Read2Me: A Cloud- based Reading Aid for the Visually Impaired

Heba Saleous, Anza Shaikh, Ragini Gupta, Assim Sagahyroon
American University of Sharjah, UAE

Abstract — **For the visually-impaired segment of the population, the inability to read has a substantive negative impact on their quality of life. Printed text (books, magazines, menus, labels, etc.) still represents a sizable portion of the information this group needs to have unrestricted access to. Hence, developing methods by which text can be retrieved and read out loud to the blind is critical. In this work, we discuss the design and implementation of two assistive platforms, in one, we combine todays' smartphone capabilities with the advantages offered by the rapidly growing cloud resources, and the other utilizes a more economical approach making use of cost-effective microcontrollers. Both approaches make us of an Optical character Recognition (OCR) engine on the cloud and use local resources for the Text-to-Speech (TTS) conversion. Prototypes are successfully developed and tested with favorable results.**

Keywords—Visual Impairment, OCR, TTS, Smartphones

I. INTRODUCTION

Approximately 285 million people around the globe suffer from some sort of visual disability, with 39 million being completely blind. According to the World Health Organization (WHO) [1], 1.4 million blind individuals are minors under the age of 15, and 90% of people with impairments live in low and middle income countries. Therefore, visual impairment and finding feasible solutions to reduce the burden of it is a timely issue that requires the attention of researchers in industry as well as in academia. Furthermore, Today's technological advances provide an ideal and necessary base for finding optimal and cost effective solutions to this frustrating problem.

However, despite of the entrenched research efforts in this area, the world of print information such as newspapers, books, sign boards, and menus remain mostly out of reach to visually impaired individuals. Hence, in an effort to seek an answer to this persistent problem, an assistive technology-based solution, referred to in this paper as *Read2Me*, is developed and tested in the work presented here.

The project aims to implement a reading aid that is small-in size, lightweight, efficient in using computational resources , cost effective and of course user friendly.

II. RELTED WORK

To address the challenge described in the previous section, researchers have attempted to ease the burden on blind people by proposing various techniques that converts text to audible sounds. Tyflos [2] is a pair of glasses that had cameras attached to the side, earphones, and a microphone. Voice commands can be used to guide the user and direct the platform. Some commands include "move paper closer," "move paper up," "move paper up, right" from the device to the user, and "rewind paragraph," "forward paragraph," and "volume up" from the user to the device. However, the voice user interface might not function perfectly in a noisy environment, rendering it limited to indoor use. Finger Reader [3] is a wearable ring with a camera on the front. The user only needs to point at the text that they would like to read, and the Finger Reader will use local sequential text scanning in order to read each line of text progressively. This device is quite small, making it easy to carry around wherever the user goes. However, this device can give inaccurate results if aimed incorrectly, and it produces segmented audio output rather than a continuous audio string, which could confuse the user. In [4], the development of mobile applications to allow blind users to read text is discussed. OCR and TTS tools were integrated into an application in order to capture images and return audio as output back to the user. The Levenshtien Distance algorithm is used as a comparison tool between the text received after OCR has been done, and the original image. Authors experimented with three OCR tools: Tesseract, ABBYY, and Leadtools. ABBYY and Leadtools proved to be more accurate, each with approximately 18.8% of the median value of string distance while Tesseract had approximately 23.4% [4]. However, due to budget limitations for the project, Tesseract was used, since it is free, whereas Leadtools and ABBYY are commercial. The design of a microntroller-based prototype that converts images to audio is described in [5]. However, the prototype is only tested using large text images such as labels or large font titles on covers. A product Label Reader for the blind is discussed in [6]. A camera capture a video of the product then the captured video is split into frames. A text detection algorithm is then used to separate the text from sequence of frames. The OCR and TTS techniques are used to read the label back to the user. The application in [7] works by scanning the room using a wearable camera or the smartphone's integrated camera for QR codes placed on objects around the room. The scan occurs from the left side of the room to the right, and an audio output lists objects in three different ways using AT&T's TTS Demo. The app was tested with blind individuals to gather accurate opinions on their feelings towards the application and their uses and it received positive input.

