Text Reader for Visually Impaired Person

Mohd Nadhir Ab Wahab, Ahmad Sufril Azlan Mohamed, Abdul Syafiq Abdullah Sukor, Ong Chia Teng

School of Computer Sciences, Universiti Sains Malaysia, 11800, Penang
Centre of Advanced Sensor Technology (CEASTech), Universiti Malaysia Perlis, 02600, Arau, Perlis

*Corresponding author: mohdnadhir@usm.my

Abstract. There are approximately 1.3 billion people in the world have visual impairment issue. They usually have to read printed material using Braille. However, there are limitations for these people when the material is not printed in Braille. Although there is much electronic equipment that can help them to read, the prices are too expensive to afford. Thus, this paper proposes an affordable mobile application which is designed for the visually impaired person. The mobile application is able to capture the image of printed material with a mobile camera. The captured image is then converted to text by using image-to-text conversion in Optical Character Recognition (OCR) framework. Finally, the text will be read out into speech format using text-to-speech conversion in Text to Speech (TTS) framework. As a result, a person who has visual impairment can understand the printed material which is not written in Braille through listening instead of touching. Some alert sound is provided to allow the users to know what exactly happened in the mobile application. It is user friendly for the visually impaired person since the designed system has sound for guideline so they can always get to know the process of the application.

1. Introduction

Visual impairment is known as a person who has lost its ability to see clearly as a normal person. According to the World Health Organization (WHO), there are about 2.3 billion people have some form of visual impairment worldwide, which represents one-third of the world population [1]. These people are usually deprived of the ability to read textual information, which limits their reading mobility in the world. Referring to the International Classification of Diseases (ICD-11 version 2019), a person is considered as visually impaired if presented distance visual acuity is worse than 3/60 and there are approximately 36 million people have fallen in this field [2].

Table 1 shows the categories of visual impairment worldwide. Distance visual acuity is the basic measurement to identify the categories of individual’s visual impairment. The test in distance visual acuity is carried out by using different size letters arranged in decreasing size called Snellen Chart [3]. A person carries on the eye test by one eye and covers another eye. The visually impaired person has visual acuity less than 3/60, which means the smallest size of letter that the person can identify is 60 in 3 meters or worse than that [4].

When it comes to read or write, the visual impairment person uses Braille’s system. The Braille’s system is a ‘basic cell’ of six raised dots arranged like a domino and each braille, the letter of the alphabet is made up of the combinations of dots from this basic cell [5]. Total of 63 possible combinations of dots not only included in the alphabet but also can represented punctuation, and also a group of letters
known as contraction words such as isn’t, aren’t, don’t and others [6]. Although Braille is a useful language for visual impairment person when reading without the use of sight, there are some limitations that are associated with the system.

| Category of vision | Degree of impairment | Presenting distance visual acuity | Alternative Definition |
|--------------------|----------------------|-----------------------------------|------------------------|
| Mild or Normal     | None / Slight        | Better than 6/18                  | Normal/ near normal    |
| Low vision         | Moderate             | Less than 6/18                    | Moderate low vision    |
|                    | Severe               | Less than 6/60                    | Can see 60 font letters in 6 meters or less |
| Blindness          | Profound             | Less than 3/60                    | Can see 60 font letters in 3 meters or less |
|                    | Near total           | Less than 1/60                    | Can see 60 font letters in 1 meter or less |
|                    | Total                | No light perception               | Total blindness. Include absence of eye |

A survey conducted by researchers found that the average reading rate of children through Braille just under half the reading rate of print readers with the use of sight [7]. This is because normal human eyes can read several words in one time but with fingers, it can only pass over Braille in one-by-one word. There are also limited reading materials such as book, paper, and journal that are not printed in Braille. There is much electronic equipment that can help the visually impaired person to read but they are comparatively expensive [8], and it leads to limiting the reading media that the visually impaired person can use. Furthermore, it is imperatively time-consuming to learn a new language in Braille and use a new piece of electronic equipment [9].

This paper propose to build an application which is convenience for the visually impaired person since it can be applied using smartphone. A mobile application is designed to allow users to capture the image of printed material. Then the image will be converted to text using Optical Character Recognition (OCR) approach [10]. After that, the built text will be converted to speech and speak out using Text to Speech (TTS) framework [11].

2. System Design

2.1. System Architecture

![Fig. 1. System Architecture of Text Reader](image_url)
Figure 1 shows the system architecture of the text reader system. Users need to install the Text Reader application in their android smartphone. They will take an image as an input to the app and it will process the image in the system (convert image to speech). After processing, Text Reader application will convert the output from the speech to the user app and speak out in a voice mode to the user.

