ICIC Express Letters
Part B: Applications
Volume **13**, Number **2**, February **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **203–210**

REAL-TIME FACE RECOGNITION SYSTEM AS PERSONAL ASSISTANT FOR PEOPLE WITH BLINDNESS

Indrabayu[1,*], Intan Sari Areni[2], Anugrayani Bustamin[1] and Vestiana Aza[1]

[1]Department of Informatics Engineering
[2]Department of Electrical Engineering
Universitas Hasanuddin
Jl. Poros Malino KM. 6, Bontomarannu, Gowa 92119, Sulawesi Selatan, Indonesia
{ intan; anugrayani }@unhas.ac.id; azav17d@student.unhas.ac.id
*Corresponding author: indrabayu@unhas.ac.id

Received March 2021; accepted June 2021

Abstract. *Recognizing others is a significant challenge for people with blindness and can hinder their social activities. This work's primary goal is to build a real-time face recognition system to support people with blindness in recognizing other people around them. The system is built on an Android platform and functioned as a personal assistant. It implements Haar-cascade classifier and local binary pattern histogram methods to detect and recognize faces from input video and produces voice as the output. Three scenarios are conducted to measure system performance. Each procedure is repeated three times and experimented with different distance values between 1 and 2.5 meters and various threshold (T) values between 45 and 90. Besides, this study also involved a parameter grid for system performance evaluation. The test scenario shows the best performance is shown in scenario 1 with the user did not move. Hence, the camera's position remained steady and could capture stable images. The experiment shows that the system results in the highest average accuracy of 90.77%, using a threshold value of 90 and 8 × 8 for grid parameters.*

Keywords: Android, Face recognition, Local binary pattern, Personal assistant, People with blindness

1. Introduction.

Blindness or visual impairment is the condition of lacking vision ability due to neurological and physiological factors. The term visual impairment includes all kinds of vision loss, ranging from complete blindness to partial vision impairment. People with visual impairments tend to experience more difficulties in doing daily activities, both indoors and outdoors. They also feel discouraged to be involved in many social activities and concerned about their privacy and physical security [1].

The rapid development of computer vision has motivated some researchers to integrate their smart systems into smartphones instead of building a separate device. It has been applied in many fields such as disaster management [2], medical care [3], education [4], and agriculture [5]. Another example of education is book reader technology for blindness. Galarza et al. developed automatic book reader tools based on Optical Character Recognition (OCR) with integrated depth maps of images [6]. Face recognition is one of the most popular research studies within computer vision. Numerous face recognition systems have been built to be complementary systems for blind people. Jin et al. built a white cane that generates unique vibration patterns for each person to inform a blind person about the detected person's identity using a camera-mounted eyeglass worn by the blind [7]. Neto et al. introduced real-time face recognition using the Microsoft Kinect sensor as a wearable device. The system used temporal coherence and a simple biometric procedure to generate a sound associated with the recognized person's name [8]. Also, Aza

DOI: 10.24507/icicelb.13.02.203

et al. built an Android-based real-time face recognition system for people with blindness. The system used gamma correction and difference of Gaussian methods to improve the Local Binary Pattern Histogram (LBPH) algorithm and produced the highest accuracy of 94.65% [9].

Various methods have been used in face recognition technology development, but LBPH has demonstrated high robustness in extracting features. An experiment conducted by Alhindi et al. showed that the LBP feature extractor outperformed HOG and deep network methods and achieved the highest accuracy of 90.52% [10]. Özdil and Özbilen compared the performance of LBP, Eigenfaces, and Fisherface in recognizing faces. This study also showed that LBP performed better than the other methods [11].

On the other hand, the Haar-cascade classifier has also been used to perform object detection, which is necessary before performing any recognition. It is used extensively on a broad range of systems, including face detection [12], emotion recognition [13], and driver drowsiness recognition [14]. Virtual Personal Assistant (VPA) has been developed concurrently with the development of face recognition technology. Many big companies have already established their VPAs, such as Cortana, Siri, Alexa, and Google Assistant. For example, Facebook has launched Messenger. This personal assistant works by combining machine learning algorithms with contextual memory to give it the ability to learn and gain knowledge from examples by humans [15].