978-1-4673-8743-9/16/$31.00 ©2016 IEEE

In context of the above discussed research and development attempts, clearly, a number of technical approaches have been embraced to develop an image-based application and a standalone device. However, most of the systems developed are built using expensive hardware components and are inefficient in text recognition when the conditions are not ideal. Moreover, most of the work has been done with the implementation of OCR engine on the local platform of mobile/standalone device such as PDA. This has led to certain limitations due to the limited hardware, power and memory resources of the phone. These applications also lack a user-friendly interface that can guide the visually impaired people to navigate through. In an effort to address these shortcomings we opted for two different approaches that capitalize and take advantage of cloud-based resources. In the subsequent section we details the design and implementation steps of the prototypes built in this work.

III. APPROACHES

An overall view of the two designs used to develop the Read2Me system is summarized below.

A. Approach 1: RPi-based Platform

The first proposed design of Read2Me consists of a Raspberry Pi 2 Model B (RPi) microcomputer, along with its compatible camera module. This camera module can be mounted on a number of accessories, such as glasses, or on a stand. The RPi will run Optical Character Recognition (OCR) on the image captured by the camera followed by Text-to-Speech (TTS) synthesis. An image of the text will be captured using the RPi's camera module and is then sent to an OCR cloud service. The text resulting from this process is then downloaded back onto the RPi and processed into audio on the device itself before being read aloud through speakers or a headset.

B. Approach 2: Android Application

Mobile phones are one of the most commonly used electronic gadgets today. Here, we intend to develop a modular and friendly application using cloud based OCR platform and the built in Android TTS for producing an audible result of the text file.

C. Selection of OCR and TTS Engines

OCR processing is CPU intensive. Therefore, in an effort to keep the power consumption of the RPi system and mobile application to minimum, and to increase the speed of the OCR, we selected to have the processing be done using the ABBYY Cloud OCR SDK. Its service is platform-independent due to the fact it is accessible through Web API and is not running on the device itself. So a Web API provided by ABBYY can be developed to run under any OS platform. However, the software is commercial and requires an internet connection. Since ABBYY incorporates pre-processing and post processing stages for the OCR-ed text, therefore it stands out as the most optimum platform for characters recognition. It eliminates the overhead cost of improving image quality before extracting text from the image. This software is not only limited to the recognition of the text documents but as well as barcode recognition, hand-printed text recognition, business card recognition and supports up to 198 recognition languages including French, English and German in varied font styles such as Normal (Ariel, Times New Roman or Courier), Gothic, typewriter text, magnetic ink characters and matrix [9]. Moreover, in [10], after running tests for the comparison of recognition accuracy, it was found out that ABBYY has an accuracy of 95.96% compared to 89.78% of Tesseract [10] an alternate OCR engine.

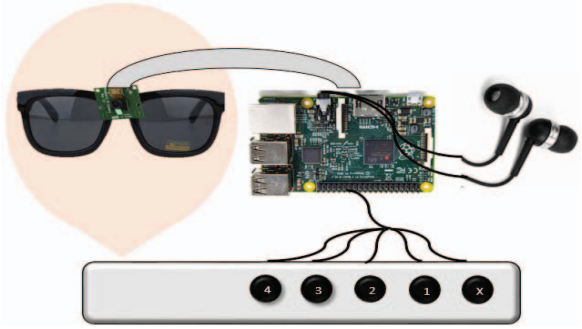


Figure 1: *RPi-based platform*

The TTS platforms considered for this work are eSpeak [11] which is a compact open source software speech synthesizer for English and other languages, for Linux and Windows. We also experimented with Festival [12] a free conversion software, however it's reproduced voice quality was substandard. After weighing options and pros and cons of available TTS tools, Pico TT [13] is used for the Raspberry-based design and an Android TTS [14] (released in version 1.6 of the Android platform) is used in the mobile based Read2Me application. The Pico engine produces quite good voices that sound natural. This engine supports up to five languages only but the quality of this engine outweighs other offline engines.The Android TTS is built-in in almost all the android devices. The TTS engine that ships with the Android platform supports a number of languages: English, French, German, Italian and Spanish [14]. No library needs to be installed on Android before using this. A simple Text-To-Speech Object needs to be created in the java code and its functions can be then used to use the object.

