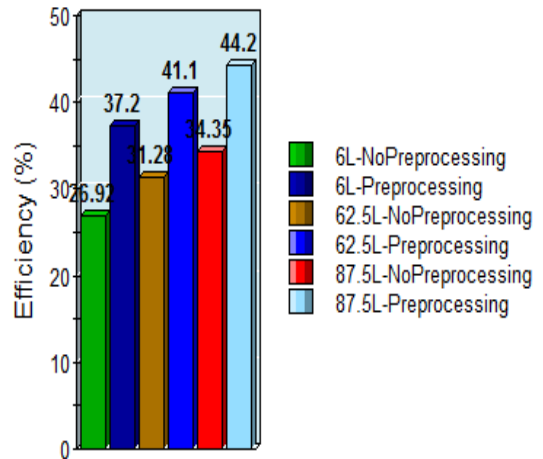


Figure 6. The comparison between the results of the tests without Preprocessing and using Preprocessing

The median value results of this research is compared with two other trial edition commercial OCR tools, Read Iris and Abby. The comparison results between three OCR tools, tesseract OCR, Read Iris 14, and Abby Fine 12 Pro can be shown in the following graph in Figure 7.

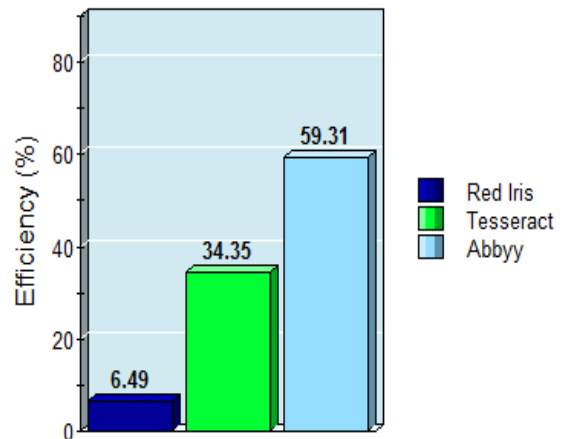


Figure 7 The median value comparison of recognition result between Read Iris, Tesseract, and Abby

By analyzing the test results in Figure 5, it can be seen that the lightening conditions gives affects to the efficiency of the optical recognition process. It presents that the higher light conditions give better results to the optical recognition process. By analyzing the test results in Figure 6, it can be seen that the using of image thresholding, brightness and contrast adjustment improve the optical recognition results of Tesseract with the median improvement value about 10.9 %. By analyzing the test results in Figure 7, it can be seen that the optical recognition results of Tesseract is better than the optical

recognition results of Read Iris with difference value of efficiency about 39.31 %, but the optical recognition results of Tesseract is lower than the optical recognition results of the new version of Abbyy with difference value of efficiency about 13.5 %. Then the results in this research will be used as reference to try new method to improve the results of optical recognition.

5 Conclusions and future works

Based on the experiment i conclude that the lightening conditions gives affect to the optical recognition results. It presents that the higher light conditions give better results to the optical recognition process. The using of image thresholding process, brightness and contrast adjustment improve the optical recognition results of Tesseract with the median improvement value about 10.9 %. It was found in the optical recognition tests represented a set of meaningless characters, then in the future works will be looking for a method to minimize the meaningless character. Although the final application does not represent reliable solution for blind people in daily life yet, because of many factors, the application has many lack of function like unautomatically adaptable brightness and contrast adjusment for difference light conditions, unautomatically text detection in daily life, the research in image to speech area still needs to achieve better results. In the future works, the adaptable of optical recognition in difference conditions is needed. It is also reserved for future works the automatically text detection in daily life.

Acknowledgement

Author would like to thank to Research Center of Electrcal Power and Mechatronic for supports in facilities and Fajar Abdurrahman for helping on the experiments.

References

1. V.Alvarez-Santos, J. Vis. Commun. Image R. **25**, 499–509 (2014)
2. R. Gnana Praveen, Roy P Paily, *Procedia Engineering* **64**, 351 – 360 (2013)
3. Javier V. Gomeza, Frode Eika Sandnes, *Procedia Computer Science* **14**, 218 – 225 (2012)
4. Helen Knight, *New Scientist* vol **214**, 19 (2012)
5. Jenn Tang, *Expert Systems with Applications* **40**, 2817–2827 (2013)
6. Roberto Neto, Nuno Fonseca, *Procedia Technology* vol **16**, 1200 – 1209 (2014)
7. BIHINA, M. M. B., Grahamstown, South Africa, November, (2012)
8. Zhou, S.Z., *Multimedia on Mobile Devices*, Edited by Creutzburg, Reiner; Takala, Jarmo H. *Proceedings of the SPIE*, Vol **6821**, article id. 682104, 6 pp. (2008)
9. OpenCV library, OpenCV website, 27 January 2015 [Online], Available: <http://sourceforge.net/projects/opencvlibrary/>.
10. Hiroki Sugano, Ryusuke Miyamoto, *Computer Vision and Image Understanding* **114**, 1273–128 (2010)
11. Python Software Foundation, Python home page, 28 January 2015 [Online], Available: <http://www.python.org>.
12. Matt Richardson, Shawn Wallace, O'Reilly Media Inc., USA, (2013)
13. Sheikh Ferdoush, Xinrong Li, *Procedia Computer Science* **34**, 103 – 110 (2014)
14. FUJISAWA H, *In Proceedings of the Ninth International Conference on Document Analysis and Recognition – Volume 01*, pages 3-7 (2007)
15. EIKVIL, LINE. “OCR Optical Character Recognition”, (1993)
16. Dutoit, T. , *Journal of electrical and electronics engineering, Australia*; **17(1)**: 25-36 (1997)