Nateq Reading Arabic Text for Visually Impaired People

Omaimah Bamasag[1](✉), Muna Tayeb[2], Maha Alsaggaf[2], and Fatimah Shams[2]

[1] Faculty of Computing and Information Technology, Computer Science Department, University of Jeddah, P.O. Box 80221, Jeddah 21589, Saudi Arabia
obamasag@uj.edu.sa

[2] Faculty of Computing and Information Technology, Information Technology Department, Computer Science Department, King Abdulaziz University, P.O. Box 42808, Jeddah 21551, Saudi Arabia
{mtayeb0007,malsaggaf0009,fshamsulalam9}@stu.kau.edu.sa

Abstract. Nateq is a system developed to aid visually impaired people in their daily life tasks. Nateq allows blind users to read text written on papers and labels using their mobile phones. It uses two sources to read text from, either from camera or photo gallery. In the camera mode, the system will automatically capture the image once the object is sufficiently detected along with an option to capture the image of the object manually. To increase the accuracy, a novel approach was implemented to ensure the correctness of the extracted text, by adding rectangular boundaries detection to the system. It helps the user to avoid partial capturing of the object which may lead to extracting incomplete sentences. Testing on target users showed high level of satisfaction on the improvement made in the field of assistive application with an overall process being faster in comparison to similar applications in the market.

Keywords: Visual impairment · Image processing · Text extraction Boundaries detection · OCR · Accessibility · Reader assistant

1 Introduction

The decrease in vision ability that cannot be fixed by usual means, such as glasses is called Visual impairment and also known as vision impairment or vision loss [5]. The term blindness is used for complete or nearly complete vision loss. Visual impairment may cause difficulties for people to carry on their normal daily activities such as driving, reading, socializing and walking.

The World Health Organization (WHO) estimates that the number of visually impaired people in the world is 285 million, 39 million of them are blind and 246 million having low vision; 65% of visually impaired people and 82% of the complete blind are at the age of 50 years and older [11].

In the world of advanced technology, many tools have been invented to ease our lives. Disabilities field is one of the most significant aspects that attracted

M. Antona and C. Stephanidis (Eds.): UAHCI 2018, LNCS 10907, pp. 311–326, 2018.
https://doi.org/10.1007/978-3-319-92049-8_23

the attention of the researchers. It should conform to the development of current technology and meet the requirements of this digital age. To accomplish this goal, many technologies have emerged to improve the daily lives of people suffering from this impairment. One type of technology that is available nearly to everyone, is the use of mobile applications. In this project, we aim at developing a mobile application to assist blind people and improve the quality of their lives. This paper will illustrate and clarify the problems faced by people with this specific disability and the suggested solutions that our work will propose to address these problems. The rest of the paper is structured as follows: Sect. 2 presents the methodology followed for data collection and analysis. Section 3 describes the specifications of the system requirements. Section 4 demonstrates the implementation process. Section 5 presents the evaluation of the implemented system. Finally, Sect. 6 concludes the paper and outlines future work (Fig. 1).

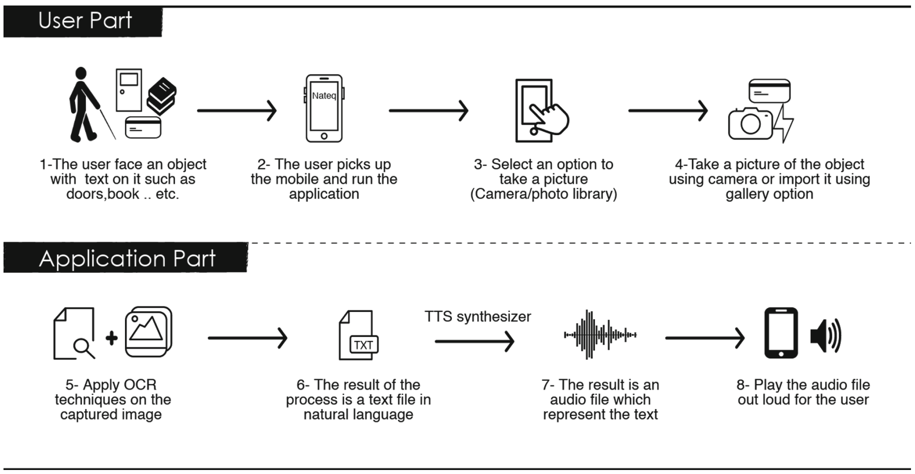

User Part
1-The user face an object with text on it such as doors,book .. etc.
Nateq
2- The user picks up the mobile and run the application
3- Select an option to take a picture (Camera/photo library)
4-Take a picture of the object using camera or import it using gallery option
Application Part
5- Apply OCR techniques on the captured image
TXT
6- The result of the process is a text file in natural language
TTS synthesizer
7- The result is an audio file which represent the text
8- Play the audio file out loud for the user

Fig. 1. Proposed process of the system

2 Related Systems Analysis

This section reviews well-known software solutions that are most related to the proposed project.