IV. SYSTEMS ARCHITECTURE

A. RPi-based Platform

1) System Overview

An overall view of the system is shown in Figure 1. A Raspberry Pi camera module will be attached to the pair of glasses allowing the user to adjust the head angles used to read text. The camera is attached on the bridge of the glasses in order to capture most of the text to be read. The earphones connected to the AV port of Raspberry Pi are available for the user to listen to the audio result. A simple design for the remote control being used can be seen in Figure 1. The X button will be used to turn off the system. The *1* button commands the camera module to capture an image. The *2* button replays the most recent audio file. The *3* button pauses or plays the audio currently being played. The *4* button allows

the user to alternate between the system's main (English) and secondary language (French). The push buttons on the clicker are debounced externally by connecting 10nF capacitors in parallel to the button and ground. Initially, the user will need training on how to use the system and maximize the benefits drawn.

The sequence of operations while using the system are as follows; the user switches the system on using the battery pack, if the user presses the Capture Button of the clicker, the camera gets activated and the user captures the image of the text he intends to read; the images are then captured and stored on the RPi memory. Next, it is transmitted to the ABBYY cloud for OCR; following a short delay, the equivalent text file is received back by the RPi, and TTS is applied on the received text file, and the audio output is next played into the ear piece. The hardware consists of a Raspberry Pi 2 Model B microcomputer , an 8 GB microSD card and a clicker as shown in figure 1 above. The operating system, images and audio results are stored on the card. The headwear components comprise of the Glasses that is attached with the Camera Module and the speakers/headset. The software main modules is comprised of Abbyy Cloud OCR, TTS Software and a Python Web API.

The Abbyy Cloud OCR software uses image processing technology to detect and recognize characters in digital text documents with a variety of qualities, including low-light, low-quality documents. It uses preprocessing to detect text orientation, correct an image's resolution, and remove texture from the image. This software will be utilized in our system due to these features and its ability to do all of its processing on a cloud system rather than the microcontroller itself. A TTS program is installed on the Raspberry Pi to convert the received text from the OCR software into audio. A web API in Python language will be used in order to allow communication between the Raspberry Pi and the Abbyy OCR cloud. The API will allow automated communication with the cloud, detecting the image file that has recently been added to the microcontroller's memory and sending it to Abbyy for preprocessing and conversion.

A Serial communication is used between the Raspberry Pi and the Camera Module (The camera plugs into the CSI socket on the Pi, using I²C for control.) And to access OCR service in cloud, the Wireless IEEE 802.11n protocol is utilized (The Wi-Fi dongle connects to the wireless network for OCR).

B. Read2Me Android Application

1) System Overview

Figure 2 illustrates an overall view of the system design flow for the Read2Me mobile application. The Read2Me application has been deployed for text recognition in two languages; English and German.

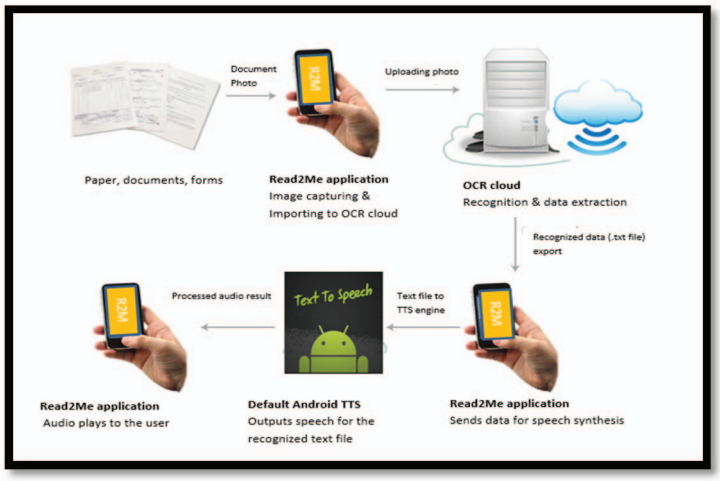


Figure 2: *System Architecture of Read2Me Application*

The application contains a simple user interface that is easy to use by the visually impaired. The smartphone screen is divided into two main menu buttons (Capture and Language) as shown in figure 3(a). They are designed to be large enough to cover the entirety of the Android phone's screen and play a sound that informs the user of the button's purpose when the button is pressed. Tapping the language button once will verbally tell the user that it will change the language to German if the current language is English ("Change language to German"), and vice versa. Tapping a button twice will lead to the execution of action.

