Journal of Positive School Psychology
2022, Vol. 6, No. 3, 6081–6088 http://journalppw.com

# Low-Cost Smart Glasses for Blind Individuals using Raspberry Pi 2

Gracie Dan V. Cortez [1], Jan Cesar D. Valenton [2], Joseph Bryan G. Ibarra [3]

*[1,2,3] School of EECE , Mapúa University , Manila, Philippines*
*Email: [1] gdvcortez@mymail.mapua.edu.ph, [2] jcdvalenton@mymail.mapua.edu.ph,*
*[3] jbgibarra@mapua.edu.ph*

**Abstract**

Blindness is the inability of an individual to perceive light. Due to the lack of visual sense, blind individuals use guiding tools and human assistance to replace their visual impairment. The study developed a smart glass that can detect objects and text signages and give an audio output as a guiding tool for blind individuals. In creating the prototype, Raspberry Pi 2 Model B will use as the microprocessor. It will be using a camera module that will be a tool for detection. The algorithms used for object detection and text detection are YOLOv3 and OCR, respectively. In-text detection, OCR helps recognize both handwritten and digitalized texts. MATLAB is the software used for the application of OCR. It is composed of three parts (3): image capturing, extraction of text, and conversion of text-to-speech. In object detection, YOLOv3 is the algorithm used in the process. It comprises four (4) parts: data collection, data preparations, model training, and inference. Then the conversion of text-to-speech will take into place. The objects that the prototype can detect are limited to 15 objects only. The prototype can function at both the 150 lux luminance and 107527 luminance in object detection. However, there are discrepancies in the detection of some objects due to distance; the detection cannot detect the specific thing at certain trials. In-text detection, the detection of the text signage has 100% reliability. In addition, text detection used five font styles. In the testing, the font style Calibri has a 30% percentage error (using the word ENTRANCE) and a 20% percentage error (using the phrase EXIT) due to the structure. The processing time of the prototype has an average time of 1.916s at maximal walking and 1.673s at a slow pace walking.

**Keywords**— blindness, Raspberry Pi 2, Raspberry Pi Camera Module v2, YOLO v3, OCR, object detection, text detection

## I. INTRODUCTION

The eyes allow humans to determine objects through light and process the gathered details in the brain. An individual with no perception of light is considered total visual impairment or, in general terms, blind. These visually impaired individuals, most especially the blind, depend on the human body's other senses. In addition, some of them use human assistance, guiding tools, or well-trained animals to replace their visual impairment. The researchers aim to design a low-cost smart glass that detects objects and even text signages that impact blind individuals.

Different technological advancements and new devices emerged, introducing a navigation tool for visually impaired individuals, including smart sticks and smart glasses. [3] Various researches support the creation of low-cost navigation tools with unique features and different algorithms such as YOLO and OCR. Smart glasses are an aid for blind individuals. It will serve as a navigation tool to guide them because it can detect objects and English text signages. It will alert the individual through an earpiece by text-to-speech.

This research aims to create a low-cost prototype smart eyeglass that can help the blind. It will detect objects and determine different text signages seen in an environment. The objects that the prototype can detect are limited to 15 objects since the prototype is programmed to detect objects in the living and garage space. The prototype can only acknowledge English text signages. It cannot recognize other signages such as logos, symbols, and figures. It is only limited to signages whose edges are sharp, specifically those that are red-bordered, rectangular, and square in shape. Moreover, it will not consider other types of visually impaired individuals. The prototype is dedicated only to blind people. It cannot give specific directions of the wearer to the detected object.

## II. METHODOLOGY

This chapter incorporates the overall setup of the system, from the components to the algorithms to be used to design a low-cost wearable smart glass for blind individuals. The methodology extensively discusses the conceptual framework, system process flow, simulation process, and testing process.

