

Robot movement controller based on dynamic facial pattern recognition

Siti Nurmaini¹, Ahmad Zarkasi², Deris Stiawan³, Bhakti Yudho Suprpto⁴, Sri Desy Siswanti⁵,
Huda Ubaya⁶

^{1,2,3,5,6}Faculty of Computer Science, Universitas Sriwijaya, Indonesia

⁴Faculty of Engineering, Universitas Sriwijaya, Indonesia

Article Info

Article history:

Received Mar 20, 2020

Revised Dec 3, 2020

Accepted Jan 11, 2021

Keywords:

Face detection

Mobile robots

Navigation technique

Ram based neural network

Ram discriminator

ABSTRACT

In terms of movement, mobile robots are equipped with various navigation techniques. One of the navigation techniques used is facial pattern recognition. But Mobile robot hardware usually uses embedded platforms which have limited resources. In this study, a new navigation technique is proposed by combining a face detection system with a ram-based artificial neural network. This technique will divide the face detection area into five frame areas, namely top, bottom, right, left, and neutral. In this technique, the face detection area is divided into five frame areas, namely top, bottom, right, left, and neutral. The value of each detection area will be grouped into the ram discriminator. Then a training and testing process will be carried out to determine which detection value is closest to the true value, which value will be compared with the output value in the output pattern so that the winning discriminator is obtained which is used as the navigation value. In testing 63 face samples for the upper and lower frame areas, resulting in an accuracy rate of 95%, then for the right and left frame areas, the resulting accuracy rate is 93%. In the process of testing the ram-based neural network algorithm pattern, the efficiency of memory capacity in ram, the discriminator is 50%, assuming a 16-bit input pattern to 8 bits. While the execution time of the input vector until the winner of the class is under milliseconds (ms).

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Siti Nurmaini

Department of Computer System

Universitas Sriwijaya, Indonesia

Email: sitinurmaini@gmail.com

1. INTRODUCTION

The robot can work in a variety of environmental conditions, both in a conditioned environment and in dangerous environments. With this reliability, robots are often used in helping various human activities on mobile [1]-[3]. Thus, the development of mobile robots has experienced a very large increase. In terms of movement, mobile robots are equipped with various navigation techniques [4]. This navigation technique helps the robot is moving and recognizing the environment well. However, conventional navigation techniques always optimize robot movements based on time variables, whereas robots that blend socially with humans do not need such a thing [5]. Mobile robots move to adjust to the human social environment, for example, avoiding collisions in crowded situations, helping in carrying goods, delivering orders, and communicating with humans by recognizing facial features. For personal interaction with the robot, accuracy is needed on facial features that can be recognized by the robot.

Object detection is one of the familiar studies in the field of image processing and pattern recognition. Object detection is intended so that the computer can detect and sort every object contained in an image, and the object can be inanimate objects or living things, such as vehicles, balls, plants, animals, or humans [6]-[7]. This science has been widely used in people's lives, such as surveillance cameras, robot vision, robot assistants, video conferences, missing person searches, and interactions between humans and machines [8]-[10]. In this study [6], [11]-[12], the object used by the system is a face object. This system is made so that the machine can recognize human faces that are being invited to interact. But to be able to be applied to the navigation system, the robot must first recognize the desired user's face [13]-[14].

The face detection system can determine which is a face and which is not good in real-time or not by distinguishing between facial features from all patterns recorded by the camera [15]-[18]. So that the machine can detect faces properly, it must be approached to certain factors. The approach is taken on this face detection, such as lighting, position views, colours, shadows, noise, or image resolution [19]-[20]. Face tracking is used to follow faces from time to time in real-time by determining face changes (scale and position of faces) to allocate faces in a frame [15]-[16], [21]. This approach will easily be implemented on a computer system. However, for systems that have an embedded platform [22]-[25], there will be many problems to implement, plus the face pattern must be used as a mobile robot motion navigation pattern. A reliable method that can be used in systems with embedded platforms is a ram base neural network [1]-[4].

