Available *online* at : http://jnte.ft.unand.ac.id/

Jurnal Nasional Teknik Elektro

| ISSN (Print) 2302-2949 | ISSN (Online) 2407-7267 |

Audible Obstacle Warning System for Visually Impaired Person Based on Image Processing

Andik Yulianto[1], Ni'matul Ma'muriyah[1], Lina[2]

[1] *Department of Information System, Universitas Internasional Batam, Batam, Indonesia*
[2] *Department of Electrical Engineering, Universitas Internasional Batam, Batam, Indonesia*

ARTICLE INFORMATION

Received: February 26, 2022
Revised: July 25, 2023
Accepted: July 31, 2023
Available online: July 31, 2023

KEYWORDS

Obstacle Warning System, Visually Impaired Person, Haar Like Feature, Hough Transform

CORRESPONDENCE

Phone: +62 811693767
E-mail: andik@uib.ac.id

A B S T R A C T

To be able to do their daily activities, a visually impaired person needs a guidance device to help him/her walk including to avoid obstacles on their way to the destination. The quick and clear instruction is given to the user is the most challenging problem to be solved. The visually impaired person should have simple guidance about the obstruction in front of him/her. Most guidance devices use simple sounds to give the warning without information about which direction the user should go. In this paper, an obstacle warning system based on image processing methods was developed. A guidance device for visually impaired persons using a single-board computer based on an image-processing algorithm has been designed. The main sensor of the guidance device is a NoIR camera. The distance measurement approximation model was developed with errors up to 4.3%. The test found that the proposed system can detect obstruction in the form of a person, the device also detects the stairs. The best detection obtains when the object position is less than 300 cm in front of the user. The stair detection was carried out by using the Hough line transform method. The output of the system is the sound of direction that can be heard through the headset.

INTRODUCTION

To improve the ability of blind or significant visual impairment person to navigate indoors or outdoors environment, many devices known as electronic travel aid (ETA) have been developed. One specific task to accomplish the goal of navigating for the blind is to avoid obstacles that may exist on their path. Some related studies on blind guides have been developed, mostly using ultrasonic sensors as in [1]–[5]. In [4] also developed the NavBelt, a belt-mounted device equipped with eight ultrasonic sensors installed around the belt providing a 120° wide range detection. NavBelt provides information to the user in the form of panoramic acoustic sound that describes the atmosphere of the surrounding environment through headphones. This belt is best when used by the trained or experienced user as the user would need to concentrate on the acoustic guidance signals. Wheeled robots with ultrasonic sensors are also utilized to detect whether there are obstacles or not in front of the user [5]. The robot is equipped with a handlebar that will guide the user to move in any directions.

Research on visual aids continues to be developed by utilizing a camera and ultrasonic technology. In [6] blind visual aids were developed using Raspberry Pi to run the programs, USB webcam for human detection, and ultrasonic sensors installed like belts for the detection of the objects. This system can detect a person using a USB webcam when the distance of the person is under 120 cm from the user and give a "beep" alert to the user. Here, the user has difficulties to make their own decisions which direction to go. Another research on visual aids using embedded system and image processing also developed in [7], [8]. In [8] a guiding system developed uses ultrasonics and a camera to detect the obstacles and estimate the distance using Fuzzy Logic Inference System. The output of this aid is an audio command to redirect the user to go. The use of ultrasonic was effective to detect the distance of any object in the range of detection without the ability to identify the object.

Visual ETA has been developed to detect an immovable obstacle, e.g stair, door, and wall, using a template matching method where it needs to train several train Stair can be detected by its handlebar, whether for upstairs or downstair images [7]. Stairs are considered an obstacle because they are an uneven surface that may cause users to trip and fall. However, the visual impaired person not only faced an immoveable obstacle while navigating

https://doi.org/10.25077/jnte.v12n2.1008.2023 Attribution-NonCommercial-ShareAlike 4.0 International. Some rights reserved

independently indoors or outdoors environments. Another person who walks toward the visually impaired person can be considered an obstacle.