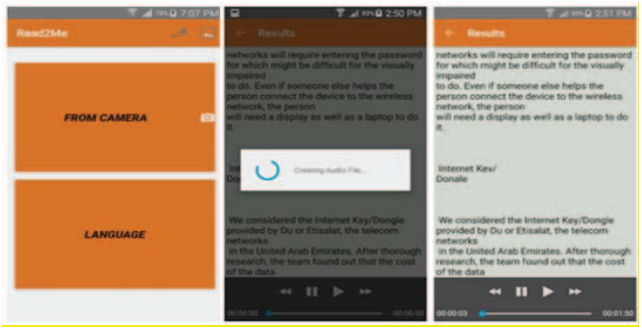


Figure 3: ***(a)*** *Main screen view of the application* ***(b)*** *Result of OCR* ***(c)*** *Text to be read out loud*

Once the capture button is tapped twice, a check for internet connection is done, if successful, the user will be informed that a picture of the text page is being taken and an autofocus capability of the phone camera starts to focus on the text area without any manual adjustment. After a timer of 5 seconds, the picture is taken and is automatically uploaded to the ABBYY OCR Cloud, where it is converted into text, and then downloaded back onto the Android device as shown in Figure 3(b). The downloaded text is then converted into audio using the Android phone's local TTS engine and then read aloud to the user. To control the audio being played out rewind, pause, play, and fast forward capabilities are added on the taskbar as shown in the taskbar of Figure 3(c).

The page photos translated and read by the application are stored in the phone's default photo gallery. Furthermore, and to optimize the functionality of the Read2Me system, *a stand* has been designed and built, as shown in Figure 4 below.

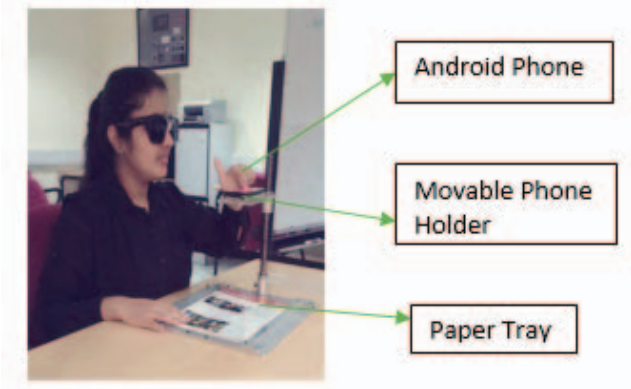


Figure 4: The Stand for Read2Me

The pages to be read are placed directly beneath a mobile holder as shown in the figure 4. . The mobile is placed on a holder directly above page tray. The user sits comfortably with his hands on the mobile phone touch screen and captures images followed by page flipping to advance to the next page guided by audio instructions.

The application only requires an Android phone as its hardware. In order to achieve a faster performance in running the application, we recommend a minimal specification of the device to be 400MHz and 128M RAM. The device must have 5MP as the minimal resolution of the camera in order to have a resolution of the captured image between 150dpi to 600dpi. The application is compatible with all the API levels 15 (Ice-cream sandwich) to 23 (Lollipop). The pacakgaes used are : Android Studio (IDE for Android Programming), Android local TTS and ABBYY OCR SDK Cloud. Android's notable features such as open source platform, multiple screen for multitasking, custom ROM, and open source libraries for Text-To Speech Synthesis superseded Android over other OS versions for the work presented here.

V. System Testing and Implementation Results

To assess the viability and usefulness of each approach , we considered the following measures: accuracy, latency, usability, power consumption , portability, weight and cost.

The text fonts used for testing are Times New Roman in 12pt, 14pt, and 16pt sizes. The application was installed on a Samsung Galaxy S4 for testing purposes, but has also been tried with the HTC One M8 model. The images in Figure 5 had been captured using the Samsung phone.