The ram base neural network method uses ram nodes to store and process data. Like conventional neural networks, ram base neural networks also have an input layer, hidden layer, and output layer. The number of input vectors will determine the number of ram nodes used, while the number of input patterns will determine the number of ram discriminators used. All these processes are done in binary data on ram nodes. The ram nodes are then grouped into several classifications of ram, also known as ram discriminators. It is in this ram discriminator that the input patterns will be trained and tested. [1]-[3]. Because the process is done directly in ram memory, the execution time is fast, so the mobile robot response is also fast

In this research proposed a new navigation technique by combining face detection systems with a ram based neural network. Facial patterns were detected and processed using the Viola-Jones method [13]-[14]. This algorithm can distinguish faces from the background in the camera frame. then facial feature extraction using Eigenface algorithm to recognize detected faces [19], [26]-[27]. The patterns are detected in the five detection areas until the face width scale on the camera reaches a predetermined limit. The value of the area boundary will be the value of the ram discriminator, which will first be processed in several stages. When the facial identity has been detected and recognized by the system on a certain scale, the robot will respond by moving according to the face position.

2. RESEARCH METHODOLOGY

In this section, the methodological stages consist of hardware development, software development, and software and hardware integration.

Step 1, Hardware development aims to explore and analyze various aspects of its use. Hardware must support every computation step in the embedded system because hardware systems have limited resources. Therefore, designing a system architecture that includes a ram based neural network architecture is of particular concern in this study. ARM processors are used for computing face detection algorithms, from data entry to final results. Then the AVR328 microcontroller will be used as a ram based neural network computing and mobile robot navigation system.

Step 2, Software development will greatly influence the results to be the software will run a facial pattern recognition algorithm and ram based neural network algorithm that is implemented on mobile robot navigation.

Step 3, Testing and validation aim to make all the main components in this research work properly and correctly. Software testing is done to test the algorithm a computation that has been developed in real terms, from input to simulation results obtained. Then hardware testing is carried out in various system blocks, where each block will certainly work and function properly. After that, the whole system is unified, and the results will be analyzed in detail and in detail.

2.1. Hardware architecture

The hardware architecture is the design of hardware system development. In hardware design, there are components that are interconnected with one another. This gives a general description of how the hardware system works. The hardware design in this study consists of two main components, namely the ARM processor module and the 32 microcontroller module. The ARM module processes image data from the camera to get certain face patterns. The data will go through several stages of the process. The process stages include facial image processing, feature extraction, and output, while the output in this research includes data

from the classification process to achieve the targets set. The target is a face pattern that has been processed before. The microcontroller module 32 will process ram based neural network algorithm and the proximity sensor data. Face classification data will be compared with the input classification pattern on a ram based neural network. The data will be trained and tested to find which classification pattern is closer to the face position on the frame. The proximity sensor functions as a deterrent to obstacle avoidance, while face detection is to search for the desired user's face. The combination of these two systems is then used as a navigation system and target search. The target is the user's face pattern that has been processed before. Figure 1 is the robot vision hardware architecture.

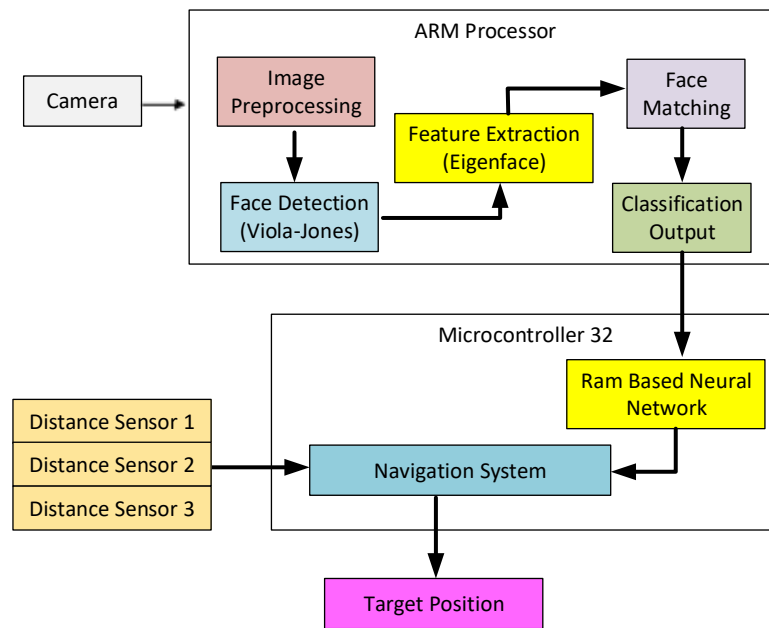


Figure 1. Robot vision hardware architecture

The camera input stage is preprocessing, which consists of two processes, are taking RGB images and then converting RGB images to grayscale images. At the face detection stage consists of four processes, are haar features, integral image, cascade classifier, and Adaboost classifier. This process is also called the Viola-Jones method. The next step is the feature extraction process. This section consists of four processes are find the distance feature, find the Eigenvalue, matching data, and face recognized. After the image is recognized, the image will be grouped into several classifications of frames. Then the classification data will be sent to the microcontroller to be processed with a ram based neural network algorithm. Three proximity sensors function to protect the robot from collisions when approaching the user's facial object.

