Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 3, September 2024, pp. 556-565
ISSN: 2338-3070, DOI: 10.26555/jiteki.v10i3.29266

Implementation of Fisherface Algorithm for Eye and Mouth Recognition in Face-Tracking Mobile Robot

Ahmad Zarkasi, Huda Ubaya, Kemahyanto Exaudi, Ades Harafi Duri
Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya

ARTICLE INFO

Article history:

Received June 28, 2024
Revised September 01, 2024
Published September 12, 2024

Keywords:

Facial recognition;
Fisherface algorithms;
Face detection;
Face tracking;
Mobile Robot

ABSTRACT

Facial recognition is an artificial intelligence algorithm that distinguishes one face from another by capturing facial patterns visually. This recognition specifically detects and identifies individuals based on facial features by scanning the entire face. Several methods are used for facial detection, including facial landmarks points, Local Binary Patterns Histograms (LBPH), and Fisherface. In the context of this research, Fisherface is used to reduce the dimensionality of facial space in order to obtain image features. The method is insensitive to changes in expression and lighting, leading to better pattern classification and making it suitable for implementation on mobile devices such as robot vision. Therefore, this research aimed to measure the response time speed and accuracy level of pattern recognition when implemented on mobile robot devices. The results obtained from the accuracy testing showed that the highest accuracy for face detection process was 90%, while the lowest was 78.3%. In addition, the average execution time (AET) for the fastest process was 1.63 seconds and the slowest was 1.72 seconds. For pattern recognition, the statistics showed 90% accuracy, 100% precision, 81.81% recall, and F-1 score of 89.5%. Meanwhile, the longest execution time was 0.084 seconds and the fastest was 0.064 seconds. In face tracking process, the mobile robot movement was based on real-time pixel sizes, determining x and y values to produce the center of face region.

Corresponding Author:

Ahmad Zarkasi, Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya
Al Gazali Mosque Road, Bukit Lama, Ilir Bar. I, Palembang City, South Sumatra, 30128, Indonesia
Email: zarkasi98@gmail.com

1. INTRODUCTION

Facial recognition is an artificial intelligence algorithm that distinguishes one face from another by capturing facial patterns visually. In general, this visual information is stored in memory for future processing [1], [2]. The technology has been widely developed and applied in image-based applications, including pattern verification, attendance tracking, communication, and medical fields. Facial recognition process operates by scanning the full face to recognize and identify individuals based on their facial traits [3], [4]. After scanning, the data collected is converted into a specified format, which can then serve as a reference pattern in database. When a new face pattern is detected during scanning, the system compares the result with an existing database to identify matches. Image processing methods allow computers to monitor, recognize, and analyze face image in greater depth [5], [6], [7]. The system focuses on features such as the eyes and nose and then presents results synchronized with the input pattern [8]. This process is consistent with the working principles of various computer vision technologies. Several methods are used in facial recognition, including facial landmark points [9], [10], Local Binary Patterns Histograms (LBPH) [11], [12], [13], and Fisherface [14], [15].

Landmark detection identifies the specific position on a face that represent important facial features, connecting these points to outline the face's structure [16], [17]. However, this landmark method requires significant training time to achieve accuracy. LBPH method converts facial pixel textures into binary patterns, using a central pixal as a reference for surrounding values [18], [19]. Typically, the method offers fast and

efficient computation but may struggle with accuracy under varying light conditions due to its simplicity. Meanwhile, Fisherface method [20], [21], [22] combines Principal Component Analysis (PCA) [23], [24], [25] with Fisher's Linear Discriminant (FDL) [26], [27] to reduce the dimensionality of facial space to extract image features. Moreover, the distance ratio between classes is increased compared to the intra-class distance of feature vectors using Linear Discriminant Analysis (LDA) method [28], [29], [30], and the dimensions are reduced by applying PCA calculations. This process is intended to optimize the ratio of pattern distribution between classes. As the ratio of classes grows, the resultant feature vector becomes less sensitive to changes in expression and illumination, leading to improved pattern classification and suitability for mobile device applications, such as robot vision.

Robot vision system symbolizes human-robot interaction and has advanced dramatically in the recent decade [31], [32], [33]. This model is a visual processing system that combines distance sensors and lasers, representing the most precise navigation methods available to robots. However, most of these robotic systems have various flaws [34], [35], such as poor performance, which are unsuitable for use in embedded systems with real-time requirements [36], [37], [38]. Some previous investigations provided the required capability, but not as an embedded device because the current procedure feels suboptimal. Furthermore, image processing in embedded systems is not well established. To address these limitation, this research proposes facial recognition system using Fisherface method on a mobile robot equipped with an embedded platform. The research aims to determine the response time speed and pattern recognition accuracy of this system when implemented on mobile robot devices. Given the resource constraints of embedded platforms, Fisherface is adapted to develop a more specific facial test pattern, focusing on face and lips. The robot is programmed to move according to recognized facial pattern, with a specific focus on analyzing dynamic moving eyes and lips.