In this paper, we proposed a design of an audible obstacle warning system on visual aid device using low-cost single board computer named Raspberry Pi and NoIR (No Infra Red) filter camera to detect a person who walks toward the user. Raspberry Pi known has been succesfully used in various image processing applications such as traffic density monitoring [9], Braille letter identification [10], or in detect the home intruder based on face detection [11].

The ETA proposed in this paper provides audible commands that are easy to understand for the users. This ETA also adds the ability to detect the presence of stairs in front of the user, so that users can be more careful with the object in front of him/her. We employed the Haar-like feature [12] which is one feature that is widely used for frontal face detection as the person detector. The Haar-like feature found successfully used in related various applications, for face detection and recognition in [13]–[18] for pedestrian detection on the road as in [19], [20], and face tracking as in [21]. In this research we used the Hough Transform and for stairs detection.

METHOD

The hardware block diagram of the proposed design is shown in Figure 1. The device consists of a single board Raspberry Pi Model B with the Raspbian Jessie Operating System Version 4.1. As an input, a 5-megapixel NoIR camera is used to acquire the picture of the situation in front of the user. OpenCV 3.0 installed to Raspberry Pi was used to process the images captured by the camera. The output sound for the user can be heard through an earphone that is attached to a 3.5mm jack at Raspberry Pi. The system is powered by a 5800 mAH battery.

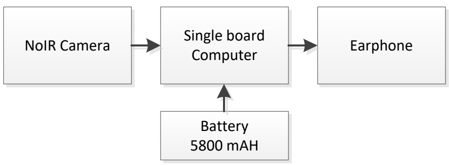


Figure 1. Audible Obstacle Warning System Block Diagram

Figure 2 shows the image is processed to obtain facial features and the stairs and obtain the correspons output. The process begins with the preprocessing image stage, which converts the colored images captured into the gray-scale image. The next stage is face detection and stair detection then determining the direction the user must follow and convert it into sound.

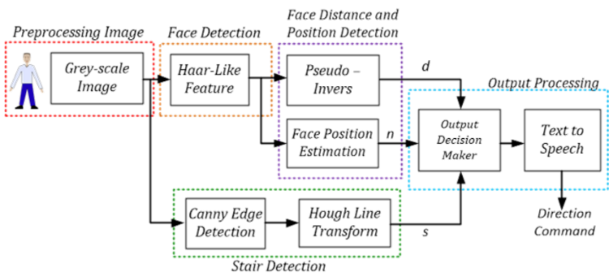


Figure 2. Proposed Flow Diagram to Obtain the Direction Command from the Facial Features and Stairs Existence

The initial stage of the whole process is to capture a colored image of the situation in front of the device then convert it into gray-scale image. The image was captured by camera in real-time with 30 frames per second and then resized the image to 320 × 240 pixels. This size is selected due to the processor speed, the higher size of the picture will cause the slow response of processing.

Face Detection

Face detection is needed to determine the presence of the human body in front of the user. The face detection was carried out by the Haar-like feature cascade classifier. In order to detect the object (face), in the Haar cascade classifier the images classified using the value of simple features rather than the pixels directly. This method operates much faster than a pixel-based method [12]. Haar Classifier use three kinds of features which consist of grey and white rectangle shown in Figure 3. The *two-rectangle feature* Figure 3(a) and (b) computes the difference between the sum of the pixels within two rectangular regions horizontally or vertically. The *three-rectangular feature* computes the sum of the pixels within two white outside rectangles substracted from the sum of in center rectangles. Finally, the *four-rectangle feature* computes the difference between diagonal pairs of rectangles.

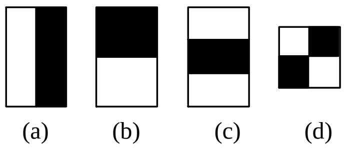
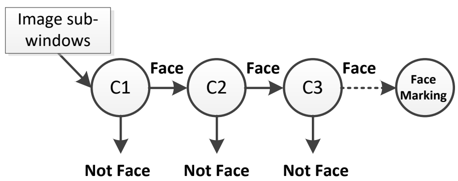


Figure 3. Haar features. The *two-rectangle feature* shown in (a) and (b). Figure (c) shows the *three-rectangle feature*, and (d) *four-rectangle feature* [12].