2.1 KNFB Reader

The KNFB Reader [10] is capable of reading different types of documents that one might face in daily life. The application uses VoiceOver on iOS and Google TalkBack on android to assist visually impaired users and guide them vocally while using the application.

2.2 Amedia Live Reader

Amedia Live Reader [12] takes the scan of a captured live image and reads any texts in it in real-time. It is designed for visually impaired users. it uses the VoiceOver on iOS platform as a guidance.

2.3 Google Translate

Google translate [7] is a mobile application running on both iOS and Android that translates between many languages. It provides instant camera translation using phone camera and for higher-quality translations, it takes picture of text.

2.4 Acapela TTS Voices

Acapela TTS [1] turns written text into speech and gives you the advantage of buying and installing voices as well as integrating them in your Android device in order to be used with the system or any TTS compatible applications such as Google TalkBack.

2.5 Adel TTS Voice

The Best-of-Vox [14] the application gives you the advantage of typing or pasting text and having it read out loudly by the sound that you have chosen.

2.6 Text Fairy (OCR Text Scanner)

TextFairy [15] turns a scanned image of a document to text and correct the viewpoint of the image. It has the facility to edit extracted text and copy it into the clipboard for utilization in other apps. It can also transform the scanned page into a PDF form.

2.7 CamScanner

CamScanner [6] helps in scanning, storing, syncing and collaborating on numerous contents across smartphones and computers. It uses the phone camera to scan documents and provides text extraction from Image with OCR (optical character recognition) for further editing or text sharing (Table 1).

KNFB Reader is a good multi language OCR reader, however, it does not support Arabic language as Amedia Live Reader. Amedia Live Reader has a real-time OCR feature but difficult to use because it repeats reading the captured text when the phone camera moves. Text Fairy and CamScanner perform the OCR without the TTS feature and both do not support the Arabic language. CamScanner has the feature of edge detection which is an important feature for our proposed app. Google Translate has a good OCR that supports Arabic language and its API is available in the Google Cloud Platform services.

Table 1. Summary comparison between related systems

Systems	Cost	Provide OCR	Provide TTS	Platform	Support Arabic Language	Supports Blind User
KNFB Reader	Not free	Yes	Yes	iOS and Android	No	Yes
Amedia Live Reader	Not free	Yes	Yes	iOS only	No	Yes
Google Translate	Free	Yes	Yes	iOS and Android	Yes	No
Acapela TTS voices	Free	No	Yes	Android only	Yes	Yes
Adel TTS voice	Not free	No	Yes	Android only	Yes	Yes
Text Fairy	Free	No	No	Android only	No	Yes
CamScanner	Free with licensed	Yes	No	iOS and Android	No	No
Nateq System	Free	Yes	Yes	iOS	Yes	Yes

3 Data Collection and Analysis

This section describes the data collection process to identify the needs of the potential users. The data was collected from 30 visually impaired persons in both genders. Their ages range from 18 to over 50 years old.

3.1 Data Collection Methods

Questioners and interviews have been conducted with visually impaired people and heads of departments from the Governmental foundation for rehabilitation and visual impairment in Jeddah, Saudi Arabia and Special Needs Center in students affairs of King Abdulaziz University.