### *2.1 System Process Flow*

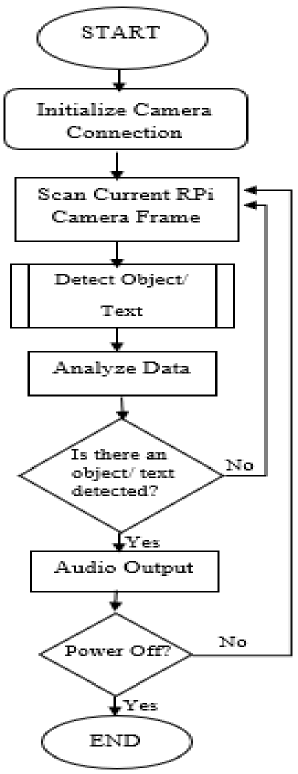


***Figure 2.1 System Flow Diagram***

Figure 2.1 represents the system flow diagram of the smart glass that signifies the building blocks of the system. Once the system starts by pressing the button from the power bank, the camera will initialize for connectivity. It will then scan the current Raspberry Pi camera frame to detect the object or text. Once the system detects an object or text, it will analyze the data. The processed image will be converted into an audio output if the analyzed data is successful. On the other hand, if it did not successfully analyze the data, the system will repeat the process from scanning the current raspberry pi camera frame. Once the system process is complete, the user will decide if the power will be turned off by pushing the button from the power bank or continuing the process.

### *2.2 Block Diagram*

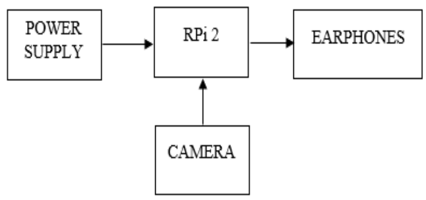


***Figure 2.2 Block Diagram of the System***

Figure 2.2 shows the components' connections that are relative to each other. The power supply comes from the power bank connected to the Raspberry Pi 2. The Raspberry Pi Camera Module v2 connected with the Raspberry Pi 2 that are both pasted into the frame of an eyeglass so that when it is worn, it could recognize on the same level of what ordinary people see. The programmed data will then have the recognized subject as audio output. The user, through an earphone, will hear the audio output.

### *2.3 Methodology for the Objective*

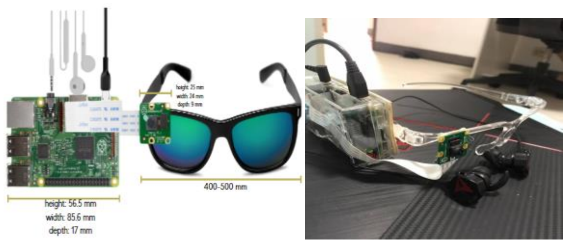

***Figure 2.3 Prototype Setup***

Figure 2.3 shows the system and actual prototype setup. The prototype setup is composed of the following components: glasses, Raspberry Pi camera module v2, Raspberry Pi 2 Model B, and earphones.

### *2.3.1 Object Detection*


***Figure 2.4 Object Detection***

For object detection, YOLOv3 is the algorithm used for the prototype. Figure 2.3 shows the process for the YOLO algorithm. This algorithm divides an image into grid cells. The cells represent predictions for the five (5) bounding boxes, where these boxes are the rectangles enclosing an object detected.

There are four (4) parts for object detection: data collection, data preparations, model training, and inference. The data collection process uses images of objects in the living room and garage space to create the dataset. In this process, things from the living room and garage space will use 5675 images to create the dataset. It will be labeled and applied with annotations.

The captured images will undergo the process of data preparation. Data preparation is when the annotations will be subjected to corrections, resizing, and updating of the images if the annotations match the photos and the color corrections of the pictures. In the model training process, YOLOv3 is the algorithm to use. Lastly, the inference is the application of the trained models to make predictions. This process will take place by setting up the YOLOv3 architecture and the custom weights introduced with the architecture.

### *2.3.2 Text Detection*

***Table 2.1 Text Detection Process***

| | |
|---|---|
|   | This is the initial part of the text signages extraction, the text will be captured live using the Raspberry Pi Camera Module v2. |
|   | Binarization is the process of converting the image captured from the camera module into grayscale for the program to easily recognize the red border format. |
|   | A program will be coded in MATLAB using the color thresholding code whereas the red border will be recognized by the system. |
|   | Hough transform is a system that recognizes the lines that are present in the binarized image. It is introduced through the equation represented in Equation: $x \cos\cos\,\theta\, + y \sin\sin\,\theta\, = r$ |
|   | Extraction is the last stage of text signage extraction. This process includes image cropping, enhancement, and background subtraction. |
| Text-to-Speech  | Text-to-speech transcription is used as third-party APIs from MATLAB. It is the conversion of extracted text into an audio output. The OCR algorithm will be called to save the processed image into the desired format for the conversion of the filet. After that, the software installed converts the saved text file into an audio output. |

## III. RESULTS AND DISCUSSION

This chapter presents the results and discussion of the test result of functional testing of the object detection and text detection and the accuracy testing of both object and text detection.