2. METHODS AND DESIGN

The steps followed in this research were consistent with methods reviewed in the provided framework. These steps comprised different stages of system requirements analysis. For instance, mobile robot was used with a Raspberry Pi minicomputer that functioned as facial data processor for the mobility system of robot. In general, Raspberry technology was used to develop an operating system capable of identifying faces [39], [40] and tracking the movements of people [41], [42], [43]. The system was tested in real-time on datasets of facial image. Haar-like feature from Viola-Jones method was used in this research to detect face features [44], [45], [46]. Following the process, Fisherface method was used for face recognition. For the hardware component, OpenCR microcontroller was equipped with two DC (direct current) motors and two servo motors. Moreover, the methodology consisted of a preprocessor, facial recognition process, facial tracking, and validation of mobile movement of robots. The general system design that was adopted was described as shown in Fig. 1.

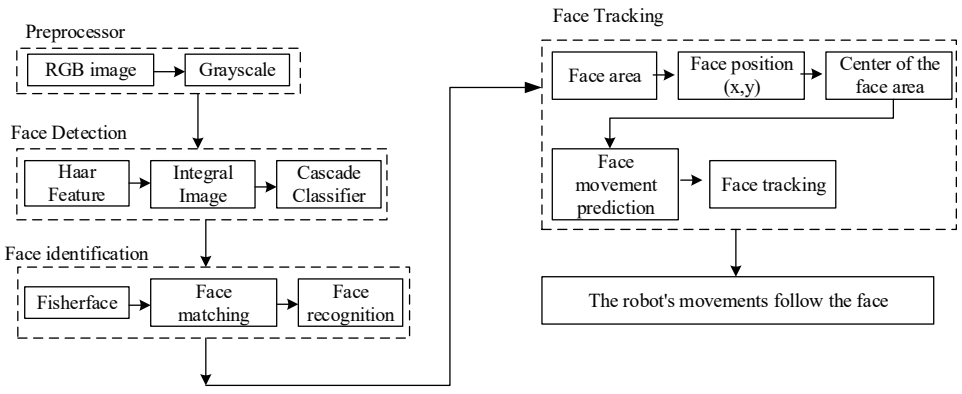

Fig. 1. System Design

In the research, the preprocessor stage comprised of mapping the environment which included capturing image of the research subjects. This image was originally in RGB (Red, Green, and Blue) format, and it was later converted to show the dimming level using (1).

$$Grayscale = \frac{R + G + B}{3} \quad (1)$$

where, R is red image, G is green image, B is blue image. The two major stages in this system included face detection and recognition. Fig. 2 showed a flowchart for the software system that processed face patterns. The

initial stage in face detection was a collection of facial patterns that were present in the workspace. Following this process, preprocessing and face detection were performed using Haar-like feature method. In preprocessing, RGB image data was turned into grayscale image, and then Haar-like features were used to identify particular portions of input image, which comprised a mix of white and black patches. The difference between the normal pixel values in the black and white sections was calculated which allowed the establishment of Haar features existence or not. Moreover, the object-detecting feature in the program was Haar Cascade [47], [48], which was a classifier for saving training results depending on diverse information, such as positive and negative image. This feature was used to identify the presence or absence of face in the camera frame. When face was discovered, the second step was to recognize facial pattern.

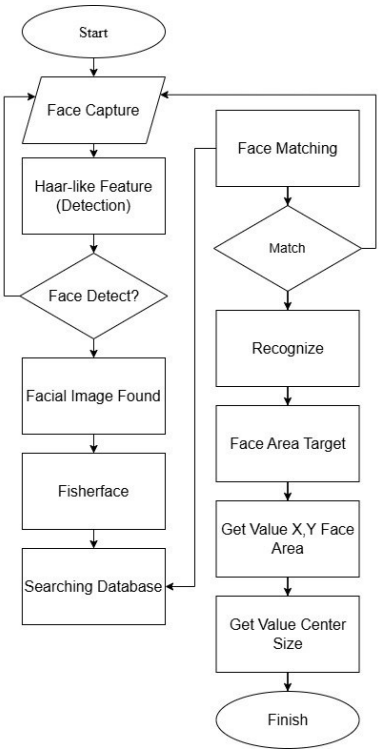

Fig. 2. Flowchart System

Additionally, Fisherface method handled faces through several phases of data processing. Facespace was first modified as part of LDA process, and an average face was generated for each class. The distribution matrix in and between classes was later computed and projected onto PCA projection matrix, followed by projection into eigenvalues as well as eigenvectors. The last process in this stage was calculating Fisherface weight for each face image. After the data was analyzed, it was compared to facial pattern data contained in the database. Several facial patterns were included in the database as reference patterns from the front, right, and left positions. After recognizing the pattern, the following step was to identify face region to retrieve pixel values (x, y) for facial pattern. When face appeared in the camera frame, its position was determined by finding x and y values, which showed the center of facial region.

The next step was face detection which aimed to extract face from a image. Fig. 3 showed facial detection procedure, starting with Haar feature analysis for feature extraction and categorization [49], [50]. This square-shaped feature showed the detail embedded in the image. The feature served as image identifier by centralizing basic integers of the feature. Following this process, computations were conducted swiftly using the computer because this feature simply used information from pixels in the rectangle shape. Image integration procedure then evaluated the general feature similarity. This procedure later summed pixel intensity for a specific region in a way that the average pixel intention for the dark region was decreased to the mean pixel strength for the bright region.