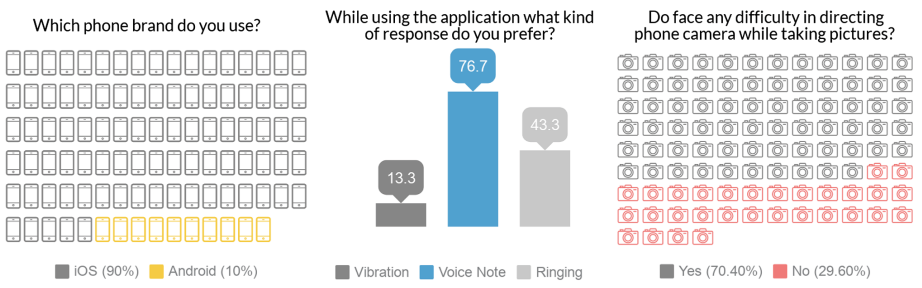


Fig. 2. Main questionnaires results

3.2 Questionnaires Results

Questionnaires results (Fig. 2) show that 90% of the volunteers use iPhone. Most volunteers prefer voice note feedback and about 70% have a problem in directing camera toward objects. The gathered data from the volunteers is considered as the main requirements of the system.

3.3 Interview Results

The interviews highlighted some findings supported by the questionnaire results, which will be considered in the requirements specification:

1. There are no good existing quality apps that support reading Arabic text through the phone camera. Also, there are inaccuracy issue in extracting the Arabic text from scanned PDF documents.
2. The reason for the domination of the iOS platform for the hand-held over all other platforms is that there is a good quality built-in accessibility features.
3. The visually impaired people are very good in using hand-held devices.
4. There are some levels of visual impairment where people can see using special techniques and devices.
5. The difficulty of learning braille system results in resistant to use this system especially for those who lost their vision in late ages.

4 Requirement Specification

Functional non-functional requirements for the proposed system were specified by using the data gathering methods illustrated in Data Collection and Analysis section.

4.1 Functional Requirements

The systems functional requirements have been categorized as User Manual, Input, Processing, Output, Feedback and Help.

User Manual

1. The system shall play the audio user manual file automatically the first time the user opens the application.
2. The system shall allow the user to replay the audio user manual upon user request.
3. The system shall allow the user to navigate through the application.

Input

1. The system shall allow the user to automatically take a picture by the hand-held device camera.
2. The system shall allow the user to manually take a picture by the hand-held device camera.
3. The system shall allow the user to import a selected photo from the mobile gallery.

Processing

1. The system shall be able to detect the edges of the object in real time in order to auto capture the required object.
2. The system shall be able to detect and extract the printed Arabic text in the captured image.
3. The system shall keep the user aware of the progress while extracting the image by voice notes.

Output

1. The system shall be able to play the extracted text using VoiceOver iOS built-in Screen reader.
2. The system shall allow the user to replay the extracted text.

Feedback and Help

1. The system shall allow the user to read a new object after a task completion.
2. The system shall allow the user to change the speed of the reading inside the application.

4.2 Non-functional Requirements

This section illustrates the non-functional, i.e. quality, requirements of the proposed system, which are categorized into response time and usability.

Quality Requirements

1. Response time:
   The System OCR gives the desired response within a reasonable time.
2. Usability:
   (a) The system shall be easy to use for the visually impaired after listening to the user manual.
   (b) All functions of the system shall have audio responses to facilitate the app usage.
   (c) The system interfaces shall be minimized and support accessibility for visually impaired.

5 Implementation

This section presents the algorithms developed to implement the required functionalities of the proposed system. This is followed by listing the technologies used in the implementation and a walk-through the system.

5.1 Developed Algorithms

Several algorithms were developed to implement the requirements gathered from end users. Three main algorithms were used for text extraction, frame extraction and boundaries detection, explained in the following subsections.

Text Extraction

Algorithm 1. Extracting Text from a Given Image

```
Input: Image.
Output: Plain text.
 1: Convert the JPEG image to PNG
 2: Check the size of the picked image before sending it to the API server.
 3: if the image is larger than the server's request limit 4MB then
 4:     Shrink the image size to fit the request limits.
 5: end if
 6: Encode the PNG image to base-64 string.
 7: Use the Alamofire library to:
      1. Specify the request parameters and embed the converted image string within
         the request.
      2. Create a custom HTTP header.
      3. Send the POST HTTP request to the Google API server included with the
         API key to authenticate the request plus the customized HTTP header.
 8: Get the server response and check
 9: if the returned response is an empty message then
10:     Play a voice over a message that there is no text has been detected from the
    picked image.
11:     End the call and return to the main application screen.
12: end if
13: Use the SwiftyJSON library to serialize the JSON server response and put it in
    Swift array.
14: Iterate the response array and get the extracted text.
15: open a new screen that displays the extracted text centered and large formatted.
```

Boundaries Detection Identifying rectangular objects within an image captured by the user camera was the core of our project. Below is the approach we followed to implement this feature.