### *3.1 Functionality Testing*

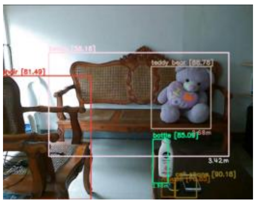

***Figure 3.1 Functionality testing***

Shown in Figure 3.1 is the functionality testing conducted for the prototype. The functionality testing is halved into three parts: object detection functionality testing, text detection functionality testing, and functionality testing of both the text and object detection. For each functionality testing, there are ten (10) trials conducted. Moreover, the gathered data will be computed using the reliability percentage formula.

$$Reliability\% = \frac{Number\ of\ successful\ trials}{Total\ number\ of\ trials} X\ 100 \quad (1)$$

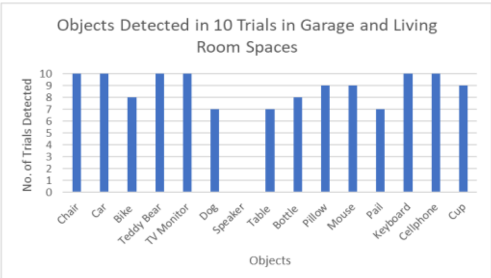


***Figure 3.2 Objects Detected in the Garage and Living room space***

Shown in Figure 3.2 is the data for the objects detected in both the garage and the living room space. The testing of the prototype, Raspberry Pi 2, will use a camera that relies on the luminance of the place. The testing for the detection of the objects will conduct at 150 lux luminance at the living room space and 107527 lux luminance at the garage space. The data shown in Figure 3.2 shows object detection. This figure shows the objects detected by the prototype from the actual setup in both the living room and garage space. In addition, things like speakers have no number of detection since it does not belong in the dataset used for the prototype.

***Table 3.1 Functionality Testing of the objects in garage space***

| Measured Distance (m) | Object | Actual Distance (m) | No. of Times the object identified | Reliability Testing |
|---|---|---|---|---|
| 10 | Chair | 10.02 | 10 | 100% |
| 10 | Car | 9.87 | 10 | 100% |
| 10 | Bottle | 9.96 | 8 | 80% |
| 10 | Pail | 9.76 | 7 | 70% |
| 10 | Bike | 9.90 | 8 | 80% |

Table 3.1 shows the functionality testing of the objects seen in the garage space. As observed, the chair and car were the only objects that were able to detect successfully in 10 trials over the distance of 10 meters measured distance. On the other hand, 80% and 70% reliability results for the bottle and bike, and pale, respectively.

***Table 3.2 Functionality testing of the objects in the living room space***

| Measured Distance (m) | Object | Actual Distance (m) | No. of Times the object identified | Reliability Testing |
|---|---|---|---|---|
| 5 | Chair | 4.98 | 10 | 100% |
| 5 | Couch | 4.87 | 10 | 100% |
| 5 | TV Monitor | 4.94 | 10 | 100% |
| 5 | Teddy Bear | 4.7 | 10 | 100% |
| 5 | Dog | 4.65 | 10 | 100% |
| 5 | Bottle | 4.89 | 10 | 100% |
| 5 | Mouse | 4.92 | 9 | 90% |
| 5 | Keyboard | 5.00 | 10 | 100% |
| 5 | Pillow | 4.79 | 10 | 100% |
| 5 | Cup | 4.86 | 9 | 90% |

Table 3.2 presents the measured distance of five (5) meters to the objects vs. the actual distance and the number of times that the object is identified. It is observed that the mouse and the cup were the only objects that had a little discrepancy, having a reliability result of 90%;

other than that, all the objects were detected successfully in 10 trials.