The value at (x, y) was the sum of pixels from the upper left corner to point (x, y). To obtain the average number of pixels in a square region, point value (x, y) was divided by the square. Additionally, an integral image was used to speed up the computation process to analyze when the feature was present in a large number of Haar features on image. Haar cascade classifier procedure was a machine learning method for reviewing

objects in image and movies. This method was separated into four stages including Haar feature selection, generating integral image, AdaBoost training, and cascading classifier. In the context of this research, Fig. 4 showed Haar cascade classification method. Fisherface algorithm was used by facial recognition system in this section. Facial recognition method consisted of numerous steps that allowed the method to recognize the features of a human face including the eyes, nose, and mouth. This method determined the position, size, and posture of an identifiable face. Following the process, system software exported face curves in the submillimeter range to produce templates. When the template was constructed, it was automatically transformed into machine-readable code. The verification procedure matched facial data individually until the face was recognized. The final stage in the process was detection, which matched the acquired face image to the whole face image stored in the database.

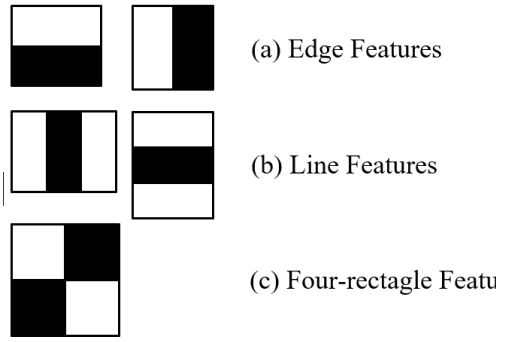

Fig. 3. Haar Feature

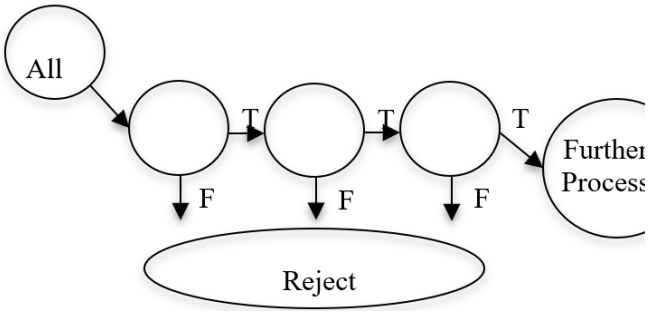

Fig. 4. Haar cascade classifier

Fisherface algorithm was facial recognition system that combined two methods including PCA and LDA. This method aimed to minimize dimensions in PCA computations by raising the scatter ratio to the class distance of feature vector that underpinned LDA method. The categorization was better when the class was more homogenous and feature vector was less susceptible to variations in illumination and facial expression. In this research, LDA aimed to find a linear projection that maximized the interclass covariance matrix, leading to more evenly distributed class members and eventually better recognition performance. To calculate LDA, the following steps were considered, which included 1*) Training sets were transformed into column vectors (facial space). 2*) The average values of each class and face space were combined to obtain a class average face. 3*) The distribution matrix was determined across classes (between-class scatter-matrix, S_B) and in classes (in-class S_W). 4*) The distribution matrix (S_W and S_B) was projected onto PCA projection matrix. 5*) The individual and personal vector values using the distribution matrix were determined. 6*) Eigenvector by eigenvalue measurement was used to calculate Fisher projection matrix. 7*) Each Fisherface was compared to face image in the training set.

After facial recognition operations were conducted, the robot was activated by mobile using vision specification, shown in Fig. 5. Mobile Robot Vision (MRV) recognized objects and controlled robot movement with image processing and computer vision technologies. Following the discussion, this vision robot required processors, inputs, and outputs. The processor device was CPU (Central Processing Unit), which interpreted image and outputs robot motion control in its implementation. Another alternative was to use small computers, such as Raspberry Pi, to replace larger CPU gear. Currently, Raspberry Pi was commonly used to replace CPU operations in computer technology and robotic vision. In this research, face recognition and detection were performed during image processing in MRV to guarantee that image in the database matched image captured by the real-time camera.

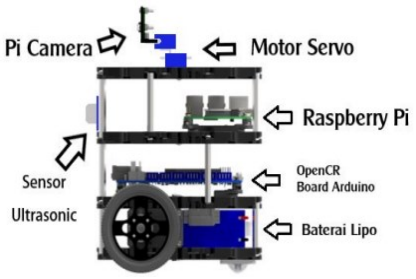

Fig. 5. Mobile robot design

3. RESULTS AND DISCUSSION

The section showed the outcomes of testing and analysis across various methods scaled to the project which included facial recognition analysis, robotic systems, and software/hardware integration. The first test was performed on the preprocessor component, which comprised two phases of RGB and grayscale imagery. Fig. 6 showed the conversion of RGB image to grayscale in this process.