Figure 3 shows the intermediate results generated from the algorithms mentioned in the pseudocode, while Fig. 4 shows the final result.

Frame Extraction. Camera Auto Capture captures the frames in real time through the mobile camera. These frames are the input for Boundaries Detection algorithm. The boundaries algorithm processes the captured frames to find the rectangular boundaries of the object, which triggers the auto capture event. To implement this functionality, several packages and algorithms were used as illustrated in the following.

Algorithm 2. Detecting Rectangular objects Algorithm

```
Input: Image frame.
Output: Image with border around the rectangle if any.
 1: Convert the image frame to matrix.
 2: Convert matrix's color space to grayScale.
 3: Apply median filter on the grayscale matrix.
 4: Apply Canny algorithm on the blurred image.
 5: Apply morphological transformations for lines closing.
 6: find contours and return in 2D array.
 7: Sort contours descending by area.
 8: for each contour do
 9:     Approximate the contour to a polygon by Douglas-Peucker algorithm.
10:     if (approximated polygon has 4 points and it's Convex) then
11:         Draw a border around the rectangle detected.
12:         break
13:     end if
14: end for
15: Convert matrix back to image.
16: Return image.
```

Algorithm 3. Capture Frames in Real Time

```
Input: Camera permission Granted.
Output: Camera frame.
 1: Choose the video capture device.
 2: Set the device camera to backward facing camera.
 3: Set focus mode to continuous Auto Focus.
 4: Create a capture session to coordinate the data flow from the input to the output.
 5: Preset the capture session with quality 1024*760.
 6: Use Capture Inputs to provide input data to a capture session.
 7: Use Capture Outputs to Get Output from a Session.
 8: Set capture output's buffer as serial queue for the transmitted video frames.
 9: Add both Capture Inputs and Capture Outputs to the capture session.
10: Start the capture session.
11: while The session is running do
12:     Add The captured data to the buffer.
13:     Convert the data to an image.
14:     Return the image as a camera frame.
15: end while
```

5.2 Technologies Used

Google Cloud Vision API provides powerful Image Analytics capabilities as easy to use APIs. It enables application developers to build applications that can see and understand the content within the images. The service enables customers to detect a broad set of entities within an image from everyday objects to faces and product logos [8].