***Table 3.3 Functionality Testing of Text Detection in the Living Room (5m) and Garage (10m) Spaces***

| Measured Distance (m) | Signage | Actual Distance (m) | No. of Times the signage identified | Reliability Testing |
|---|---|---|---|---|
| 5 | Entrance | 4.67 | 10 | 100% |
| 5 | Exit | 4.89 | 10 | 100% |
| 10 | Entrance | 10.09 | 10 | 100% |
| 10 | Exit | 9.98 | 10 | 100% |

Table 3.3 presents the functionality testing of the prototype for the text detection in both signages: Entrance and Exit in the living room and garage spaces. Both in the garage and in the living room spaces accumulated a reliability result of 100% for both signages.

***Table 3.4 Processing Time of the System 0***

| Trials | Processing Time | |
|---|---|---|
| | Slow Pace | Maximal Walking |
| 1 | 1.82s | 1.98s |
| 2 | 1.67s | 2.12s |
| 3 | 1.49s | 1.87s |
| 4 | 1.86s | 1.86s |
| 5 | 2.03s | 1.96s |
| 6 | 1.78s | 2.08s |
| 7 | 1.72s | 1.93s |
| 8 | 1.61s | 1.78s |
| 9 | 1.33s | 1.82s |
| 10 | 1.42s | 1.76s |
| Average | 1.673s | 1.916s |

Table 3.4 shows the average processing time of the object and text detection of the system while the person wearing the prototype is walking at a slow pace walking. The values show the processing time ranges from 1.33 to 2.03 seconds. The average processing time of the smart glass was 1.673 seconds. Moreover, it also shows the average processing time of the object and text detection of the system while the person wearing the prototype is maximal walking. The values show the processing time ranges from 1.76 to 2.12 seconds. The average processing time of the smart glass was 1.916 seconds.

### *3.2 Accuracy Testing*

The accuracy testing of the text detection is conducted with ten (10) trials. It is done in accordance with the functionality testing. Moreover, the gathered data will be computed using the formula in Equation 2. Percent Error will be used to determine the accuracy of the test performed for the object and text signage detection. In this equation, the Vo is the variable for the number of Yes in the trial. Va is accepted as the truth in the system which is the Yes.

$$Percentage\ Error = \left(\frac{|V_o - V_a|}{V_a}\right) X\ 100\% \qquad (2)$$

***Table 3.50 Accuracy Testing for Object Detection***

| Objects | % Error | Objects | % Error | Objects | % Error |
|---|---|---|---|---|---|
| Chair | 0% | TV Monitor | 0% | Bottle | 20% |
| Car | 0% | Dog | 30% | Pillow | 10% |
| Bike | 20% | Speaker | 100% | Mouse | 10% |
| Teddy Bear | 0% | Table | 30% | Pail | 30% |
| Keyboard | 0% | Cellphone | 0% | Cup | 10% |

Table 3.5 shows the accuracy testing for object detection. With the conducted trials, 100% error was the computed value for the speaker. It is because the object is not configured in the system. On the other hand, some objects have little to no errors. These minimal errors were caused by some factors: distance (garage space) and identifying and detecting different objects (e.g., dog to a cat).

***Table 3.6 Accuracy Testing for Text Detection (Entrance)***

| Detection of the Word "ENTRANCE" using fonts styles: | No. of times the word "ENTRANCE" is detected out of 10 Trials | Percentage Error |
|---|---|---|
| *Arial* | 10 | 0% |
| *Calibri* | 7 | 30% |
| *Garamond* | 10 | 0% |
| *Times New Roman* | 10 | 0% |
| *Bodoni MT* | 10 | 0% |

As shown in Table 3.6 the results for the accuracy testing for the text detection. In this testing, five (5) different font styles were used to determine if the system was able to detect the word “ENTRANCE”. It can be seen that out of all the font styles used, the font style Calibri has the greatest percentage error. It is observed that the font style Calibri has a small structure thus, the system cannot detect the word “ENTRANCE” properly.