The Figure showed the value for face's total RGB which included pixel value (x, y) 170×170, also shown in Fig. 7. Additionally, Fig. 8 showed a 4×4 pixel value followed by a 2×2 RGB value (x, y) used as an analysis sample which was shown in Fig. 8 as a blue box. Following this process, the value was transformed into a grayscale pixel value. The size of RGB conversion to a 2×2 Grayscale value was 219, 196, 220 and 212. Following the preprocessing process, Haar method was used to examine face detection for other matrix conversion values used by the software. During the process, Haar-like feature detected a specific region of the input image. Facial recognition procedure included counting all the pixels in the closest region at a given place, which contained a combination of white and black patches. Moreover, calculating the difference between typical pixel values in the black and white region helped to show any features were present or absent because the function of Haar Cascade was to detect objects. In this research, face tracking in the camera frame occurred immediately after face was correctly identified. Fig. 9 showed the counting of integral image using data produced from grayscale data.

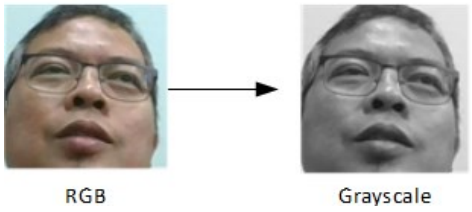

Fig. 6. RGB to Grayscale

Fig. 7. RGB size 170×170

[197, 229, 230]	[180, 209, 200]	[107, 209, 220]	[187, 270, 220]
[198, 230, 231]	[185, 210, 241]	[197, 129, 240]	[197, 229, 230]
[168, 210, 201]	[178, 235, 239]	[157, 240, 230]	[147, 129, 210]
[197, 209, 210]	[197, 129, 210]	[166, 200, 130]	[194, 229, 228]

219	196
220	212

Fig. 8. RGB conversion to a 2×2 Grayscale value

137	137+142
137+58	137+142+58+169

→

137	279
194	506

Fig. 9. Result of counting of integral image

Fig. 9 showed a calculation process to find the integral value of each pixel image (x, y) and the calculation result for a 2×2 matrix sample. The result of counting of integral image are 137, 297, 194 and 506 The next step was face detection, which was conducted to confirm the work and effectiveness of the system. In addition, the results of face detection process were shown in Fig. 10. In the image, the position of face was marked by a four-square red frame. This marking process showed that facial recognition process worked perfectly. In the examination, about 60 tests were performed at different distances. The total test results were shown in face detection examination graphics in Fig. 11.

The Figure showed data from 10 properly identified faces at distances of 30 cm and 40 cm. However, at distances of 50 cm and 60 cm, there were eight identified faces, signifying that four faces were not adequately detected. At a distance of 70 cm, back only detected 6 faces, leaving 4 undetermined. Five faces were recognized at a distance of 80 cm, implying that five more faces were not shown. These results showed that as the distance became farther between the camera and face object, more undiscovered faces occurred. In face detection test using Haar features, a 90% success rate was achieved. The testing was conducted on patterns located precisely in the ideal detection region, which was from 30 cm to 60 cm. Meanwhile, when the testing was conducted in the range of 30 cm to 80 cm, an average success rate of 78.3% was achieved. The result occurred because the largest error appeared at the farthest distance outside the working region of the system,

which was between 70 cm and 80 cm. Understanding the time speed of facial pattern detection was very important because the systems built for this research were based on embedded platforms. In essence, all data processing devices should be capable of supporting system performance. In this research, the execution time was measured based on face detection time at the data capture distance, with the results of the graphic shown in Fig. 12.

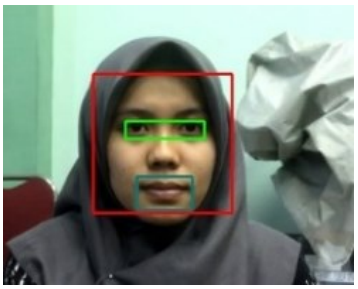

Fig. 10. Result of face detection

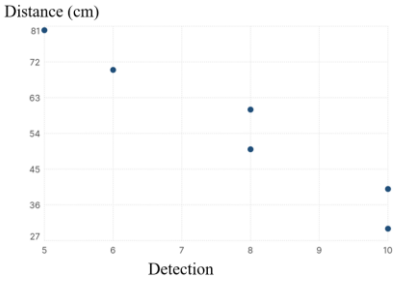

Fig. 11. Face detection test graphics

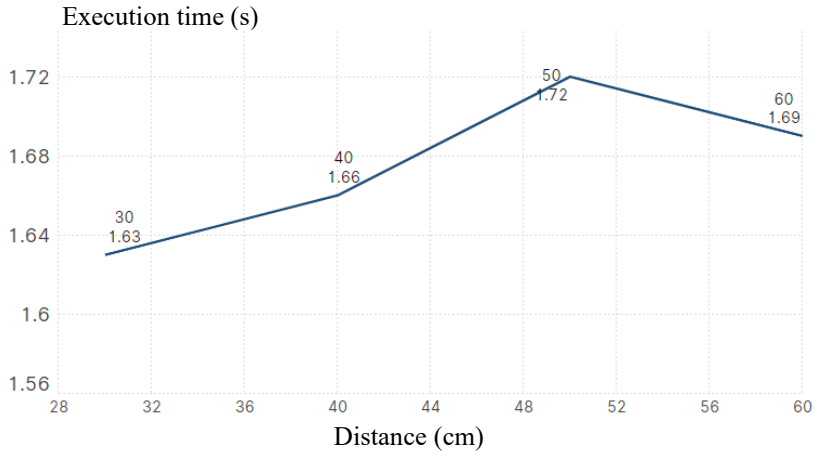

Fig. 12. Execution time graphics

In the Figure, 60 facial patterns were used as test samples and at a distance of 30 cm, AET was 1.63 seconds. For a distance of 40 cm, the execution time increased to 1.66 seconds. At 50 cm, AET was 1.72 seconds and at 60 cm, AET was 1.69 seconds. The next step in this examination was face detection for the eye and mouth. There were two green squares in Fig. 10, which were right on the eye and nose region, marked by a purple box. In this research, facial database was developed before testing Haar-like feature method in real time with cameras in various face postures. Fig. 13 showed a sample of faces saved in the system database.