The value of Haar-like feature computes rapidly using integral image techniques explained in [12]. When the value satifisfied the threshold value it will considered as the feature. The threshold value can be adjusted using machine learning so that the classifier can minimized the false negative rates. AdaBoost algorithm is employed to learn a classification function from given feature set and a training set of positive and negative images. The positive images contain the face images, while the negative images are non-face images. To get more accuracy of object detection, the more number of Haar-like feature process should be employed, this means the computation time will increase. A cascade classifier constructed to reduce the computation time radically[12]. Figure 4 shows a series of classifiers are applied to every sub-window of an image. Each classifier employed at least one Haar-like feature with a trained threshold value. A positive result from the first classifier triggers the second classifier to achieve high detection rates, and so on. The last process is to mark the true-positive sub-window region with a rectangle.



https://doi.org/10.25077/jnte.v12n2.1008.2023

Figure 4. A cascade classifier diagram for face detection.

We implement the Haar-like cascade classifier using the Python by utilizing the function from OpenCV shown in Figure 5 which return boundary rectangles parameter for the detected faces denotes *x*, *y*, *w*, and *h* from variable `faces`. Visually, the function output parameters on an image shown in Figure 6. The *x* and *y* denotes the initial coordinates of the detected face. Whereas, *w* and *h* are the width and height of the face respectively.

```
face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_d
efault.xml')
faces = face_cascade.detectMultiScale(image,
scale factor, minNeighbors, minSize, flags)
```

Figure 5. Cascade classifier function for face detection

From Figure 5, the `scaleFactor` determines the factor of increase in window size while classifying the sub-window. The higher `scaleFactor` will cause the detection process faster but they may ignore the face size between the two scales. In our system, we set the scaleFactor to 1.1. The `minNeighbors` specifies how many neighbors each candidate rectangle should have to retain it. This parameter controls how many rectangles (neighbors) need to be detected for the window to be labeled as a face. In our system, we set the `minNeighbors` parameter to 5, and another two parameters `minSize` and `flags` remain not set.

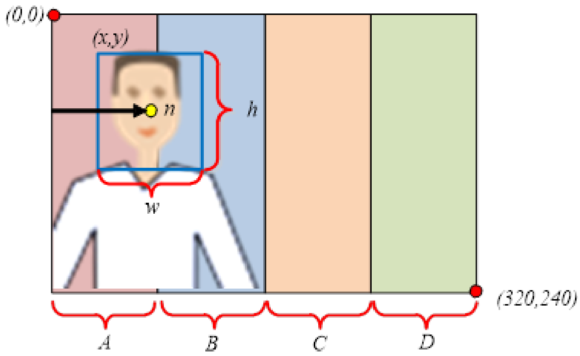


Figure 6. Human Position Calculation

Face Position and Distance Estimation

Figure 6 shows how to calculate the human position relative to the face detected from the image aquired. If one face is detected, then we mark the image with blue rectangle and the value of *x*, *y*, *w*, and *h* can be obtained. The face position is estimated to determine whether humans are on the right or left side of the user. This can be done by divide the 320 × 240 pixels image frame into four parts vertically denotes A, B, C and D which has range of $n < 80$ pixels, $80 < n < 160$ pixels, $80 < n < 160$ pixels and $160 < n < 240$ pixels respectively. The variable *n* denotes the midpoint of the face that is solved by (1),

$$n = x + \left(\frac{w}{2}\right) \tag{1}$$

where *x* is the coordinate on the upper left of the face and *w* is the width of the pixel of the face. This system only pays attention to the value of *x* in determining the direction to go, while the value of *y* only shows the high and low of humans detected. If there are many faces detected, the closest distance to the user will be considered as an obstacle to be avoided. Based on the coordinates obtained, all faces were marked in the blue box, and faces with the largest area of pixel was marked with a yellow box. The largest pixel indicates that the face detected has the closest distance from the user.