2.2. Face detection algorithm

Software algorithm design is a general description of the process that will be run by the software. In the process, the algorithm is divided into several steps, namely input, feature extraction, learning process, and output. The following are the steps in the face data process,

(i) Image data in a set S.

$$S = \{ T_1, T_2, \dots, T_M \} \tag{1}$$

(ii) Find the average value of training data (Ψ)

$$\Psi = \frac{1}{m} \sum_{n=1}^m T_n \tag{2}$$

(iii) Find the difference (ϕ) between the training image (T_i) and the mean value (Ψ)

$$\phi_i = T_i - \Psi \tag{3}$$

(iv) Calculate the value of the Covariance matrix (c)

$$C = \frac{1}{M} \sum_{n=1}^m \phi_n \phi_n^t \quad (4)$$

(v) Calculate eigenvalue (λ) and eigenvector (v) of the covariance matrix (C)

$$C \times v_i = \lambda_i \times v_i \quad (5)$$

(vi) Calculate eigenface can be searched with

$$\mu_i = \sum_{k=1}^m v_{ik} \phi_k \quad (6)$$

(vii) Calculate the Euclidean distance to find the smallest distance between the *eigenface* value of the training image in the database and the *eigenface* of the test face image.

$$\epsilon_k = (\mu_{\text{new}} - \mu_i)^2 \quad (7)$$

Figure 2 is a general system algorithm design. The input part (i) is a face image obtained from the camera. The image will be processed using the Haar Cascade method as face detection to get the face area. Feature extraction (ii) will process facial data to be more specific again. This aims to get the face position. Data from feature extraction is binary data. Output (iii) determines what the hardware system will do if the accuracy of face data has been obtained.

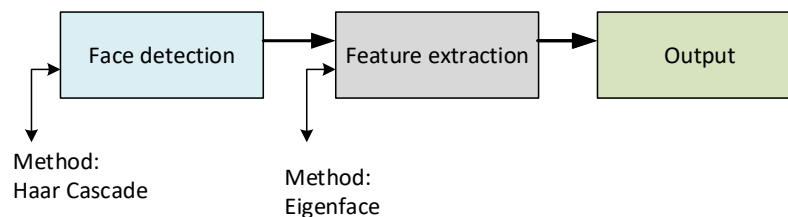


Figure 2. Face algorithm design

Each face stored in the training data is transformed into facial space, and its components are stored in memory as a reconstruction. Face space must be filled with known faces. When the face that we recognize is entered into the system, then the input face is projected into the facial space. The system will then calculate the distance from all saved faces. Because the face is also represented by facial space, the reconstruction must be similar to the original, and then the reconstruction error will be small. Non-face images will have large reconstruction errors that are greater than a certain threshold. The basic principle of Eigenface is to project an image into the eigenface field by finding the eigenvector that each image has and projecting it into the Eigenface obtained. The amount of Eigenface depends on the amount of training image held.

2.3. Ram based neural network architecture

The ram based neural network model is based on the weighted-sum-and-threshold neural network model, which has been pioneered by the Threshold Logic Unit, by McCulloch and Pittsram based neural network is developed based on a neural network model that has no weight on each branch of the network. The input and output networks are binary numbers. Its neural network functions are stored in a ram lookup table. The learning process in this network is a change from the contents of the data in the ram lookup table, the results are very flexible, and the training process is fast. Each RAM node is taught to respond to a value of 1 for each pattern in the training process [1]-[2]. An n input node has 2^n memory locations. Addressed by an n-bit input vector = {bit 0, bit 2,... .., bit 15}, binary signal $x = \{x_0, x_1, \dots, x_{15}\}$. Ram node scheme can be seen in Figure 3.