Fig. 13. Face database

Facial recognition method of Fisherface used a sample of identifiable faces of two people. During this process, both faces were highly processed using Raspberry Pi minicomputers. Each person took 80 facial shots, summing up to a total of 160 face image used as sample data. Moreover, the size of the facial image varied depending on the position of face relative to the camera. When face was farther from the camera frame, the image size appeared smaller, and as the camera frame moved closer, the image size increased. The outcome of the process was 9 True Positive, 2 False Negative, and 9 True Negative. After this process, computed accuracy, precision, recall, and F-1 score results were analyzed. From the calculations, accuracy was 90%, precision 100%, recall 81.81%, and F-1 score 89.5%. An examination was conducted after the procedure on an unidentified pattern of face. The results were 8 True Positive, 2 False Negative, and 10 True Positive, having an accuracy of 90%, 100% precision, 80% recall, and F-1 scores of 80%. Additionally, the result of facial recognition testing and analysis showed that the system was working effectively with a level of accuracy above 85%.

After face detection was completed, the system of the mobile robot was analyzed. Face tracking was the next stage using a camera with a recording feature of facial objects in real time. When face was in the camera frame, the position was estimated by finding x and y values, which generated the center of facial region. In addition, two servo motors moved horizontally and vertically while tracking the region of face. During this process, DC motor movement system allowed the mobile robot to move closer to face individual. Table 1 showed the test results for DC motor and servo motor in the process.

Table 1. The test results for DC motor and Servo motor.

DC Motor				Servo Motor					
pixel (x,y)	Leff	Right	Robot Movement	Face Position		Servo Position		Movement	
				x	y	x'	y'	X	Y
100	1	1	Forward	300	300	90	60	Center	Center
				305	305	90	60	Center	Center
125	1	1	Forward	200	190	100	55	Right	Center
				205	195	100	55	Right	Center
150	-1	-1	Backward	220	185	105	50	Right	Upper
				225	190	105	50	Right	Upper
175	-1	-1	Backward	355	330	80	70	Left	Lower
				360	340	75	75	Left	Lower

In DC motor analysis, the mobile robot traveled about the size of acquired pixel (x, y). When the values of pixels ranged from 100 to 140, the mobile robotic traveled forward, but when pixels ranged from 160 to 180, the robot moved backward. Moreover, the position of face object determined the direction of movement of servo motor, according to the analysis. Each position had two servos including x' and y'.x' along x-axis, while y' moved toward y-axis. The servo had four directions which included central, right, left, upper, and lower. The size of pixel (x, y) was used to determine the position and direction of servo motor. Specifically, the size of pixel was (300, 300), implying that servo x' moved to 90^0 and servo y' to 60^0. This process affected the direction of movement of two servo motors toward the center, and to the position of pixels (360, 340), where X servo moved to left and Y servo to lower.

4. CONCLUSION

In conclusion, the implementation of Fisherface algorithm for eye and mouth recognition produced results that were consistent with the system design. Every testing process conducted during this research was successful. In addition, face detection accuracy testing using Haar features achieved a 90% success rate when the testing pattern was placed in the recommended working region of the system of about 30 cm to 60 cm. Testing inside the 30 cm to 80 cm range achieved a success percentage of 78.3%. The highest accuracy occurred at the farthest distance beyond the operating region of the system, which was between 70 cm and 80 cm. Additionally, the execution time tests at distances ranging from 30 cm to 60 cm achieved AET of 1.63 seconds, 1.66 seconds, 1.72 seconds, and 1.69 seconds at a distance of 60 cm, respectively. The pattern recognition statistics showed 90% accuracy, 100% precision, 81.81% recall, and an F-1 score of 89.5%. In this research, pattern 1 had the longest execution time at 0.084 seconds, while pattern 45 had the shortest time of 0.064 seconds. Mobile robot progressed when pixel value acquired during face tracking test was between 100 and 140. During this research, a camera was used to collect facial items in real-time, and the position of face was computed by identifying x and y values, leading to the center of facial region.

Acknowledgments

The authors are grateful to DIPA of Public Service Agency of Universitas Sriwijaya 2023 for funding this research through Nomor SP DIPA-023.17.2.6775151/2023, On November 30, 2022 in accordance with the Rector's Decree Number: 0188/UN9.3.1/SK/2023, On April lB, 2023. Also, the authors are also grateful to the Faculty of Computer Science, Faculty of Engineering, Strategic Pervasive Computing and Intelligent Embedded Systems Research Group, Intelligence System Research Group, and COMNETS Research Group, as well as Universitas Sriwijaya's Embedded Systems, Control System, and Robotic Laboratory.