There are three different types of modules in the designed system. First, the Pre-process Image module is used to convert the original image into a better representation. This will help to obtain a clear output result. Then, the OCR module is used to convert the captured image into text using OCR algorithm. Finally, the obtained text will be converted to the speech mode in the TTS module. Table 2 presents the breakdown information of each module in the system of the user app.

| No | 1 | 2 | 3 |
|----|---|---|---|
| Module | Pre-process Image | Optical Character Recognition (OCR) | Text to Speech (TTS) |
| Input | Image | Pre-processed Image | Text |
| Output | Pre-processed Image | Text | Speech |

Description

The image captured by user will undergo pre-process to obtain a clearer image and more accurate result in OCR.

The pre-processed image will convert to text in English by using OCR technique.

Text obtain from OCR will convert to speech by using TTS technique and transfer back to front-end user app to speak out to user.

2.2. Flowchart of Text Reader System

![Flowchart of Text Reader System](image)
Figure 2 shows the flowchart of the text reader system. From the start, it will open the camera to allow users to capture an image. If the user has captured the image, it would proceed to process the image and the application will stop if the user does not want to capture anymore image. In the image processing, it will convert the captured image to the text and convert the text to speech. After the conversion is completed, it will speak out the text. Then, the user can choose whether to capture another image or terminate the application.

3. System Implementation

3.1 Module Implementation

Android Studio [12] is used to create the mobile application for the text reader system. It is divided into two parts which are known as the user side and the system side. The user side can be regarded as the front interface for the user use (front-end) which can be viewed by the user. Meanwhile, the system side is the process taken to produce the output of the system. It acts as the back-end of the system. Figure 3 presents the module implementation of the system. Each of the systems is composed of three modules respectively which are “Check Permission”, “Capture Image”, and “Show result of Image” in the user side while “Pre-processing Image”, “Convert Image to Text”, and “Convert Text to Speech” are in the system side.

![Fig. 3. Implementation of Mobile Application](image)

3.2 User Implementation

3.2.1 Check Permission Module

When the user has launched the text reader application, a notification sound “Text Reader is started, welcome.” will be played to alert the user that the application has been launched and processed (Figure 4). Then, the image storage and camera permissions will be asked by the system using a message pop-up.
If the android phone system is running lower than Android 6.0 (API level 23) and the app’s targetSdk version is lower than 23, it does not have any app permissions, so the user can proceed to use the phone camera directly such in Figure 5.

For the case which requires the user to allow the permissions, the text reader application will give notification sound “Phone permissions is needed” to alert the visual impaired people to ask for the setting. If the permissions are denied, the application will be terminated, it will only process to use the camera if the user allows both camera and storage permissions. If “deny & don’t ask again” is chosen, it will terminate the application and not be able to launch unless the user changes their permissions in the phone settings. The message “Some permissions is Denied” will pop up when the app is terminated due to permissions issue.

Since the text reader application needs people to help visual impairment person to install, it also requests people to set up the permission of phone when launching the application for the first time.
3.2.2 **Capture Image Module**

After allowing both camera and storage permissions, the system will launch the OpenCV camera of the phone and automatically adjust the screen in landscape mode. The notification sound “The camera is on, please point to the document.” will appear when the camera is on and alert the user that he/she needs to point to document to capture the image.

The system will detect a rectangle shape of the image which represents the printed document and red colour is used to draw the outline of the rectangle such in Figure 6. It will keep on alert if the user cannot detect any rectangle shape (no document) and ask them to point directly to the printed document. When detected a rectangle shape object, it will play notification sound such as “move right”, “move left”, “move in front”, “move backward” or “move down” to make sure the detected rectangle object is in the middle of the screen and not too small.

![Fig. 6. Camera in Text Reader](image)

When all the requirements are fulfilled (rectangle object fix in the middle of the screen and not too small), it will play sound “Hold for 2 seconds”, before it is automatically captured the image. The user also can use volume down button to capture the image. There are 3 ways to capture the image in this application, auto-capture, click on the button in the middle bottom of the screen and volume down button which can easily found by touching.

When the image is captured, “Image is captured. Saved Image” is played to alert the user. Then, it will process to convert the captured image to speech (back-end) and will notify by the sound “Please wait, the image is converting to text “. The system also allows the user to use volume up button to exit the application.

3.2.3 **Image Result Module**

After converting the image to text, it will show the text on the screen as shown in Figure 7. It has 2 cases, if the string text contains more than 60% English word, it will play sound “Conversion is completed “. Use volume down button to go back Camera and volume up button for the exit. Press any place to start or stop speech”, if in the second case which the English word in string text is lower than 60%, the alert sound will play, "Done conversion. Poor result, you can use volume down button to go back Camera or volume up button for the exit. Press any place to start and stop speech if you still want to check the result “. 
Fig. 7. View Captured Image

For case 1, the captured image is clearer and the result is good, but for case 2, it can be assumed as a bad captured image with a bad result may due to the blur image, not an image with text nor in English language. It will request the user to capture image again but also allow the user to listen to the result text.

For both cases, it will keep on alerting the user to play and stop the speech by pressing any space on the screen. Then the text will speak out to the user in format “Text of image that had been captured is ……” when the user press on any space on the screen. It will also use volume down button to go back to the camera (to capture another image) or use volume up button to exit this application and sound “bye, see you” will be played.