In Figure 3, the five face position classifications have been compared with the input pattern, forming five pattern classifications to the input vector. This pattern is then converted into binary bits. The output of the ram node will be the input for the discriminated ram. A ram discriminator consists of a series of x bit ram words with n input vectors and a summing device (Σ). Each ram discriminator can accept binary pattern ($x \cdot n$)

bits as input. A discriminator must condition all ram memory locations to logic 0 and select a training pattern formed by the binary pattern (x-n) bits. Ram discriminator consists of five classifications, namely the neutral discriminator, upper discriminator, lower discriminator, right and left discriminator. The winner class produced the output of the ram discriminator process.

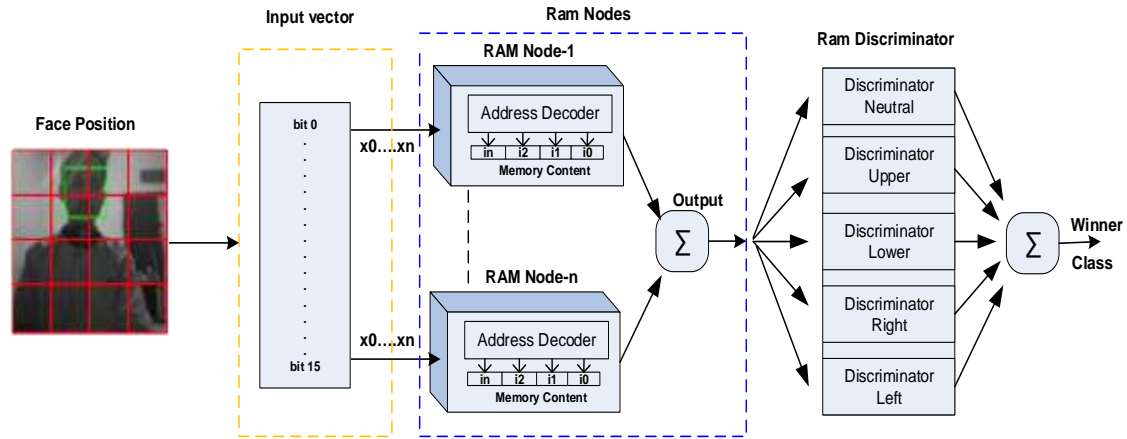


Figure 3. Ram based neural network architecture design

3. EXPERIMENTAL RESULTS

This section will conduct testing, collection, processing and validation of data. Testing is carried out on both hardware and software. Hardware testing, with a focus on motion response, which will determine the robot's real-time movement. Software testing focuses on proving the computational process and learning algorithms, both on face detection and on ram-based neural network algorithms. The learning and testing process will also be explained in this section. The face position data will determine the robot's movement.

3.1. Face position for robots movement

The method for recognizing robot face patterns will move to do commands based on the user face position. The technique used to recognize faces by robots is to divide the camera area into sections. Each part has a specific function. In this stud, the camera frame used was f (240x240) pixels. Then, the frame is divided into several ROI, which have many different pixels. The frame area consists of upper, lower, right, left, and neutral. The upper frame area has 120x120 pixels, the lower is 120x120, the right is 60x120, the left is 60x120, while the neutral position is located in each corner of the frame with the pixel size of each corner 60x60. The rest of the pixels are nutritional areas. Figure 4 is a division of the face detection area for robot motion.

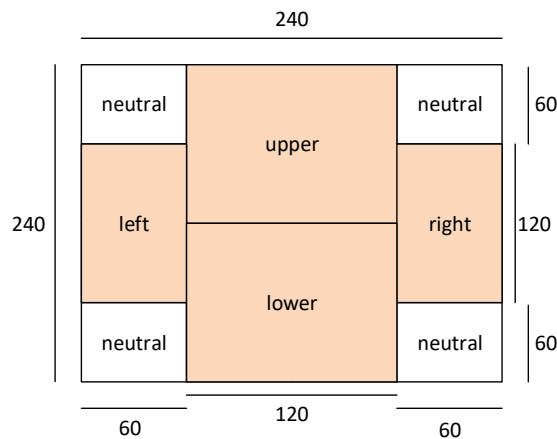


Figure 4. Face position area design

In the picture above, there are four neutral positions located at each corner of the camera. If the face is exactly in this position, the robot will stop moving. After that, the robot starts searching for the object of the user's face by moving the camera to various positions using a servo motor. The position of the face object in the training process has 21 datasheets for one user, so the total datasheet for 3 users is 63. The datasheet for the first user can be seen in Figure 5.