REFERENCES

[1] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors (Switzerland)*, vol. 20, no. 2, 2020, https://doi.org/10.3390/s20020342.

[2] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021, https://doi.org/10.1016/j.neucom.2020.10.081.

[3] M. Taskiran, N. Kahraman, and C. E. Erdem, "Face recognition: Past, present and future (a review)," *Digit. Signal Process. A Rev. J.*, vol. 106, p. 102809, 2020, https://doi.org/10.1016/j.dsp.2020.102809.

[4] F. Marcolin, E. Vezzetti, and M. G. Monaci, "Face perception foundations for pattern recognition algorithms," *Neurocomputing*, vol. 443, pp. 302–319, 2021, https://doi.org/10.1016/j.neucom.2021.02.074.

[5] A. Sharifara, M. Shafry, M. Rahim, and Y. Anisi, "A General Review of Human Face Detection Including a Study of Neural Networks and Haar Feature-based Cascade Classifier in Face Detection," *In 2014 International symposium on biometrics and security technologies (ISBAST),* pp. 73–78, 2014, https://doi.org/10.1109/ISBAST.2014.7013097.

[6] A. Zarkasi *et al.*, "Face Movement Detection Using Template Matching," in *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018*, Jan. 2019, pp. 333–338, https://doi.org/10.1109/ICECOS.2018.8605215.

[7] C. Peng, W. Bu, J. Xiao, K. chun Wong, and M. Yang, "An improved neural network cascade for face detection in large scene surveillance," *Appl. Sci.*, vol. 8, no. 11, 2018, https://doi.org/10.3390/app8112222.

[8] A. Zarkasi, H. Ubaya, K. Exaudi, A. Almuqsit, and O. Arsalan, "Robot Vision Pattern Recognition of the Eye and Nose Using the Local Binary Pattern Histogram Method," *Computer Engineering and Applications Journal*, vol. 12, no. 3, pp. 147–158, 2023, https://core.ac.uk/download/pdf/590237578.pdf.

[9] D. Spulak, R. Otrebski, and W. Kubinger, "Evaluation of PCA, LDA and fisherfaces in appearance-based object detection in thermal infra-red images with incomplete data," *Procedia Eng.*, vol. 100, pp. 1167–1173, 2015, https://doi.org/10.1016/j.proeng.2015.01.480.

[10] M. Taskiran, N. Kahraman, and C. E. Erdem, "Face recognition: Past, present and future (a review)," *Digital Signal Processing*, vol. 106, p. 102809, 2020, https://doi.org/10.1016/j.dsp.2020.102809.

[11] C. Singh, E. Walia, and K. P. Kaur, "Color texture description with novel local binary patterns for effective image retrieval," *Pattern Recognit.*, vol. 76, pp. 50–68, 2018, https://doi.org/10.1016/j.patcog.2017.10.021.

[12] R. S and Y. P. Gowramma, "Image Training and LBPH Based Algorithm for Face Tracking in Different Background Video Sequence," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 9, pp. 349–354, 2018, https://doi.org/10.26438/ijcse/v6i9.349354.

[13] X. Zhang and H. Zhao, "Hyperspectral-cube-based mobile face recognition: A comprehensive review," *Inf. Fusion*, vol. 74, pp. 132–150, 2021, https://doi.org/10.1016/j.inffus.2021.04.003.

[14] A. M. Jagtap, V. Kangale, K. Unune, P. Gosavi, "A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV," *In 2019 International conference on intelligent sustainable systems (ICISS)*, pp. 219-224, 2018, https://doi.org/10.1109/ISS1.2019.8907965.

[15] M. M. Ahsan, Y. Li, J. Zhang, M. T. Ahad, and K. D. Gupta, "Evaluating the Performance of Eigenface, Fisherface, and Local Binary Pattern Histogram-Based Facial Recognition Methods under Various Weather Conditions," *Technologies*, vol. 9, no. 2, p. 31, 2021, https://doi.org/10.3390/technologies9020031.

[16] Y. Yan *et al.*, "Fine-grained facial landmark detection exploiting intermediate feature representations," *Comput. Vis. Image Underst.*, vol. 200, p. 103036, 2020, https://doi.org/10.1016/j.cviu.2020.103036.

[17] Y. Jin, X. Guo, Y. Li, J. Xing, and H. Tian, "Towards stabilizing facial landmark detection and tracking via hierarchical filtering: A new method," *J. Franklin Inst.*, vol. 357, no. 5, pp. 3019–3037, 2020, https://doi.org/10.1016/j.jfranklin.2019.12.043.

[18] T. Song, L. Xin, C. Gao, T. Zhang, and Y. Huang, "Quaternionic extended local binary pattern with adaptive structural pyramid pooling for color image representation," *Pattern Recognit.*, vol. 115, p. 107891, 2021, https://doi.org/10.1016/j.patcog.2021.107891.

[19] X. Shu, Z. Song, J. Shi, S. Huang, and X. J. Wu, "Multiple channels local binary pattern for color texture representation and classification," *Signal Process. Image Commun.*, vol. 98, no. July, p. 116392, 2021, https://doi.org/10.1016/j.image.2021.116392.