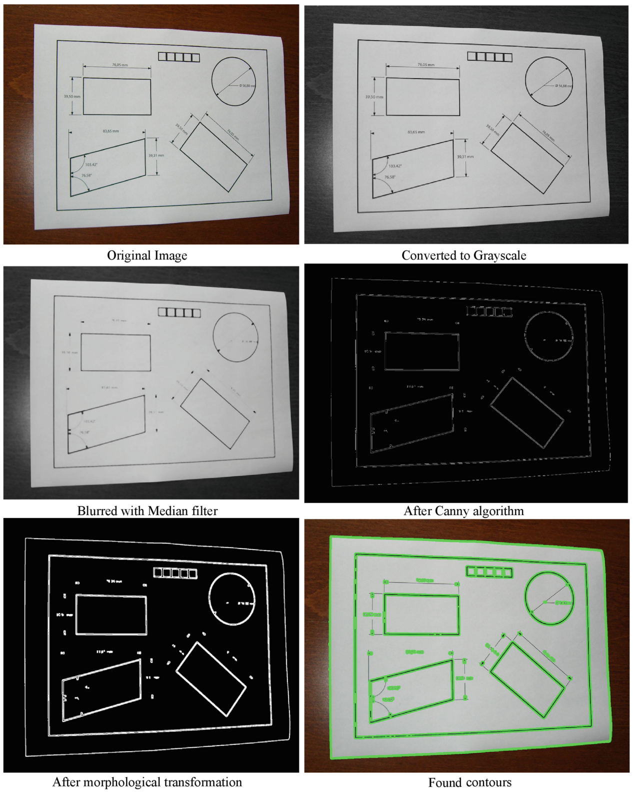


Fig. 3. Detecting paper's boundaries

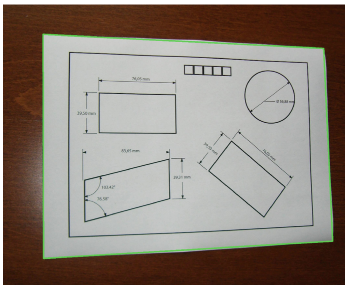


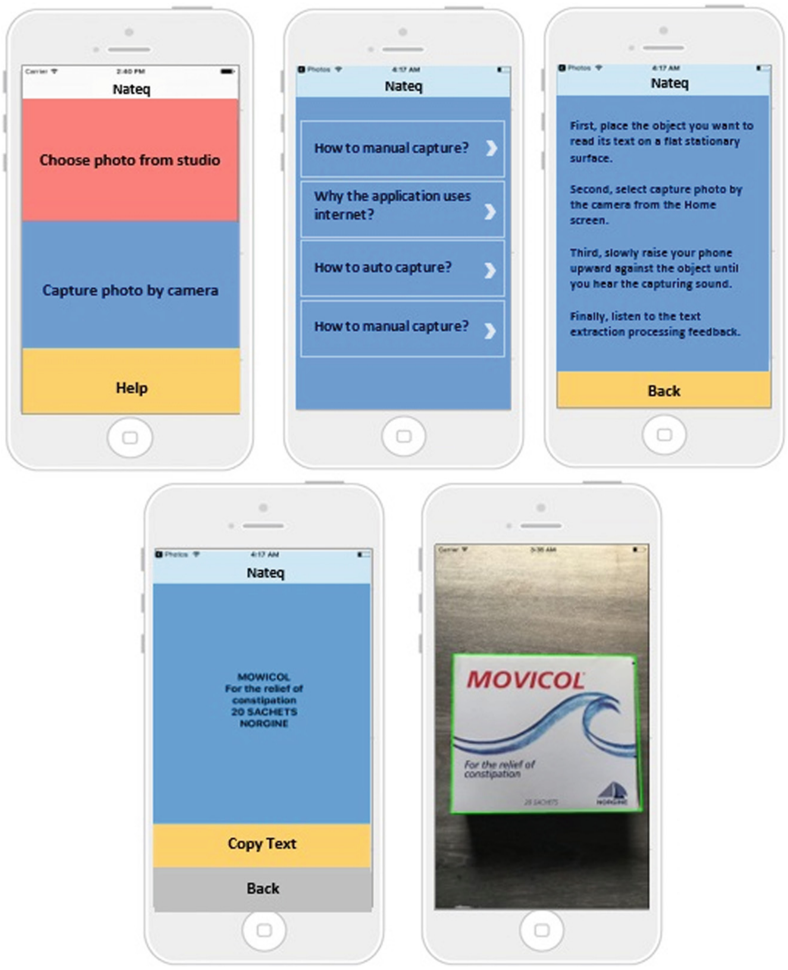
Fig. 4. Detecting boundaries final result

OpenCV is an open source computer vision and machine learning software library mainly aimed at real-time computer vision applications and its originally developed by Intels research center [13].

Voice Over is a screen reader built into Apple Inc.s macOS, iOS, tvOS, watchOS operating systems. By using VoiceOver, the user can access their Macintosh or iOS device based on spoken descriptions. The feature is designed to increase accessibility for blind and low-vision users [4].

5.3 Walk-Through the System

See Fig. 5.



First, place the object you want to read its text on a flat stationary surface.

Second, select capture photo by the camera from the Home screen.

Third, slowly raise your phone upward against the object until you hear the capturing sound.

Finally, listen to the text extraction processing feedback.

MOWICOL
For the relief of
constipation
20 SACHETS
NORGINE

Fig. 5. Application main screens

6 Evaluation

The components of Nateq system are tested by applying different measures and testing approaches. The testing process is divided into two sub-tests: System test which is conducted on the application itself to examine the resources and time consumption, and the usability test which is conducted on the applications end users to observe and record their interaction with it. The data gathered from both sub-tests are considered for the applications reliability and satisfaction in terms of functions and user experience.

6.1 System Testing

The system testing examines the applications consumption levels of resources, the efficiency of the boundaries detection algorithm, the speed and accuracy of the text extraction operation.

Resources Consumption Test. Table 2 shows the applications resources consumption according to different applications status.