This research proposes an Android-based personal assistant system that can detect and recognize faces to assist people with blindness in their social life. The system uses the Haar-cascade classifier method to detect a face, while the LBPH is used in the recognition stage. This paper also discusses how static and dynamic user conditions can affect system performance. This paper is structured as follows. In Section 2, the proposed system is described. In Section 3, the result is discussed. And finally, Section 4 concludes the paper and discusses the future work.

2. **The Proposed System Scheme.** This work aims to create an Android-based face recognition system for people with blindness. The smartphone camera detects the face of the person in front of it, and then performs the process of detection and recognition. The output of this system is a voice indicating the person's name. The flowchart of the face recognition system is illustrated in Figure 1.

2.1. **The data acquisition.** The input data are in the form of videos taken in a real-time condition using a 16-megapixel smartphone camera with 1920×1080 video resolutions. Input data are face data from 30 non-identical respondents. The blind user's height is 158 cm, and the Point-of-View (POVIE) camera holder is hanging on the user's neck at a 20-degree angle. Data acquisition is carried out indoors during the day with light intensity 282.9-301 Lux.

The collected data consists of training and testing data. Training data are in the form of images that have been processed and stored in the database, whereas testing data are collected from the input video. System testing is performed in three scenarios, i.e., 1) each respondent walks straight towards the user who is standing still, 2) each respondent walks diagonally towards the user who is standing still, and 3) the user walks straight towards a respondent.

Each scenario is performed three times with a varying distance from 1 to 2.5 meters. The distance of the camera to the respondent's face can be calculated using Equation (1).

$$B = \frac{A}{\cos\theta} \tag{1}$$

where A = Distance between user and respondent; B = Distance between camera and respondent's face; θ = Point-of-view angle.

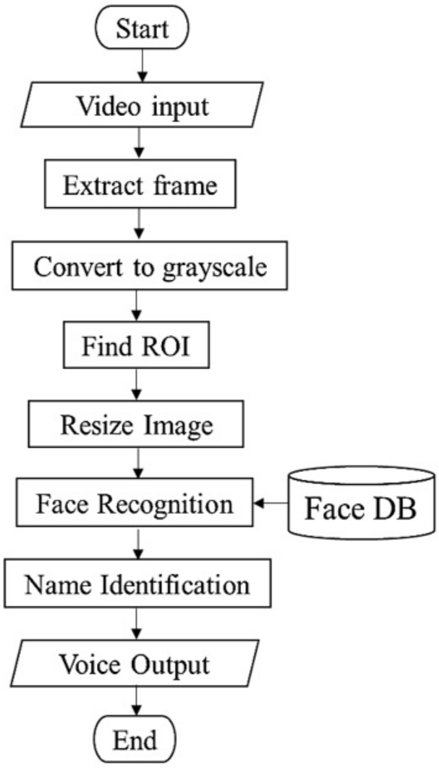


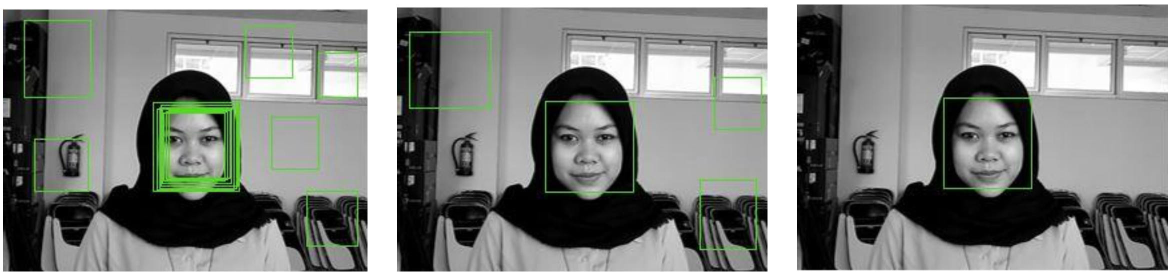
FIGURE 1. Flowchart of the face recognition stage

2.2. **Data storage.** This step is conducted as a training process of algorithm and storing the images into the database. The first step is frame extraction from the input video. The extracted frames are RGB images of 1920×720 pixels. Then, the frames are converted to grayscale because the Haar-cascade classifier algorithm only takes grayscale images as input. The next step is implementing the Haar-cascade classifier to find the Region of Interest (ROI). The algorithm uses a sliding window with a continuously reduced size to scan the image and find the Haar features, as seen in Figure 2.