To estimate the distance of person in front of the user we constructed a regression model using the pseudoinverse linear regression. Before finding the equation of the approach that will be used to estimate the distance, the data on the number of pixels in a face box that is detected is compared to a certain distance. In this case, we measured the distance of the face in a range of 50 – 220 cm with an interval of 5 cm and then record the number of pixels of the face width *w* detected. The relation between the distance and the width of face detected shown in Table 1.

Tabel 1. The Comparation of the Face Width Detected Measurement and in range of 50 – 220 cm of Camera Distance to User

No	Actual Distance (cm)	Face Width (pixel)
1	50	81
2	55	75
3	60	72
4	65	68
5	70	65
6	75	62
7	80	57
8	85	56
9	90	51
10	95	50
11	100	48
12	105	47
13	110	46
14	115	45
15	120	44
16	125	43
17	130	41
18	135	39
19	140	37
20	145	35
21	150	34
22	155	33
23	160	32
24	165	29
25	170	28
26	175	27
27	180	26
28	185	25
29	190	24
30	195	23
31	200	22

From Table 1 we define the regression model to estimate the distance of the person in front of camera and enter it into the program. Distance estimation is done using the quadratic equation as in (2), where *d* is the estimation of distance in cm, c_1, c_2, c_3 is the value of the polynomial constant and *x* is the value of the pixel width of the face.

$$d = c_1 x^2 + c_2 x + c_3 \tag{2}$$

By mapping each data point in Table 1 to equation (2) and we can rewrite the equation in matrix form as shown in (3).

$$\underbrace{\begin{bmatrix} 50 \\ 55 \\ \vdots \\ 195 \\ 200 \end{bmatrix}}_{\mathbf{D}} = \underbrace{\begin{bmatrix} 81^2 & 81 & 1 \\ 75^2 & 75 & 1 \\ \vdots & \vdots & \vdots \\ 23^2 & 23 & 1 \\ 22^2 & 22 & 1 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} c1 \\ c2 \\ c3 \end{bmatrix}}_{\mathbf{C}} \tag{3}$$

Since $\mathbf{X}$ is a non-square matrix and they are not invertible, the pseudoinverse of **X,** denoted by $\mathbf{X}^{\dagger}$, which is the generalized inverse of $\mathbf{X}$ was employed to solve $\mathbf{C}$. The matrix $\mathbf{X}^{\dagger}$ define by (4), where T denotes the transpose [22].

$$\mathbf{X}^{\dagger} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T \tag{4}$$

Thus, we can rearange the (3) to solve the $\mathbf{C}$ as in (5)

$$\mathbf{C} = \mathbf{X}^{\dagger}\mathbf{D} \tag{5}$$

Finally, by using MATLAB we can obtain the matrix $\mathbf{C}$ using (5) and get the value of $c_1 = 0.0336$, $c_2 = -5.9634$ and $c_3 = 313.5201$. Subtitutes these values to equation (2) to get the distance regression model shown in (9).

$$d = 0.0336x^2 - 5.9634x + 313.5201 \tag{6}$$

From Figure 5, it is observed that the measured distance is almost linear to the actual distance. The result of estimation using (6) has an average error up to 4.3%. This varies depending on the width of a person's face, usually, women have smaller than men.

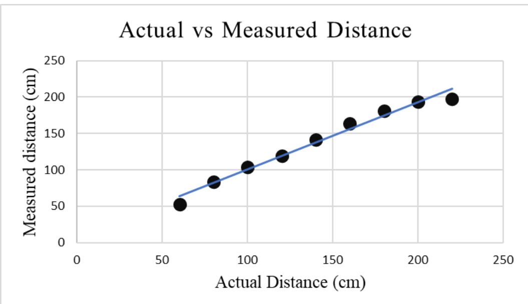


Figure 7. Distance Measurement Linearity using Regression Model Derived

Stair Detection

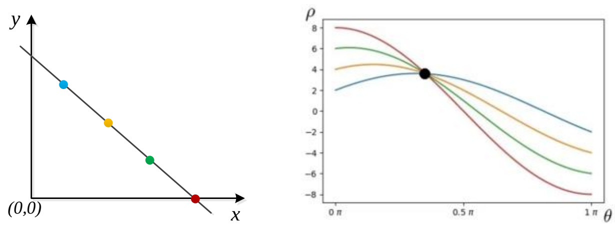
The existence of the stairs is determined by the number of lines detected in the image at the stage of image preprocessing. We employed the Hough line transform to detect lines based on the edge points in a binary image which can be obtained by edge detector algorithm. It transforms between the cartesian space and a parameter space in which a straight line exists in an image. In a cartesian space, a line represented as (7) and can be graphically plotted for each pair of image points (x_i, y_i).