***Table 3.7 Accuracy Testing for Text Detection (Exit)***

| Detection of the Word "EXIT" using font styles: | No. of times the word "EXIT" is detected out of 10 Trials | Percentage Error |
|---|---|---|
| *Arial* | 10 | 0% |
| *Calibri* | 8 | 20% |
| *Garamond* | 10 | 0% |
| *Times New Roman* | 10 | 0% |
| *Bodoni MT* | 10 | 0% |

As shown in Table 3.7 the accuracy testing for the text detection of the word “EXIT”. Five (5) different font styles were used. It is observed from the data that out of the five (5) font styles used, Calibri has the highest percentage error of 20%. It is observed that the font style Calibri has a small structure thus, the system cannot detect the word “ENTRANCE” properly.

## IV. CONCLUSION AND RECOMMENDATION

The researchers could design a low-cost smart glass using Raspberry Pi 2 for blind individuals. To achieve this, the algorithms used for the prototype are YOLOv3 as the object detection tool and OCR for the text detection. The audio output will give the detected object or text. In object detection, the objects seen in the garage were identified and detected at 10 meters distance. The pail was the least in the reliability test, having a 70% result in 10 trials.

Meanwhile, in the living room space, the objects were detected and identified well in 5 meters distance in 10 trials with 100% reliability test except for the mouse and cup having 90% reliability. For the text detection, it was observed that Arial, Times New Roman, Bodoni MT, and Garamond were the best font style leading to a percentage error of 0% for both signages: Entrance and Exit that was conducted in 10 trials. On the other hand, Calibri was a result of a 30% error for the word Entrance and a 20% error for the word exit since the font style Calibri was smaller in characters. The average processing time of the system in slow pace walking was 1.673 seconds and 9.196 seconds in maximal walking.

The study is about detecting objects and texts. The researchers recommend that others who desire to pursue the same topic take a more significant step in detecting different languages and even symbols, figures, and logos. Moreover, for future studies, give specific directions, warning messages, and audio output to prevent unwanted accidents. Additionally, since there were many restrictions because of the pandemic, the researchers recommend testing the prototype on a broader environment, and lastly, providing a GPS notification for navigation purposes of the blind person.

## REFERENCES


1. A. Mandal, “What is visual impairment?” https://www.news-medical.net/health/What-is-visual-impairment.aspx (accessed Apr. 19, 2021).
2. E. A. Hassan and T. B. Tang, “Smart glasses for the visually impaired people,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, vol. 9759, no. January, pp. 579–582, DOI: 10.1007/978-3-319-41267-2_82.
3. H. Bhorshetti, S. Ghuge, A. Kulkarni, P. S. Bhingarkar, and P. N. Lokhande, “Low Budget Smart Glasses for Visually Impaired People,” pp. 48–52, 2019.
4. C. Davis, “Medical Definition of Blindness.” https://www.medicinenet.com/blindness/definition.htm (accessed Apr. 15, 2021).