FIGURE 2. Illustration of Haar-cascade classifier

After one scanning cycle ends, the box's size for the sliding window will be resized and re-scan the image as before. The sliding window size reduction is set using the *ScaleFactor* parameter, which is assigned to 1.2. Additionally, the *minNeighbors* parameter is set to 5 to minimize the system's possibility of detecting non-face objects with face-like features. The *minNeighbors* value is determined after carrying out several experiments, which can be seen in Figure 3.

The detected ROIs are displayed with a green bounding box. Because the detection is executed five times with different sizes of sliding windows, the face ROI sizes vary for each detection result. For equal scaling of image size, the resizing process is performed to

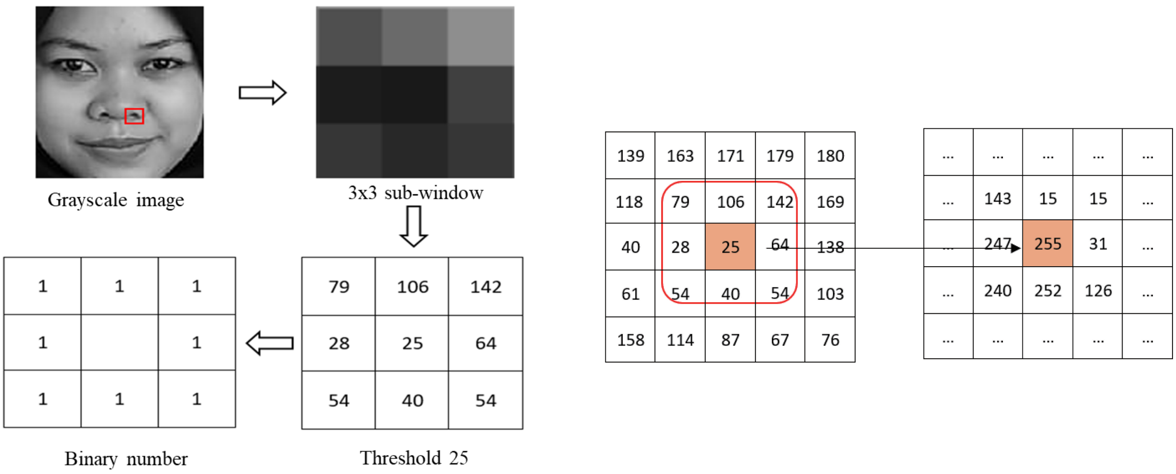


(a) (b) (c)

FIGURE 3. The result of face detection using *minNeighbors*: (a) 0; (b) 3; (c) 5

change the image size from 152×182 pixels to 110×128 pixels. This process reduces the size of the image without causing loss of information in the image, thereby speeding up the recognition process without reducing accuracy. The results are saved to the database and given an ID label according to the name of each person, which will be used for the face recognition process using the LBPH algorithm.

2.3. **Face recognition stage.** The face recognition stage is performed using the LBPH algorithm. The first step in LBPH is to clarify the original image that accentuates facial features by dividing the image into 3×3 sub-windows and converting it to LBP code. After that, the 8-bit binary number is converted into a decimal value placed on the central pixel value, which is the pixel from the original image. The process of conversion from the grayscale image to the LBP pixel value is shown in Figure 4.



(a) (b)

FIGURE 4. Conversion process from (a) grayscale to LBP codes, and (b) original pixel value to decimal value of LBP codes

The LBP pixel value image obtained from the last step is converted into a histogram by dividing the image into several grids using Grid X and Grid Y parameters. This step produces several histograms, which would be combined to create one histogram representing the whole image. Figure 5 illustrates the histogram extraction process.

It is necessary to compare the histograms of the images in the database with the input image histogram. The matching process is completed by calculating the distance between two histograms using the Chi-square test, which formula is given in Equation (2).

$$d(H_1 + H_2) = \sum_i \frac{(H_{1_i} - H_{2_i})^2}{H_{1_i}} \tag{2}$$

H_1 and H_2 are the two compared histograms, and i refers to the ith bin in the histogram. The output of the algorithm is the ID from the image in the database with the closest histogram. The algorithm also returns the calculated distance between the histograms. The flowchart for the image matching process is shown in Figure 6.