$$y_i = mx_i + c \tag{7}$$

In Hough transform the lines is defined by slope-intercept parameter which is the pair of m and c values. However, if the line is vertical the m value is infinite makes the slope parameter for vertical line is not defined. Therefore, the in generalized Hough transform the (x, y) space is converted into (ρ, θ) space using (8)[23].

$$\rho = x\cos\theta + y\sin\theta \tag{8}$$

The ρ denotes the perpendicular distance of the straight line, and θ is the angle between the perpendicular line and the horizontal axis. In parametric space (ρ, θ), points are forming a sinusoid. If several sinusoids are intersected, means that a straight line is in the image. Figure 8 illustrated how the points in a line plotted in cartesian plane (a) then represented in Hough space plane as intersected sinusoids (b). To obtain the pair of (ρ, θ) each edge point in the image observed, then the different ρ calculated using (8) by varying θ in range of [0π, 1π] radian with resolution of *theta*. The occurances of (ρ, θ) can be accumulated as votes in a matrix, and finally the high accumulated votes of (ρ, θ) are identified as a line and reconstructed to the actual line on the image.



(a) (b)

Figure 8. The points in a line plotted in cartesian plane (a), converted to Hough space plane as intersected sinusoids (b).

A several image preprocessing must be conducted before the Hough Line Tranform is implemented. After image acquisition, first we convert the color image into greyscale image, then the canny edge detection conducted to obtain the binary image with edge points. This image result will use as the input of Hough process. The image grayscaling and edge detection was done using `cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)` and `cv2.Canny()` respectively. The canny function parameters set as: `lower_threshold = 50`, `upper_threshold = 150` and `apertureSize = 3` which is determine the use of Sobel Operation. The Sobel operation is a filter by looking for the middle gradient value of two 3x3 matrices.

The function `cv2.HoughLines(image, rho, theta, threshold)` from OpenCV was utilized to implement the Hough Line Transform. The first parameter is the input binary image obtained from preprocessing process above. The next parameters were set as *rho* = 1 and *theta* = π/180, both define the ρ resolution and θ resolution respectively. Based on our experiment the best result of line detection achieved with `threshold` parameter set to 180. The *threshold* parameter determines the threshold value of line detection. For each line detected x and y coordinates will be determined using the values of *rho* and *theta*. From the coordinates obtained, a line with a red

https://doi.org/10.25077/jnte.v12n2.1008.2023

color is drawn according to the results of the detection. If there are more than 3 horizontal lines paralyzed detected, it is considered that there are stairs in front of the user.

Output Decision Maker

The algorithm implements a simple method of output decision-maker flow as shown in Figure 9. To determine where the user should move, there are three decisions taken by the system. First decisions based on stairs detection indicates by the detected lines, *s*. If the line is found more than 3 lines, then the system will give an output text of "Stair Detected. Be Careful". Second decision based on the distance of the face detected, *d*. If there are many faces with a distance of less than 300 cm, the system will consider the closest face. Meanwhile, faces with a distance of more than 300 cm will be ignored. The third decision is based on the position of the face detected, n. From the range of data that has been made at the position detection stage, the resulting output is based on:

1) The value of n < 80 pixels gives the output text of "Go Right".
2) The value of 80 ≤ n < 160 pixels gives the output text of " Go Right. Go Right".
3) Value 160 ≤ n < 240 pixels gives the output text of "Go Left. Go Left.
4) Value n ≥ 240 pixels gives the output text of "Go Left."