Figure 5. Face datasheet for trainin

Table 1 is a table of pixel boundary values for robot movement. If the face is in one of the positions of the pixel area, the action to be performed by the robot is to move according to the rules of the robot's movement. For each position of the frame area, the robot will move according to the function of the frame area. The following are the rules of robot motion based on the table.

- a) In the upper area, the robot moves forward. The pixel boundary values are 1600 up to 14400 pixels.
- b) In the lower area, the robot moves to stop. The pixel boundary values are 1600 up to 14400 pixels.
- c) In the right area, the robot moves turn right. The pixel boundary values are 1600 up to 7200 pixels.
- d) In the left area, the robot moves turn left. The pixel boundary values are 1600 up to 7200 pixels.
- e) In the neutral area, the robot does searching face. The pixel boundary values are 1600 up to 3600 pixels.

Table 1. The pixel boundary values for robot movements

Face detection	Face position	Robot action	Accuracy detection
	 upper	Forward min = 1600 pixels max= 14400 pixels	The experiment was carried out with 93 face samples, with an accuracy rate of 95%
	 lower	Stop min = 1600 pixels max=14400 pixels	
	 right	Turn right min = 1600 pixels max =7200 pixels	The experiment was carried out by taking face samples of 93. At the lowest pixel value, the accuracy rate is 75%, while the pixel value is above 2000, the acquisition rate is 93%
	 left	Turn left min = 1600 pixels max= 7200 pixels	

3.2. Facial recognition testing process

The preprocessing process will search for images that contain faces. One of the methods used in detecting face objects is the Viola-Jones method [28]-[30]. After that, the face identification process is done by the eigenface method. A new image will be tried to be recognized by finding the difference between the new image with the middle value of the previous data. If a value is found below zero, then the value is replaced with a zero value. This method produces binary data.

Viola-Jones data retrieval uses a number of face data taken continuously. While the introduction to Eigenface used facial image samples obtained from 3 respondents who had been saved as recognizable faces, consisting of pictures (a), (b) and (c). Figure 6 is a sample of face testing. While Figure 7 shows the matrix value of each of these images.

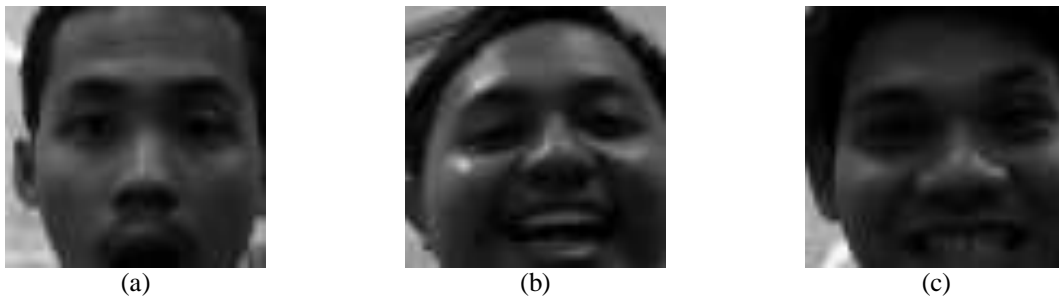


Figure 6. Face testing, (a) user_1, (b) user_2. (c) user_3

$$\begin{matrix}
 a = \begin{bmatrix} 77 \\ 55 \\ 41 \\ 82 \\ 40 \\ 46 \\ 122 \\ 36 \\ 58 \end{bmatrix} &
 b = \begin{bmatrix} 77 \\ 55 \\ 69 \\ 76 \\ 52 \\ 56 \\ 63 \\ 29 \\ 43 \end{bmatrix} &
 c = \begin{bmatrix} 26 \\ 46 \\ 25 \\ 31 \\ 53 \\ 60 \\ 75 \\ 33 \\ 43 \end{bmatrix}
 \end{matrix}$$

Figure 7. Matrix value of face testing

The next step is to calculate each variable from (1) until (7). The results of the calculations can be seen in Table 2, which is the result of testing (a) and Table 3, which is the result of testing (b). Table 2 displays data from calculations (Ψ), ϕ_i , covariant (c), and eigenvalue (λ). Then in Table 3 displays the results of the calculation of eigenvector (v) and Eigenface (μ_i). The next step is to test the data that has been obtained with new data contained in Figure 8, for which the image matrix has been obtained. The purpose of this test is to obtain the closeness of the value of the new image with the image that has been previously saved by finding the Euclidean value.