3.3 System Implementation

3.3.1 Image Pre-processing Module

After capturing the image, it will proceed to the back-end system. The image will make into 4 different direction and each image will undergo preprocessing by converting to a grayscale image, obtain Otsu threshold, remove noise, adjust angle of the document (fit to the screen), and others to obtain a better result. Python programming and OpenCV is used to do the back-end process.

3.3.2 Image to Text Module

The preprocessed image (4 different direction images) will convert into text by using Tesseract and store each result. The string text for each image will compare and choose the image which has the best result. Each string text will check whether it contains more than 15% special symbol such as “( ); * ~ » .” then it will be rejected. If the string image contains less than 15% of special symbol, then it will count the English word contain in the string text and get the string text with the highest count in the English word. Natural Language Tool Kit (NLTK) is used to check the English word in the string text.

Once chosen the highest count in English word, it will check whether the English word is more than 60% in the overall text. If more than 60%, it will be case 1 and the string text will be corrected by using NLTK to correct some words such as “lisr” to “list” in order to get a more accurate result. If the English words contain in the string text is less than 60%, it will be in case 2 and it will not be doing correction in string text.

3.3.3 Text to Speech Module

The string text will be speaking out by using Text to Speech engine in Android Studio. A wave file in the form of .wav containing the speech format is created and stored in the phone memory. The speech
can be played using wave file player in the smartphone. The speech waveform is varied according to the different text from OCR output.

4. Conclusion and Future Work
In conclusion, the text reader system is a useful mobile application that can help the visually impaired person to read printed materials which are not written in Braille through capturing the image using a camera in smartphone. A person who is visually impaired can capture the image by pressing the volume down button of the smartphone which can be easily found using touching or letting the image being auto-captured. A notification sound will be played to keep on alerting the user about the process of the application. The text of the image will be converted as audio so the user can read the printed material by listening without the use of sight. However, it is only limited to detect English word and work fine in no format printed material.

Future work will include the enhancement of the OCR technique to recognize the word with very small fond size and increase the accuracy of the printed image. Furthermore, the mobile application can be made to not only convert the English language but also any other languages. It is also worth to mention that the audio can also be played in other languages than English (Translation the English text to different language speech).

Acknowledgement
This project is supported by USM Short Term Grant (PKOMP/6315262) and part of the collaboration project under Robotics, Computer Vision, and Image Processing (RCVIP) Research Group of Universiti Sains Malaysia (USM) and Centre of Advanced Sensor and Technology (CEASTech), University Malaysia Perlis (UniMAP).

References
[1] World Health Organization (WHO), World Report on Vision. 2014.
[2] W. H. Organization, “International Statistical Classification of Diseases and Related Health Problems,” 2011.
[3] I. S. for the E. of Eyesight, “20/20 Vision Activity – Eye Chart,” 2006.
[4] M. Bowen et al., “The Prevalence of Visual Impairment in People with Dementia (the PrOVIDe study): a cross-sectional study of people aged 60–89 years with dementia and qualitative exploration of individual, carer and professional perspectives,” Heal. Serv. Deliv. Res., vol. 4, no. 21, pp. 1–200, 2016.
[5] T. Saba, G. Sulung, and A. Rehman, “A Survey on Methods and Strategies on Touched Characters Segmentation,” Int. J. Res. Rev. Comput. Sci., vol. 1, no. 2, pp. 103–114, 2010.
[6] K. Vijayabharathi and V. Mahalakshmi, “Implementation of OCR Using Raspberry Pi for Visually Impaired Person,” Int. J. Pure Appl. Math., vol. 119, no. 15, pp. 111–117, 2018.
[7] D. Dimitrova, “Students with Visual Impairments: Braille Reading Rate,” Int. J. Cogn. Res. Sci. Eng. Educ., vol. 3, no. 1, pp. 1–6, 2015.
[8] L. A. Vader, “Measuring Vision and Vision Loss,” Nurs. Clin. North Am., vol. 27, no. 3, pp. 705–714, 2009.
[9] E. Ashrafi et al., “National and sub-national burden of visual impairment in Iran 1990–2013; Study protocol,” Arch. Iran. Med., vol. 17, no. 12, pp. 810–815, 2014.
[10] S. K. Singla and R. K. Yadav, “Optical character recognition based speech synthesis system using LabVIEW,” J. Appl. Res. Technol., vol. 12, no. 5, pp. 919–926, 2014.
[11] N. Jondhale and S. Gupta, “Reading text extracted from an image using OCR and android Text to Speech,” Int. J. Latest Eng. Manag. Res. (IJLEMR). ISSN 2455-4847, vol. 03, no. 04, pp. 64–67, 2018.
[12] H. Esmaecl, “Apply Android Studio (SDK) Tools,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 5, no. 5, pp. 88–92, 2019.