The last step is to compare the distance with a threshold value. The algorithm successfully recognizes faces if the difference between the input image and an individual image in the database is below the threshold, whose value is determined based on an experiment.

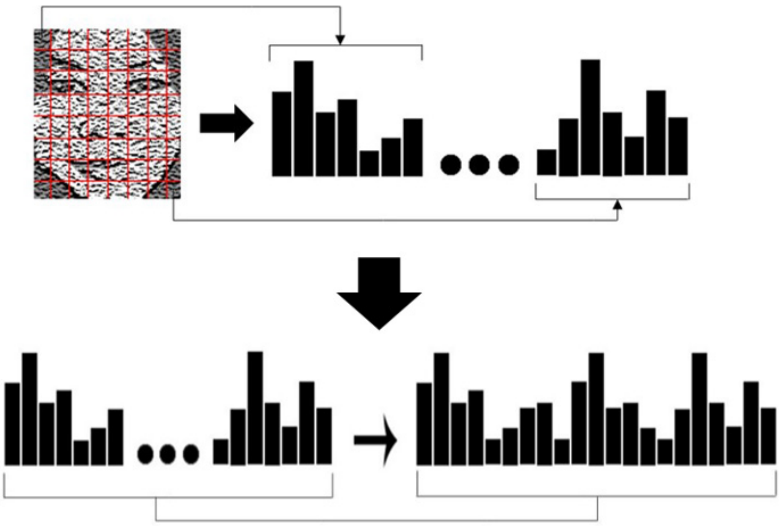


Figure 5. Histogram extraction process

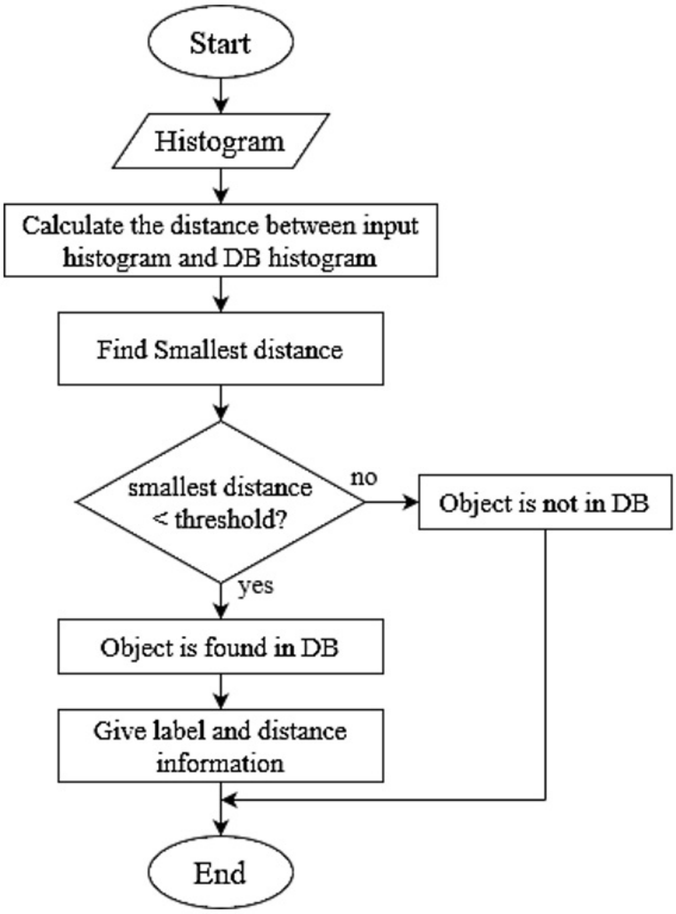
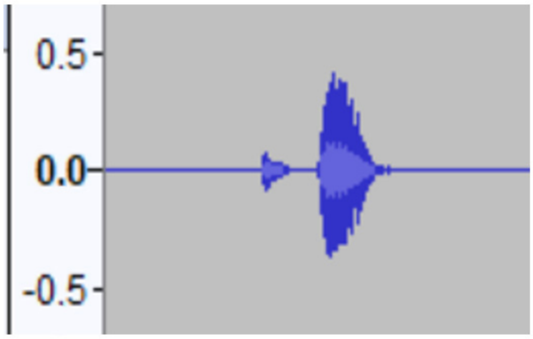


Figure 6. Flowchart for image matching process

The output of this system results from identification in the form of a voice reading out the respondent's name. The face ROI is also marked with a bounding box accompanied by the name ID of the respondent. An example of the recognition result is shown in Figure 7.