Table 2. Resources consumption

	Idle state	Edge detection state	Text extraction state
CPU	0%	46%	1%
Memory	5.2 MB	7.7 MB	15.1 MB
Energy	0	Low	High
Network	-	-	Up less than 2 MB
			Down depends to text in the image

When the application launches, it takes about 5.2 MB of the phones memory. When camera works, it processes the frames which results in an increase in the apps memory allocation to 7.7 MB while the CPU usage is about 46%. When the captured image is sent to the text extraction server, the application requires a network connection to upload the image to the server where the image size should not exceed 2 MB and the downloading amount depends on the text written in the captured image. Regarding the battery consumption, the energy required will be high because of the network overhead required to establish the connection [2].

Boundaries Detection Test. Two testing approaches were applied on Boundaries Detection algorithm, performance and speed testing.

Performance Testing: The ability to identify and detect the boundaries of rectangular objects were tested with different background situations (modes) Includes:

1. Test detection in solid color background with high contrast between the background and the object colors.
2. Test detection in solid color background with low contrast between the background and the object colors.
3. Test detection with wobbly (Textured) background.

The evaluation is performed by applying 30 different sample images. Table 3 shows the testing results and success rate along the previously mentioned three cases.

Table 3. Boundaries performance testing results

Background mode	Expected result	Success rate	Performance
Solid - high contrast	Detect	100%	Excellent
Solid - low contrast	Detect	40%	Not accepted
Textured	Detect	60%	Needs enhancement

According to the results shown above, following is the constraints on the boundaries detection:

1. The contrast between the background and the object must be high for better detection.
2. Solid color backgrounds provides better detection rate.
3. Reflected light on the object surface decreases the detection rate.

Speed Testing: The average execution time for the algorithm is **0.05** s, which is considered good compared to Apples CIDetector API [3], which detects rectangles in **0.03** s. The execution time for our algorithm is calculated by running the algorithm in real time and taking the average of 500 different readings.

OCR Test

Response Time Testing: The speed of the OCR provided in the application is tested under different Internet speed. Table 4 shows (in seconds) the effect of the Internet provided on the device in the OCR processing speed.

Table 4. OCR response time in seconds

Measure	3G	4G	Fiber optics
Short text/cover page	16	6	4
Long text/full page	36	25	16

Extraction Accuracy Testing: The accuracy is tested by applying the OCR on 40 images as a sample size with different fonts styles. The testing is divided into to categories; Testing on document fonts (books, papers, etc) and products fonts (food products, cleaning products, medicines, etc). Table 5 shows the results in each category.
Observations:

1. Google OCR gives the best results when applied to documents/Books fonts.
2. It cannot detect stretched Arabic words.

Table 5. Accuracy testing results

Font type	Word count	Wrong words	Undetected words	Success percent
Documents fonts	550	32	16	91%
Products fonts	150	26	32	61%

6.2 Usability Testing

The goal of conducting the usability test is to evaluate Nateqs interface and its functions on real users. The test will help to determine whether the application design is usable for the user who uses it for the first time. The users feedback which will be gathered during the test will help the team to improve the application.

Tasks to be Evaluated. Six tasks were chosen to test Nateq functionality. Each of which was piloted to determine the suitable performance measures used for each task. An iPhone- expert visually impaired user was timed when doing these tasks and provided us with a baseline to judge the times that participants would take.

1. **Task 1:** Access the user manual and navigate through the questions.
2. **Task 2:** Get Text from a photo saved in photo gallery.
3. **Task 3:** Repeat reading the extracted text.
4. **Task 4:** Reach copy text button.
5. **Task 5:** Get text from a photo captured using Manual Capture
6. **Task 6:** Get text from a photo captured using Auto Capture

In addition to task completion time, the number of navigation clicks and the number of (action) clicks were added as a measurement. On accessibility mode, users flick (swipe) to the left or right to move to the next item on screen and single-tap to hear a description of what is tapped, these counts as navigation clicks. On the other hand, users double-tap on an item to open or activate it, these counts as action clicks.

The following tables (Tables 6, 7 and Table 8) indicates the tasks performed and measurement levels used for each task.

Testing Results for five participants (Tables 9, 10, 11, 12, 13 and 14).