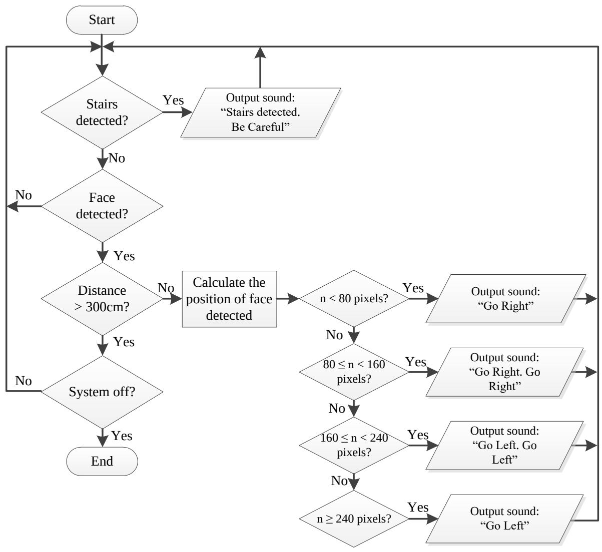


Figure 9. Output Decision Maker Flowchart

After an output text from decision-maker is obtained, then the text will be converted to sound using Text to Speech (TTS) engine. In this reseach carried out by implementing the eSpeak speech synthesizer which supports various types of international (multi-lingual) languages (ESpeak Text to Speech, n.d.)[24].

RESULTS AND DISCUSSION

The proposed design of the ETA is shown in Figure 10(a) consisting of Raspberry Pi and battery enclosed with a box and an earphone attached. The ETA hung on the chest as seen in Figure 10(b), as the result, the device only captures the picture of object exist in chest level capture area in a certain distance. The first test was carried out by testing the detection of one face, two faces, and more than two faces. Furthermore, we also test to the person who is using accessories. Finally, stair detection testing is done by taking images of several types of stairs from different camera points of view.

(a) The Prototype Design

(b) Device proposed in use

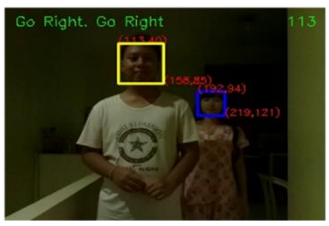
Figure 10. Prototype Design of ETA

The next test is carried out by placing two persons at a distance of less than 300 cm and more than 300 cm as shown in Figure 11(a). The test results show that only one face that is less than 300 cm can be detected. The next test is done by placing two persons in front of the user at a distance of <300 cm. Person A is conditioned facing the camera, while the Person B is conditioned to look down on his face. The results shown in Figure 11(b) show that only one face facing the camera (frontal) can be detected. In this case the output command that comes out is "Go Left. Go Left". However, we can see that there is one vertical line detected as result of Hough Lines Transform algorithm we used, but it is not considered as a stair.

The next test was done by placing two persons at < 300 cm to the user in a room with low light intensity. As seen in Figure 11(c), both faces can still be detected. Next test done by testing the device using three persons standing at different distances in the range of <300 cm in front of the user. The result is shown in Figure 11(d). The system can detect three frontal faces and can display each of the coordinates and give a warning based on the position of the closest face detected (yellow box).

(a)

(b)



(c)

(d)

Figure 11. Test result for multi faces at a different distance. Where (a) two persons with different distance, (b) two persons with frontal face and otherwise, (c) with dimmed lamp environment, (d) three persons located at a different distance.

The next test was done for detection of person using accessories such as clear glasses, black glasses, a hat, and a hijab, respectively shown in Figure 12(a-d). The result shows that the device can detect the person face while using mentioned accessories if their faces are facing the camera and in range of detection. The output sound also comes out correctly according to the position of the detected face.

The stair detection algorithm tested by taking pictures of outdoor upstairs with some different angle. When the device headed straight as in Figure 13(a) or sideways against the stair as in Figure 13(b), the device can detect the steps and give a warning sound. We found that the detection of steps cannot be more than three lines, but this is sufficient to give a decision that there is a stair in around the user. In Figure 13(c) we take a picture where there is a person in front of the stair. In this condition, the stairs can be detected but the person is not detected by the device because the camera is not capturing the frontal face. The system failed to detect the downstairs because the stairs are lower or out of range of the camera capture ability.