(a)



(b)

FIGURE 7. Example of recognition result: (a) Bounding box containing face region, and (b) sound spectrum of the respondent's name

3. **Result and Discussion.** Several experiments are carried out to obtain the best parameters for the highest system accuracy. Two parameters of the LBPH algorithm, Grid X and Grid Y, and different threshold values between 45 and 90 are tested on the system to find the optimal values. Table 1 shows the average system accuracy using different thresholds and grid values for all scenarios.

TABLE 1. Average system accuracy using different thresholds and grid values

T	**7 × 7 grid**			**8 × 8 grid**			**9 × 9 grid**		
	1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
45	75.73	64.95	77.4	39.24	19.51	39.89	9.68	3.35	5.43
50	80.73	75.23	85.14	56.64	33.1	51.71	17.69	6.19	15.61
55	85.68	78.01	92.64	62.77	47.64	61.79	29.32	12.11	26.31
60	87.08	81.15	94.37	74.59	57.83	71.48	38.38	19.55	37.32
65	88.97	81.65	94.93	82.03	72.19	81.39	43.68	27.4	46.21
70	89.53	81.89	95.15	85.2	78.92	85.64	55.67	40.29	58.92
75	89.53	81.89	95.15	88.79	83.17	87.59	63.69	50.84	64.43
80	89.53	81.89	95.15	91.16	83.82	89.60	75.09	59.91	72.56
85	89.53	81.89	95.15	92.19	86.47	91.13	78.68	65.66	78.13
90	89.53	81.89	95.46	93.3	86.7	92.33	82.73	70.9	84.68

Based on the test results, the 8 × 8 grid is more optimal than the 7 × 7 grid and 9 × 9 grid. Higher accuracy is obtained in the first two tests compared using an 8 × 8 grid, while a 7 × 7 grid gives the highest accuracy in the third testing condition only. Meanwhile, the 9 × 9 grid has the lowest accuracy in all three test conditions compared to other grids. The 8 × 8 grid shows better performance than smaller grids and larger grids because it displays 16,384 (8 × 8 × 256) features from each class. The 7 × 7 grid has fewer (12,544) features; hence, the system's difference is not fully perceived. The 9 × 9 grid has too many (20,736) features, which caused the system to get excessive similarity between classes.

The result also shows that the system performs better using the 90-threshold value. This threshold value functions as the maximum limit of the difference between the input image and an image in the database. Further testing shows lower accuracy using a threshold below 45, while a threshold above 90 does not show changes in inaccuracy. Camera conditions, face position, and camera movement also affect the recognition rate. In the first test, the user did not move; hence the position of the camera remained steady and could capture stable images. The second test is similar to the first test, but the respondents are walking horizontally towards the user, thus affecting the angle of the faces. Whereas in the third test, the user walked towards the immobile respondents, which caused the camera to sway and blurred the image. All these factors affect the detection performance of the face ROI.

4. **Conclusions.** A face recognition system for blind people has been developed and tested in this study with three scenarios, i.e., each respondent walks straight towards the user who is standing still; each respondent walks diagonally towards the user who is standing still, and the user walks straight towards a respondent. System performance is based on accuracy values by studying the effect of threshold values and grid parameters. The results showed that the threshold of 90 shows the best performance for all scenarios and all grid parameters. As for the grid parameters, the 8×8 grid shows better performance than the 7×7 and 9×9 grids at the threshold of 90. For the best conditions, the accuracy of scenarios 1, 2, and 3 is 93.3%, 86.7%, and 92.33%, respectively. As future works, different testing methods will be carried out and will be developed to recognize more than one face at a time.