Table 6. Excellent performance

Measure	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Time to complete the task	<23 s	<45 s	-	-	<26 s	<40 s
Number of navigation clicks	<13 clicks	<25 clicks	1 click	1 click	<5 clicks	<6 clicks
Number of clicks	<4 clicks	<4 clicks	-	-	<3 clicks	<3 clicks

Table 7. Acceptable performance

Measure	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Time to complete the task	23–31 s	45–55 s	-	-	<26–34 s	<40–60 s
Number of navigation clicks	13–21 clicks	25–34 clicks	2–3 clicks	2–3 clicks	<5–12 clicks	<6–9 clicks
Number of clicks	4–6 clicks	4–6 clicks	-	-	3–5 clicks	<3–5 clicks

Table 8. Unacceptable performance

Measure	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Time to complete the task	>31 s	>55 s	-	-	>34 s	>60 s
Number of navigation clicks	>21 clicks	>34 clicks	>3 clicks	>3 clicks	>12 clicks	>9 clicks
Number of clicks	>6 clicks	>6 clicks	-	-	>5 clicks	>7 clicks

Table 9. Task 1: Access the user manual and navigate through the questions

Measure	1	2	3	4	5	Performance
Time to complete the task	21	37	25	16	23	Accepted
Number of navigation clicks	9	15	12	10	12	Excellent
Number of clicks	2	2	2	2	3	Excellent

Table 10. Task 2: Get text from a photo saved in the studio

Measure	1	2	3	4	5	Performance
Time to complete the task	53	40	77	53	55	Accepted
Number of navigation clicks	23	33	90	30	19	Accepted
Number of clicks	4	2	3	2	3	Excellent

Table 11. Task 3: Repeat reading the extracted text

Measure	1	2	3	4	5	Performance
Number of navigation clicks	1	1	1	1	2	Excellent

Table 12. Task 4: Reach copy text button

Measure	1	2	3	4	5	Performance
Number of navigation clicks	2	1	2	2	2	Accepted

Table 13. Task 5: Get text from a photo captured using manual capture

Measure	1	2	3	4	5	Performance
Time to complete the task	20	25	19	30	24	Excellent
Number of navigation clicks	7	9	15	10	12	Accepted
Number of clicks	2	4	2	1	2	Excellent

Table 14. Task 6: Get text from a photo captured using auto capture

Measure	1	2	3	4	5	Performance
Time to complete the task	56	6	120	120	94	Not accepted
Number of navigation clicks	7	9	3	10	11	Accepted
Number of clicks	1	2	1	1	2	Excellent

According to the previous tables, task 1 to task 5 present acceptable results by taking all participants average performance per each task. While task 6 demonstrated some difficulties faced when using the camera. As per users feedback, more instructions were needed to explain how auto capture works.

7 Conclusion

The paper is concluded by listing the findings of project work and testing on the target users:

1. There is no adequate software that guide blind people in directing the camera toward objects while taking photos.
2. The need of descriptive photo gallery, as blind people face difficulties in finding photos, which are currently labeled only by the date of creation.

These could be future research directions in the field of special needs aid.

References

1. Acapela TTS. http://www.acapela-group.com/. Accessed 15 Nov 2016
2. Apple Inc.: General Battery Information. https://www.apple.com/sa/iphone/battery.html
3. Apple Inc.: CIDetector. https://developer.apple.com/reference/coreimage/cidetector. Accessed 20 May 2017
4. Apple Inc.: Apple VoiceOver. http://www.apple.com/accessibility/mac/vision/. Accessed 20 Nov 2016
5. Blindness and Vision Impairment, 8 February 2011. http://www.cdc.gov/healthcommunication/ToolsTemplates/EntertainmentEd/Tips/Blindness.html. Accessed 10 Nov 2016
6. CamScanner. https://www.camscanner.com. Accessed 15 Nov 2016

7. Google corp. Google Translate. https://play.google.com/. Accessed 20 Nov 2016
8. Google Cloud Platform. cloud.google.com/vision/docs. Accessed 17 Mar 2017
9. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
10. Sensotec: KNFB Reader. http://www.knfbreader.com/. Accessed 15 Nov 2016
11. World Health Organization: Global Data on Visual Impairments. Silvio P. Mariotti, Geneva (2010)
12. Mochizuki, Y., Mochizuki, T.: Amedia Live Reader. http://www.amedia.co.jp/english/product/iphone/livereader. Accessed 15 Nov 2016
13. OpenCV: OpenCV Library (2011). docs.opencv.org/2.4/doc/. Accessed 16 Mar 2017
14. VOXYGEN: Adel TTS. https://best-of-vox.com/android (n.d.)
15. Wellnitz, R.: Text Fairy (OCR Text Scanner). https://play.google.com/. Accessed 15 Nov 2016.