Table 2. Result of testing (a)

Ψ	$\phi_i (\Phi_1, \Phi_2, \Phi_3)$	covariant (c),	eigenvalue (λ_i)
$ \begin{bmatrix} 60 \\ 58.66 \\ 45 \\ 63 \\ 48.33 \\ 54 \\ 86.66 \\ 32.66 \\ 48 \end{bmatrix} $	$ \Phi_1 = \Phi_2 = \Phi_3 = \begin{bmatrix} 17 & 17 \\ -3.66 & 16.33 \\ -4 & 24 \\ 19 & 13 \\ -8.33 & 3.66 \\ -8 & 2 \\ 35.33 & -23.66 \\ 3.33 & -3.66 \\ 10 & -5 \end{bmatrix} $	$ \begin{bmatrix} 2172.44 & -564.88 & -1607.55 \\ -564.88 & 1916.77 & -1351.88 \\ -1607.55 & -1351.88 & 2959.44 \end{bmatrix} $	$ 0 = -\lambda^3 + 7048.65\lambda^2 - 11534953.5099\lambda + 115348.830236435 $ <p>so :</p> $ \lambda_1 \approx 4.46556 $ $ \lambda_2 \approx 2.58307 $ $ \lambda_3 \approx 0 $

Table 3. Result of testing (b)

eigenvector (v)			eigenface (μ_i)		
-0.4891	0.6537	-0,5773	-3.26862	-3.3227318	-1.07288
-0.3216	-0.7504	-0,5773	-2.149201	3.8142776	-3.099415
0.8107	-0.09671	-0,5773	5.417824	-4.915466	-2.9206276

After the training data value (Ψ) is obtained, it is continued by calculating the eigenface value for the new image. Figure 8 is an image that will be tested to get the value of closeness to the training image. The following equation is:

$$\mu_{\text{new}} = v(T_{\text{new}} - \Psi) \quad (8)$$

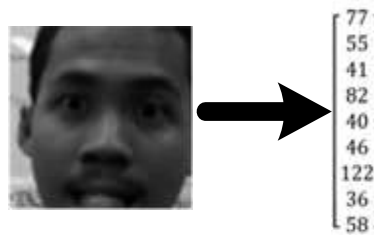


Figure 8. New image testing

In order to obtain the eigenface value for $\mu_{\text{tester}} = [-3.2686 \ -3.32318 \ -2.38418]$. Next, look for the Euclidean distance value to find the smallest distance between the Eigenface value of the training image in the database and the Eigenface of the face testing image. The formula used is as follows (7). Then the smallest Euclidean distance (ϵ) value compared for each database image i

calculation results for face testing (a) is:

$$\epsilon_{k1} = ([-3.2686 \ -3.32318 \ -2.38418] - [-3.26862 \ -3.322731 \ -1.07288])^2$$

$$\epsilon_{k1} = [0.0000000e+00 \ 1.4551915e-11 \ 6.9633188e-13]$$

$$\epsilon_{k1} = 0.0000000e+00$$

calculation results for face testing (b) is

$$\epsilon_{k2} = ([-3.2686 \ -3.32318 \ -2.38418] - [-2.14920 \ 3.81427 \ -3.09941])^2$$

$$\epsilon_{k2} = [1.2531072e+02 \ 5.0936895e+03 \ 8.1854523e-12]$$

$$\epsilon_{k2} = 8.1854523e-12$$

calculation results for face testing (c) is

$$\epsilon_{k3} = ([-3.2686 \ -3.32318 \ -2.38418] - [5.41782 \ -4.91546 \ -2.92062])^2$$

$$\epsilon_{k3} = [7.5454390e+03 \ 8.0156073e+02 \ 7.1942452e-12]$$

$$\epsilon_{k3} = 7.194245e-12$$

From the results of the above calculation, it is known that the smallest distance value that indicates the proximity of the testing image is on face testing (a). The distance value obtained is the minimum distance $\epsilon_{k1} = 0.0000000e + 00$. In the learning process, the Euclidean value that can detect it well is 3,7638 to 18,7182, with an accuracy rate of 90%. The representation graph of Euclidian data values for user one can be seen in Figure 9.