(a) (b)

(c) (d)

Figure 12. Test Result of with Various Accessories

(a) (b) (c)

Figure 13. Staircase detection test result, (a) headed straight posistion, (b) sideways against the stair, (c) with a pearson in front of the stair

CONCLUSIONS

The development of an obstacle warning system based on image processing has been presented in this paper. Based on the results of research conducted, the device has features to detect one or persons by his/her face with or without accessories such as clear glasses, sunglasses, hats, and hijab even in a dimmed light, but only if the frontal human face is detected. The device can also detect the upstairs through the Hough Line Transform algorithm but failed to detect the downstairs. For further development, we suggest using another method to detect many moveable and immoveable objects in a wider range of camera detection.

ACKNOWLEDGMENT

This work is funded by The Ministry of Research, Technology and Higher Education of Indonesia and thanks to the SMU Kartini Batam for the support of user testing.

REFERENCES

[1] D. E. Gbenga, A. I. Shani, and A. L. Adekunle, “Smart Walking Stick for Visually Impaired People Using Ultrasonic Sensors and Arduino,” *International Journal of Engineering and Technology*, vol. 9, no. 5, pp. 3435–3447, 2017, doi: 10.21817/ijet/2017/v9i5/170905302.

[2] S. Hashino and R. Ghurchian, “A blind guidance system for street crossings based on ultrasonic sensors,” in *The 2010 IEEE International Conference on Information and Automation*, IEEE, Jun. 2010, pp. 476–481. doi: 10.1109/ICINFA.2010.5512383.

[3] A. Sen, K. Sen, and J. Das, “Ultrasonic Blind Stick for Completely Blind People to Avoid Any Kind of Obstacles,” in *Proceedings of IEEE Sensors*, India: IEEE, Oct. 2018, pp. 1–4. doi: 10.1109/ICSENS.2018.8589680.

[4] S. Shoval, J. Borenstein, and Y. Koren, “Auditory guidance with the navbelt-a computerized travel aid for the blind,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 28, no. 3, pp. 459–467, 1998, doi: 10.1109/5326.704589.

[5] S. Shoval, I. Ulrich, and J. Borenstein, “NavBelt and the GuideCane,” *IEEE Robot Autom Mag*, vol. 10, no. 1, pp. 9–20, Mar. 2003, doi: 10.1109/MRA.2003.1191706.

[6] M. Tripathi, M. Kumar, V. Kumar, and W. Kandlikar, “Darshan : Electronics Guidance For The Navigation Of Visually Impaired Person,” *International Journal For Research in Applied Science and Engineering Technology (IJRASET)*, vol. 2, no. 6, pp. 335–342, 2014.

[7] A. Abdurrasyid, I. Indrianto, and R. Arianto, “Detection of immovable objects on visually impaired people walking aids,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 2, p. 580, Aug. 2018, doi: 10.12928/telkomnika.v17i2.9933.

[8] N. Ma’muriyah, A. Yulianto, and Lili, “Design prototype of audio guidance system for blind by using raspberry pi and fuzzy logic controller,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Sep. 2019, p. 12024. doi: 10.1088/1742-6596/1230/1/012024.

[9] F. Rahmadina and . Z., "Sistem Informasi Lalu Lintas Berbasis Raspberry Pi PC Board," *JURNAL NASIONAL TEKNIK ELEKTRO*, vol. 5, no. 1, Mar. 2016, doi: 10.20449/jnte.v5i1.190.

[10] R. Ramiati, S. Aulia, and L. Lifwarda, "Aplikasi Identifikasi Huruf Braille Menggunakan Computer Vision Berbasis Raspberry Pi," *JURNAL NASIONAL TEKNIK ELEKTRO*, vol. 9, no. 1, p. 12, Mar. 2020, doi: 10.25077/jnte.v9n1.707.2020.