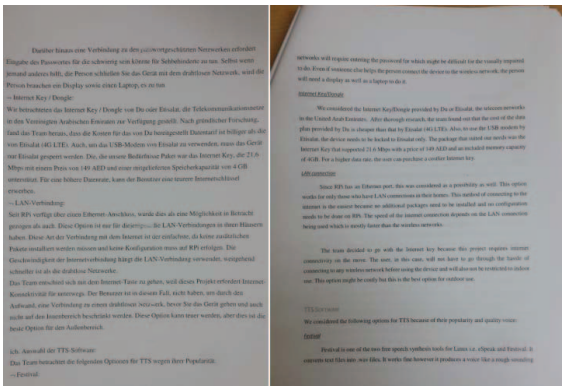


***Figure 5:** Text image captured by Android smartphone*

The images seen in Figure 6 were captured by the Raspberry Pi camera module.

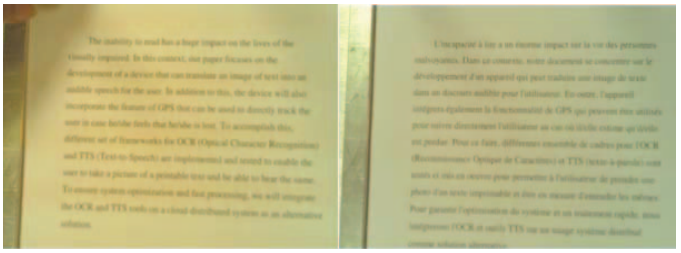


***Figure 6**: Images captured by the Raspberry Pi device*

A. Testing Read2Me on RPi

a) Accuracy

With size 16 font, the accuracy achieved in converting the image to text was approximately 99%. Due to the cropped shutter of the Raspberry Pi camera module, only a few letters were converted incorrectly. However, the audio file produced from this text file is almost perfect, with errors only in the pronunciation of some words or resulting from an error in the text file itself. However, as the font size is reduced, the accuracy level decreased, with approximately 60% with size 14 font, and 30% with size 12. Due to the worsening accuracy in text conversion, the audio conversion also began to produce inaccurate results of approximately the same degree. The formula used to calculate accuracy is:

*Accuracy %= (Number of words read correct/Total number of words) *100*

b) Latency

It is observed that different font sizes resulted in different times during the image-to-audio conversion process. Size 16 font took about 14 seconds from the moment the image began being converted into text to the moment the final audio result began to play out loud. With size 14, this process took about 20 seconds, while size 12 font took about 27 seconds for the whole process. Occasionally, size 12 font would return an error stating that the conversion process failed. One main factor of these times was the speed of the network the device was connected to. These tests are carried

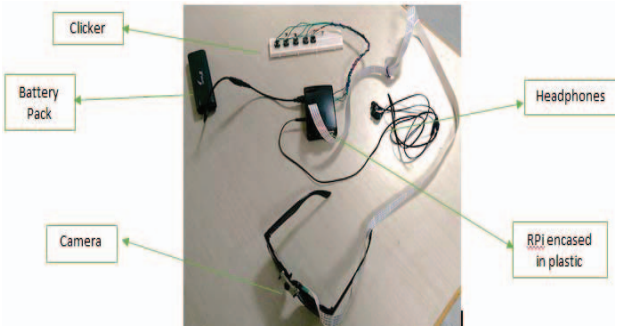
out at the American University of Sharjah, where the network speed was measured at approximately 30 Mbps. An image from the RPi had a size, on average, of about 500 kB. This means that the uploading process would take about:

$$Time = \frac{500x8x1x10^3}{30x10^6} = 0.133s$$

This time duration is considered when measuring the amount of time, it took for an image to be converted into audio.

c) Usability

The RPi device was encased in plastic to allow the user to easily handle it as shown in Figure 7 below.



.

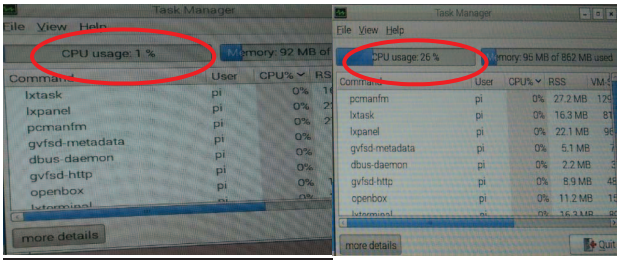
Figure 7: Components of Read2Me RPi System

The device can be controlled using a clicker as described earlier and whose buttons are distanced in a way such that the user would not press two buttons at the same time and would be able to distinguish between each one as they are numbered with 3D stickers.