Acknowledgment. This research is supported by the LPPM – Universitas Hasanuddin (UNHAS) via PDUPT-Ristekdikti Grant 2020, Indonesia and AIMP Research Group.

REFERENCES

[1] T. Ahmed, R. Hoyle, K. Connelly, D. Crandall and A. Kapadia, Understanding the physical safety, security, and privacy concerns of people with visual impairments, *IEEE Internet Computing*, vol.21, no.3, pp.56-63, DOI: 10.1109/MIC.2017.77, 2017.

[2] S. Deniz et al., Computer vision for attendance and emotion analysis in school settings, *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.0134-0139, DOI: 10.1109/CCWC.2019.8666488, 2019.

[3] S. Yeung et al., A computer vision system for deep learning-based detection of patient mobilization activities in the ICU, *NPJ Digital Medicine*, vol.2, no.11, DOI: 10.1038/s41746-019-0087-z, 2019.

[4] B. Arshad, R. Ogie, J. Barthelemy, B. Pradhan, N. Verstaevel and P. Perez, Computer vision and IoT-based sensors in flood monitoring and mapping: A systematic review, *Sensors*, vol.19, no.22, DOI: 10.3390/s19225012, 2019.

[5] Indrabayu, I. Nurtanio, I. S. Areni, S. R. A. Bugiwati, A. Bustamin and M. Rahmatullah, A portable cattle tagging based on muzzle pattern, *International Journal of Interactive Mobile Technologies (iJIM)*, vol.14, no.13, pp.134-149, DOI: ijim.v14i13.13237, 2020.

[6] L. Galarza, H. Martin and M. Adjouadi, Integrating low-resolution depth maps to high-resolution images in the development of a book reader design for person with visual impairment and blindness, *International Journal of Innovative Computing, Information and Control*, vol.14, no.3, pp.797-816, 2018.

[7] Y. Jin, J. Kim, B. Kim, R. Mallipeddi and M. Lee, Smart cane: Face recognition system for blind, *Proc. of the 3rd International Conference on Human-Agent Interaction*, pp.145-148, DOI: 10.1145/2814940.2814952, 2015.

[8] L. B. Neto, F. Grijalva, V. R. M. L. Maike, L. C. Martini, D. Florencio, M. C. C. Baranauskas, A. Rocha and S. Goldenstein, A Kinect-based wearable face recognition system to aid visually impaired users, *IEEE Trans. Human-Machine Systems*, vol.47, no.1, pp.52-64, DOI: 10.1109/THMS.2016.2604367, 2016.

[9] V. Aza, Indrabayu and I. S. Areni, Face recognition using local binary pattern histogram for visually impaired people, *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp.241-245, DOI: 10.1109/ISEMANTIC.2019.8884216, 2019.

[10] T. J. Alhindi, S. Kalra, K. H. Ng, A. Afrin and H. R. Tizhoosh, Comparing LBP, HOG and deep features for classification of histopathology images, *International Joint Conference on Neural Network (IJCNN)*, DOI: 10.1109/IJCNN.2018.8489329, 2018.

[11] A. Özdil and M. M. Özbilen, A survey on comparison of face recognition algorithms, *2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT)*, DOI: 10.1109/ICAICT.2014, 2014.

[12] A. Priadana and M. Habibi, Face detection using Haar Cascades to filter selfie face image on Instagram, *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, pp.6-9, DOI: 10.1109/ICAIIT.2019.8834526, 2019.

[13] D. Yang, A. Alsadoon, P. W. C. Prasad, A. K. Singh and A. Elchouemi, An emotion recognition model based on facial recognition in virtual learning environment, *Procedia Computer Science*, vol.125, pp.2-10, DOI: 10.1016/j.procs.2017.12.003, 2018.

[14] Indrabayu, S. K. Mufti and I. S. Areni, Car driver drowsiness recognition Android-based system, *IOP Conf. Series: Materials Science and Engineering*, DOI: 10.1088/1757-899X/619/1/012021, 2019.

[15] K. Wagner, *Facebook's Virtual Assistant 'M' is Super Smart. It's also Probably a Human*, https://www.vox.com/2015/11/3/11620286/facebooks-virtual-assistant-m-is-super-smart-its-also-probably-a-human, Accessed on 30 Mar. 2020.