5. "Two Kinds of Blind – Retina Specialist | Fairfax, Virginia | Retinal Diseases."https://retinaeyedoctor.com/2010/02/diabetic-retinopathy-and-macular-degneration-cause-legal-blindness-and-complete-blindness/#:~:text=There are two types of, a degeneration that causes only legal blindness (accessed Apr. 15, 2021).
6. P. Bainter and A. Dahl, "Blindness," 2020.https://www.emedicinehealth.com/blindness/article_em.htm.
7. "Top 5 Electronic Glasses for the Blind and Visually Impaired." https://irisvision.com/electronic-glasses-for-the-blind-and-visually-impaired/ (accessed Apr. 15, 2021).
8. R. Noe, "IrisVision: A Smartphone/Goggles Combination Intended to be a Cheaper, Less-Invasive Alternative to Retinal Chip Implants for the Visually Impaired," 2019. https://www.core77.com/posts/90755/IrisVision-A-SmartphoneGoggles-Combination-Intended-to-be-a-Cheaper-Less-Invasive-Alternative-to-Retinal-Chip-Implants-for-the-Visually-Impaired (accessed Apr. 17, 2021).
9. "AceSight - Electronic glasses," 2021. https://www.quantumrlv.com.au/products/acesight-electronic-glasses (accessed Apr. 17, 2021).
10. "New Smart Glasses for Visually Impaired," 2016. https://www.reviewob.com/new-smart-glasses-for-visually-impaired/.
11. "Vision Made Audible OrCam MyEye 2." https://time.com/collection/best-inventions-2019/5733047/orcam-myeye-2/ (accessed Apr. 17, 2021).
12. J. Bai, S. Lian, Z. Liu, K. Wang, and Di. Liu, "Smart guiding glasses for visually impaired people in an indoor environment," IEEE Trans. Consum. Electron., vol. 63, no. 3, pp. 258–266, 2017, doi: 10.1109/TCE.2017.014980.
13. A. Audomphon and A. Apavatjrut, "Smart Glasses for Sign Reading as Mobility Aids for the Blind Using a Light Communication System," 17th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol. ECTI-CON 2020, pp. 615–618, 2020, doi: 10.1109/ECTI-CON49241.2020.9158250.
14. L. Alkhatib, H. Al Said, A. Aloraidh, and S. Alhaidar, "Smart Glasses for Blind people," 2019.
15. A. Nayyar and V. Puri, "Raspberry Pi- A Small , Powerful , Cost Effective and Efficient Form Factor Computer : A Review International Journal of Advanced Research in Raspberry Pi- A Small , Powerful , Cost Effective and Efficient Form Factor Computer : A Review," Int. J. Adv. Res. Comput. Sci. Softw. Eng. 5(12), vol. 5, no. 12, pp. 720–737, 2015.
16. "Simulink." https://www.mathworks.com/help/simulink/ (accessed Apr. 17, 2021).
17. A. Choudhury, "Top 8 Algorithms For Object Detection," 2020. https://analyticsindiamag.com/top-8-algorithms-for-object-detection/ (accessed Apr. 18, 2021).
18. Cruz, F. R. G., Santos, C. J. R., & Vea, L. A. (2019). Classified Counting and Tracking of Local Vehicles in Manila Using Computer Vision. 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management ( HNICEM ), 1–5. https://doi.org/10.1109/HNICEM48295.2019.9072808
19. G. Plastiras, C. Kyrkou, and T. Theocharides, "Efficient ConvNet-based Object Detection for Unmanned Aerial Vehicles by Selective Tile Processing," arXiv, 2019.0
20. [1]"YOLOv3: Real-Time Object Detection Algorithm (What's New?)," *viso.ai*, Feb. 25, 2021. https://viso.ai/deep-learning/yolov3-overview/.
21. P. M. Manwatkar and S. H. Yadav, "Text recognition from images," in 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Mar. 2015, pp. 1–6, doi: 10.1109/ICIIECS.2015.7193210.
22. M. Sonkusare and N. Sahu, "A Survey on Handwritten Character Recognition (HCR) Techniques for English Alphabets," Adv. Vis. Comput. An Int. J., vol. 3, no. 1, pp. 1–12, Mar. 2016, doi: 10.5121/avc.2016.3101.

23. K. Hamad and M. Kaya, “A Detailed Analysis of Optical Character Recognition Technology,” Int. J. Appl. Math. Electron. Comput., vol. 4, no. Special Issue-1, pp. 244–244, 2016, doi: 10.18100/ijamec.270374.
24. “Banner Font Size - How Big Should They be? | 48HourPrint,” www.48hourprint.com. https://www.48hourprint.com/banner-font-size.html (accessed Nov. 18, 2021).
25. [1]“What is Luminance? - Meaning, Definition, Formula, Uses,” *BYJUS*. https://byjus.com/physics/luminance/.
26. Samonte, M. J. C., Bejar, A. M. L., Bien, H. C. L., & Cruz, A. M. D. (2019). Senior Citizen Social Pension Management System Using Optical Character Recognition. 2019 International Conference on Information and Communication Technology Convergence (ICTC), 456–460. https://doi.org/10.1109/ICTC46691.2019.8940013
27. [1]E. M. Murtagh, J. L. Mair, E. Aguiar, C. Tudor-Locke, and M. H. Murphy, “Outdoor Walking Speeds of Apparently Healthy Adults: A Systematic Review and Meta-analysis,” *Sports Medicine (Auckland, N.z.)*, vol. 51, no. 1, pp. 125–141, 2021, doi: 10.1007/s40279-020-01351-3