To test the usability of the two approaches, 7 blind-folded people were asked to use the RPi-based platform and the android application and then fill in a questionnaire giving us their feedback. From the results, it was deduced that the RPi-based platform is more preferable in in terms of ease of use than the android application.

d) Power Consumption

The CPU usage was measured using the Raspberry Pi's task manager. As can be seen from Figure 8, the CPU was using 26% of its power during the text-to-audio process while taking about 3 – 6 MB of memory.



(a) *(b)*

***Figure 8**: CPU Power used (a) before and (b) during conversion*

e) Portability

The RPi device requires the glasses, the encased device, and the remote control to be carried around. The ribbon cable is flexible and can be rolled up. The glasses can be folded without damaging the camera. The remote control used for capturing images is thin enough to be carried without taking too much space. The dimensions of the Raspberry Pi are 85.60mm x 56mm x 21mm and the remote control measures 5.5 cm x 8.5 cm which makes the whole system very portable and easy to carry and operate. The Raspberry Pi weighs around 65g (including the case) and the breadboard weighs only 35g. Therefore, the total weight of the system is approximately 100g.

e) Cost

The Raspberry Pi 2 Model B costs 209 AED. The camera module, sold separately, can be bought online for 175 AED. Glasses and headphones, if not already owned, can be bought anywhere at any price depending on the quality. The battery pack used, without the batteries, costs about 116 AED. The case for the Raspberry Pi costs 39 AED. The five push buttons used for the remote control cost 50 AED altogether. The USB wireless adapter used to internet connectivity costs 57 AED. The total cost of this system is approximately 646 AED equivalent to approximately $175.

A free trial of OCR service was used therefore no cost for that has been considered. Moreover, the cost for the internet connection has also not been considered because the university's wireless network was used for testing. If OCR service and the internet connection was purchased, the cost for using them in both of the approaches will be the same and hence can be eliminated for the purpose of comparing.

B. Testing Read2Me on Android

a) Accuracy

The pictures (German and English) captured and shown in Figure 5, passed the OCR and TTS conversion process with an accuracy of 99.9%, with only 2 letters being misread because they were underlined (for example, y is mis read as v) or faded (and sometimes E is misread as F), as shown in Figure 5.

b) Latency

The application returned the text (in English) containing 334 words in 12 seconds. The text was 12pts Times New Roman. For German, the application took 13 seconds containing 327 words with 12 pts font. Creating the audio file takes about 5 seconds for both languages. Therefore, it could be deduced that the total time from capturing the image to hearing the speech output is approximatey17 seconds.

c) Usability

The application uses voice directives, as well as voice confirmations, to inform the visually impaired user of

what the application is currently doing, however playing/pausing the audio can be an issue here, since there are no voice labels for that and the blind person will possibly face some difficulty finding those buttons on the touch screen. Moreover, the blind person can also touch one of these buttons on the screen by mistake, but that will be easily detected by the user since on touching any of the buttons, the corresponding action will be committed.

d) Power Consumption

The application takes 10.39 MB of memory when installed. To check how much battery, the application uses up, the Smart Manager pre-installed on the Samsung S4 is used. Figure 9 contains screenshots from the Smart Manager which shows that the application, when used, only takes up 1% of battery and a RAM of 27.51 MB and CPU usage of 2.15%

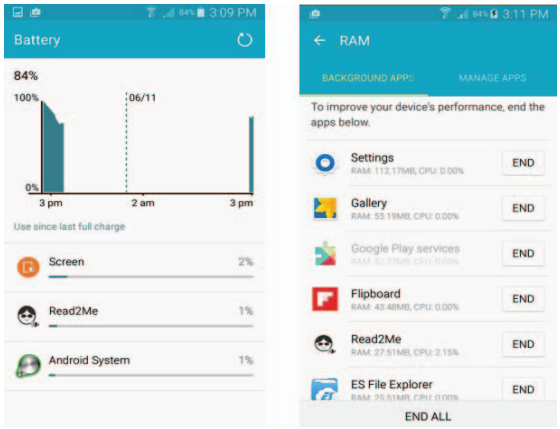


***Figure 9**: Power and RAM Consumption for Read2Me Application*

e) Portability

The application requires only a smartphone with an internet connection, today's smart mobiles are light in weight with the trend towards smaller and lighter models. The phone used for testing was Samsung S4 which only weighs 130 g.