[20] G. Cheng and Z. Song, "Robust face recognition based on sparse representation in 2D Fisherface space," *Optik (Stuttg).*, vol. 125, no. 12, pp. 2804–2808, 2014, https://doi.org/10.1016/j.ijleo.2013.11.042.

[21] M. Zhao, Z. Jia, Y. Cai, X. Chen, and D. Gong, "Advanced variations of two-dimensional principal component analysis for face recognition," *Neurocomputing*, vol. 452, pp. 653–664, 2021, https://doi.org/10.1016/j.neucom.2020.08.083.

[22] W. Ali, W. Tian, S. U. Din, D. Iradukunda, and A. A. Khan, "Classical and modern face recognition approaches: a

complete review," *In 2019 International conference on intelligent sustainable systems (ICISS)*, pp. 219-224, 2019, https://doi.org/10.1109/ISS1.2019.8907965.
[23] K. C. Paul and S. Aslan, "An Improved Real-Time Face Recognition System at Low Resolution Based on Local Binary Pattern Histogram Algorithm and CLAHE," *Opt. Photonics J.*, vol. 11, no. 04, pp. 63–78, 2021, https://doi.org/10.4236/opj.2021.114005.
[24] N. O'Mahony *et al.*, "Deep Learning vs. Traditional Computer Vision BT - Advances in Computer Vision," *Proceedings of the 2019 Computer Vision Conference (CVC),* vol. 1, no. 1, pp. 128-144, 2020, https://doi.org/10.1007/978-3-030-17795-9_10.
[25] G. M. Zafaruddin and H. S. Fadewar, "Face recognition using eigenfaces," *In Computing, Communication and Signal Processing: Proceedings of ICCASP*, pp. 855-864, 2018, https://doi.org/10.1007/978-981-13-1513-8_87.
[26] A. Tyagi, S. Deshmukh, G. Dindokar, S. Kale, M. Karale, and B. Dhakulkar, "IoT Based Smart Home Automation System," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 6, no. 3, pp. 811–822, 2020, https://doi.org/10.32628/cseit2063194.
[27] S. Bhutada and T. Iv, "Emotion Based Music (Emotify)," *International Journal of Advanced Research in Computer Science,* vol. 11, 2020, https://openurl.ebsco.com/EPDB%3Agcd%3A11%3A19554683/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A143484206&crl=c.
[28] Y. Ren, X. Xu, G. Feng, and X. Zhang, "Non-Interactive and secure outsourcing of PCA-Based face recognition," *Comput. Secur.*, vol. 110, p. 102416, 2021, https://doi.org/10.1016/j.cose.2021.102416.
[29] J. H. Shah, M. Sharif, M. Yasmin, and S. L. Fernandes, "Facial expressions classification and false label reduction using LDA and threefold SVM," *Pattern Recognit. Lett.*, vol. 139, pp. 166–173, 2020, https://doi.org/10.1016/j.patrec.2017.06.021.
[30] M. Ayyad and C. Khalid, "New fusion of SVD and Relevance Weighted LDA for face recognition," *Procedia Comput. Sci.*, vol. 148, pp. 380–388, 2019, https://doi.org/10.1016/j.procs.2019.01.046.
[31] A. Zarkasi, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Weightless Neural Networks Face Recognition Learning Process for Binary Facial Pattern," *Indones. J. Electr. Eng. Informatics*, vol. 10, no. 4, pp. 955–969, 2022, https://doi.org/10.52549/ijeei.v10i4.3957.
[32] S. Nurmaini, A. Zarkasi, D. Stiawan, B. Yudho Suprapto, S. Desy Siswanti, and H. Ubaya, "Robot movement controller based on dynamic facial pattern recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 2, p. 733, 2021, https://doi.org/10.11591/ijeecs.v22.i2.pp733-743.
[33] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Comput. Networks*, vol. 168, p. 107036, 2020, https://doi.org/10.1016/j.comnet.2019.107036.
[34] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, pp. 1–22, 2019, https://doi.org/10.1177/1729881419839596.
[35] A. Sardelis *et al.*, "2-Stage vision system for robotic handling of flexible objects," *Procedia CIRP*, vol. 97, pp. 491–496, 2020, https://doi.org/10.1016/j.procir.2020.07.008.
[36] K. Xue *et al.*, "Robotic seam tracking system based on vision sensing and human-machine interaction for multi-pass MAG welding," *J. Manuf. Process.*, vol. 63, no. December 2019, pp. 48–59, 2021, https://doi.org/10.1016/j.jmapro.2020.02.026.
[37] X. Lv, M. Su, and Z. Wang, "Application of Face Recognition Method Under Deep Learning Algorithm in Embedded Systems," *Microprocess. Microsyst.*, p. 104034, 2021, https://doi.org/10.1016/j.micpro.2021.104034.
[38] W. J. Song, "Hardware accelerator systems for embedded systems," *In Advances in Computers*, vol. 122, pp. 23-49, 2021, https://doi.org/10.1016/bs.adcom.2020.11.004.
[39] Y. Gurovich *et al.*, "Identifying facial phenotypes of genetic disorders using deep learning," *Nat. Med.*, vol. 25, no. 1, pp. 60–64, 2019, https://doi.org/10.1038/s41591-018-0279-0.
[40] E. Calgaro, N. Craig, L. Craig, D. Dominey-Howes, and J. Allen, "Silent no more: Identifying and breaking through the barriers that d/Deaf people face in responding to hazards and disasters," *Int. J. Disaster Risk Reduct.*, vol. 57, p. 102156, 2021, https://doi.org/10.1016/j.ijdrr.2021.102156.
[41] L. Maleš, D. Marčetić, and S. Ribarić, "A multi-agent dynamic system for robust multi-face tracking," *Expert Syst. Appl.*, vol. 126, pp. 246–264, 2019, https://doi.org/10.1016/j.eswa.2019.02.008.
[42] Y. Matsumura and K. Arai, "Influence of orthodontic appliances on visual attention to smiling faces by eye-tracking evaluation," *Orthod. Waves*, vol. 78, no. 4, pp. 135–142, 2019, https://doi.org/10.1016/j.odw.2019.08.002.
[43] P. Huang *et al.*, "Contribution of the mandible position to the facial profile perception of a female facial profile: An eye-tracking study," *Am. J. Orthod. Dentofac. Orthop.*, vol. 156, no. 5, pp. 641–652, 2019, https://doi.org/10.1016/j.ajodo.2018.11.018.
[44] T. Sree Sharmila, R. Srinivasan, K. K. Nagarajan, and S. Athithya, "Eye Blink Detection Using Back Ground Subtraction and Gradient-Based Corner Detection for Preventing CVS," *Procedia Comput. Sci.*, vol. 165, no. 2019, pp. 781–789, 2019, https://doi.org/10.1016/j.procs.2020.01.011.
[45] K. D. Ismael and S. Irina, "Face recognition using Viola-Jones depending on Python," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1513–1521, 2020, https://doi.org/10.11591/ijeecs.v20.i3.pp1513-1521.
[46] T. H. Obaida, A. S. Jamil, and N. F. Hassan, "Real-time face detection in digital video-based on Viola-Jones supported by convolutional neural networks," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 3, pp. 3083–3091, 2022, https://doi.org/10.11591/ijece.v12i3.pp3083-3091.