[11] R. A. Nadafa, S. M. Hatturea, V. M. Bonala, and S. P. Naikb, "Home Security against Human Intrusion using Raspberry Pi," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1811–1820. doi: 10.1016/j.procs.2020.03.200.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE Comput. Soc, 2001, pp. I-511-I–518. doi: 10.1109/CVPR.2001.990517.

[13] S. J. Elias *et al.*, "Face recognition attendance system using local binary pattern (LBP)," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 239–245, 2019, doi: 10.11591/eei.v8i1.1439.

[14] G. I. Hapsari, G. A. Mutiara, and H. Tarigan, "Face recognition smart cane using haar-like features and eigenfaces," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 2, p. 973, Apr. 2019, doi: 10.12928/telkomnika.v17i2.11772.

[15] Ngo Ba Viet, Nguyen Thanh Hai, and Ngo Van Thuyen, "Hands-free control of an electric wheelchair using face behaviors," in *2017 International Conference on System Science and Engineering (ICSSE)*, IEEE, Jul. 2017, pp. 29–33. doi: 10.1109/ICSSE.2017.8030831.

[16] I. Paliy, "Face detection using Haar-like features cascade and convolutional neural network," in *2008 International Conference on "Modern Problems of Radio Engineering, Telecommunications and Computer Science" (TCSET)*, 2008, pp. 375–377.

[17] D. Pertsau and A. Uvarov, "Face detection algorithm using haar-like feature for GPU architecture," in *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, IEEE, Sep. 2013, pp. 726–730. doi: 10.1109/IDAACS.2013.6663020.

[18] Z. Zhu, T. Morimoto, H. Adachi, O. Kiriyama, T. Koide, and H. J. Mattausch, "Multi-view face detection and recognition using haar-like features," *COE Workshop.*, no. June 2014, pp. 2–4, 2006.

[19] Y. Li, W. Lu, S. Wang, and X. Ding, "Local Haar-like features in edge maps for pedestrian detection," in *2011 4th International Congress on Image and Signal Processing*, IEEE, Oct. 2011, pp. 1424–1427. doi: 10.1109/CISP.2011.6100477.

[20] Zhang Yang, Liu Weiming, Mo Chen, and Li Zilong, "Pedestrian detection in complex scene using full binary tree classifiers based on locally assembled Binary Haar-like features," in *2011 9th World Congress on Intelligent Control and Automation*, IEEE, Jun. 2011, pp. 1180–1184. doi: 10.1109/WCICA.2011.5970702.

[21] L. K. Lee, S. Y. An, and S. Y. Oh, "Efficient face detection and tracking with extended CAMSHIFT and haar-like features," *2011 IEEE International Conference on Mechatronics and Automation, ICMA 2011*, pp. 507–513, 2011, doi: 10.1109/ICMA.2011.5985614.

[22] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, pp. 406–413, 1955, doi: 10.1017/S0305004100030401.

[23] L. Chandrasekar and G. Durga, "Implementation of Hough Transform for Image Processing Applications," in *International Conference on Communication and Signal Processing*, India.

[24] "eSpeak text to speech." http://espeak.sourceforge.net/ (accessed Aug. 27, 2021).

AUTHORS BIOGRAPHY

Andik Yulianto is a lecturer in Department of Information System, Faculty of Computer Science, Universitas Internasional Batam. He holds the degree of bachelor from Institut Teknologi Nasional Malang in Electronics (2005), and master degree from Institut Teknologi Sepuluh Nopember in Control System Engineering (2012). His current research interests include Intelligent Control, Robotics, Artificial Intelligence, Image Processing and Internet of Things

Ni'matul Ma'muriyah a lecturer in Department of Information System, Faculty of Computer Science, Universitas Internasional Batam. She received the M. Eng degree in electrical and Electronic System Engineering from Nagaoka University of Technology, Nagaoka, Japan in 1998. Her current research interests include Digital Image Processing and Machine Learning.

Lina is a graduate of Universitas Internasional Batam. He holds a bachelor's degree from Department of Electrical Engineering, Faculty of Industrial Technology, Universitas Internasional Batam, Indonesia (2016). Her current research interest includes Digital Image Processing.