f) Cost

The cost of Samsung Galaxy S4 varies depending on where it is purchased from but it ranges from 700-900 AED ($189 to $240). However, any reasonably equipped mobile for a cheaper price should be able to run the application. As with the Raspberry Pi device, a free trial of the OCR service is used. The university's wireless network was also used for this application. Therefore, their costs are not included in the estimate.

VI. Dicussion and Conclusions

We have discussed the design and implementation of two systems that add to the ability of the visually-impaired to share with us access to printed text regardless of its format. Both systems make use of the cloud as an economical and effective resource for character recognition. Interestingly, results of the questionnaire indicated that users found the microcontroller-based system is easier to use when compared to the mobile one. However, the accuracy of the mobile in the conversion efforts is better, primarily due to the high resolution camera built in the device. In future improvements of this work, the RPi-based system can be equipped with a high resolution webcam compared with the one used in this project, and we expect this will improve its accuracy. We predict more work will be produced in this critical area of assistive technology, and project that future portable gadgets will have easy to use and built in mechanism as reading aids for the blind, similar, to the mobile-based solution presented here.

References

[1] World health organization official website. August 2014. [Online]. Available at http://www.who.int/mediacentre/factsheets/fs282/en. Accessed on March 4, 2015.

[2] R. Keefer., & N. Bourbakis. 'Interaction with a Mobile Reader for the Visually Impaired'. 21st IEEE International Conference with Artificial Intelligence Tools. 18.03 (2009): 229-236. Web.

[3] R.Shilkrot.,&P.Maes.(2014,May.1).FingerReader: A wearable device to support text reading on the go.[Online]. Available: http://fluid.media.mit.edu/sites/default/files/paper317.pdf .

[4] R.Neto. &N.Fonseca. 'Camera Reading for Blind People'. Volume 11, 11.11 (2014) 1200-1209.[Online].Available at http://www.sciencedirect.com/science/article/pii/S2212017314003624

[5] Nagaraja L, et al, "Vision based text recognition using raspberry PI", *National Conference on Power Systems & Industrial Automation (NCPSIA 2015).*

[6] Nagarathna, Sowjanya V, M. "Product label readin system for visually challenged people", International journal of Computer Science and Information Technology Resreach, Vol 3, Issue 2, 2015.

[7] M. Jeon, A. Ayala-Acevado, N. Nazneen, B. Walker, O. Akanser, '"Listen2dRoom": Helping Blind Individuals Understand Room Layouts' in CHI '12 Extended Abstracts on Human Factors in Computing Systems, Austin, TX, U.S.A., 2012, pp. 1577 – 1582.

[8] "Tesseract-OCR," 2015. [Online]. Available: https://code.google.com/p/tesseract-ocr/. [Accessed 28 Oct 2015].

[9] Abbyy.technology, 'Supported OCR Text/Print Types [Technology Portal]', 2015. [Online]. Available: https://abbyy.technology/en:features:ocr:supported_ocr_text_types. [Accessed: 23- Oct- 2015].

[10] 'Smart Implementation of Text Recognition (OCR) for Smart Mobile Devices', The First International Conference on Intelligent Systems and Applications, pp. 19-24, 2012.

[11] "espeak text to speech," 2015. [Online]. Available: http://espeak.sourceforge.net/. [Accessed 25 Oct 2015].

[12] "ArchLinux: Festival," 13 Oct 2015. [Online]. Available: https://wiki.archlinux.org/index.php/Festival. [Accessed 25 Oct 2015].

[13] "Raspberry Pi," 05 Feb 2014. [Online]. Available: https://www.raspberrypi.org/forums/viewtopic.php?f=38&t=68693. [Accessed 28 Oct 2015].

[14] "Android Developers Blog," 23 September 2009. [Online]. Available: http://android-developers.blogspot.ae/2009/09/introduction-to-text-to-speech-in.html. [Accessed 28 Oct 2015].