[47] R. Raj *et al.*, "Feature based video stabilization based on boosted HAAR Cascade and representative point matching algorithm," *Image Vis. Comput.*, vol. 101, p. 103957, 2020, https://doi.org/10.1016/j.imavis.2020.103957.
[48] S. R. Rudraraju, S. R. Rudraraju, and A. Negi, "Edge computing for visitor identification using eigenfaces in an assisted living environment," *In Assistive Technology for the Elderly*, pp. 235-248, 2020, https://doi.org/10.1016/B978-0-12-818546-9.00008-7.
[49] A. B. Shetty, Bhoomika, Deeksha, J. Rebeiro, and Ramyashree, "Facial recognition using Haar cascade and LBP classifiers," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 330–335, 2021, https://doi.org/10.1016/j.gltp.2021.08.044.
[50] Y. N. Lin, T. Y. Hsieh, J. J. Huang, C. Y. Yang, V. R. L. Shen, and H. H. Bui, "Fast Iris localization using Haar-like features and AdaBoost algorithm," *Multimed. Tools Appl.*, vol. 79, no. 45–46, pp. 34339–34362, 2020, https://doi.org/10.1007/s11042-020-08907-5.

BIOGRAPHY OF AUTHORS

Ahmad Zarkasi, was born in Palembang on August 25, 1979. In 2023, he obtained his Doctor in Informatics Engineering from the Sriwijaya University, Faculty of Engineering. His research interests include microprocessors, system-on-chip (SoC), embedded systems Engineering, and robotics. They include subjects such as WNNs in robotic systems and pattern recognition for robot vision. Currently, he is a lecturer in Department of Computer Engineering, Faculty of Computer Science, Sriwijaya University, Indonesia. He can be contacted at email: zarkasi98@gmail.com

Huda Ubaya, was born in Yogyakarta on June 16, 1981. He obtained his M.Sc. and B.Sc. in Computer Engineering from the Institut Teknologi Bandung, School of Electrical Engineering and Informatics. Currently, he is in the process of completing his doctorate at the Faculty of Engineering, Sriwijaya University. His research interests include Mobile and Pervasive Computing, Internet of Everything, Embedded Systems, Distributed Ledger Technology, and Blockchains. Currently, he is a lecturer in Department of Computer Engineering, Faculty of Computer Science, Sriwijaya University, Indonesia. He can be contacted at email: huda@unsri.ac.id.

Kemahyanto Exaudi, is a master's holder in electrical engineering from Sepuluh Nopember Institute of Technology (ITS) Surabaya. He works as a lecturer, instructor, and researcher at Sriwijaya University Indonesia. His broad research interests cover topics relating to computer smart farming, the Internet of Things, embedded systems, and pervasive computing. He can be contacted at kemahyanto@ilkom.unsri.ac.id.

Ades Harafi Duri, domiciled in the city of Muara Enim, born on December 27 2000. Graduated from the computer engineering, Faculty of Computer Science, Sriwijaya University, Indonesia in 2023. Graduated with a Thesis that focused on Facial Recogntion in Robots. Interest about Informaton Technology especially Robotics, Embedded Systems, and Internet Of Things. Get in touch at email: adesharafiduri27@gmail.com