The next step is to find out the level of accuracy in the training process by using a confusion matrix. The test was carried out as many as 93 face samples, with six respondents. The six respondents were 63 samples of 3 user faces as shown in Figure 5 30 samples of faces from 3 unknown people. The results obtained can be seen in Table 4, which is a confusion matrix eigenface. In the table above, the true positive is 61, the true negative is 23, the false positive is 5, and the false negative is 4. So the result of measuring the level of accuracy in the eigenface testing process is 90.32%.



Figure 9. Graph of euclidian data values user 1

Table 4. Confusion matrix eigenface

Testing Results	Data Type	
	Known Data	Unknown Data
Match	61 True Positive	4 False Negative
Not Match	5 False Positive	23 True Negative

3.3. Ram based neural network testing

The input vector data is obtained from five face position data in pixels, namely neutral, upper, lower, right and left. The data is converted first into binary data. After that will be obtained 16 bits of binary data, assuming one binary data bit represents 1-pixel data. Because there are five face position data, then at least memory must provide five addresses for ram nodes, with a data width of 16 bits as well so that the incoming data through the input vector is then sent to the ram node. The requirement for data to be entered via the input vector is if the data is entered in the activation data of the ram node. For details, see Table 5, which is the data on the activation function of the ram node.

Table 5. Ram node activation data

Face Position	The pixel boundary data	Binary data for the input vector	The activation function of the ram node
Upper	1600 to 14400	0000 0110 0100 0000 to 0011 1000 0100 0000	ram node Upper
Lower	1600 to 14400	0000 0110 0100 0000 to 0011 1000 0100 0000	ram node Lower
Right	1600 to 7200	0000 0110 0100 0000 to 0001 1100 0010 0000	ram node Right
Left	1600 to 7200	0000 0110 0100 0000 to 0001 1100 0010 0000	ram node Left
Neutral	1600 to 3600	0000 0110 0100 0000 to 0000 1110 0001 0000	ram node Neutral

In Table 6, ram nodes are classified into five groups of ram, namely upper ram node, lower ram node, right ram node, left ram node, and neutral ram node. Each ram node has 16 bits of data. The data consists of 16 bits of upper limit data and 16 bits of lower limit data. The 16-bit ram data nodes are further divided into 8 bit MSB and 8 bit LSB. Then, each ram node data will be the input on the ram discriminator. In this section, the RAM discriminator consists of five groups, namely the discriminator upper, discriminator lower, discriminator right, discriminator left and discriminator neutral. Each ram discriminator is designed to only store 8 bits of data. Therefore, before the ram node data entered the ram discriminator, the ram node data must first be optimized. This means that the data values have a high priority. When viewed from the binary data priority value, the data in the MSB group has a higher priority value. So that 8 bits of MSB data, both at the upper limit (Max) and the lower limit (Min), become input data on the RAM discriminator, during 8 bit LSB will be the hidden layer. Because of the data optimization process, the ram discriminator data has 8 bits (255) invisible patterns. So that the training and testing process becomes faster, and memory capacity becomes 50% more efficient.

Figure 10 is a graph of the learning and training process on a ram based neural network. The graph consists of processing time (T) in ms and the number of patterns. Time is set in the range 0.02 ms to 0.18 ms, while the number of sample patterns is 200 patterns. The results obtained in testing show that the highest time in the learning and training process is 0.18 ms, and the lowest time is 0.03 ms. Meanwhile, the average execution time lies between 0.03 ms to 0.08 ms. With these results, the learning and training process on the implementation of the ram based neural network algorithm is in accordance with the implementation of the embedded platform.

Table 6. Ram discriminator optimization data

Ram node	Ram node binary data				Ram discriminator	Ram discriminator binary data	
	16 bit Max		16 bit Min			8 bits Max	8 bits Min
	8 bits MSB	8 bits LSB	8 bits MSB	8 bits LSB			
ram node Upper	0011 1000	0100 0000	0000 0110	0100 0000	discriminator Upper	0011 1000	0000 0110
ram node Lower	0011 1000	0100 0000	0000 0110	0100 0000	discriminator Lower	0011 1000	0110 0100
ram node Right	0001 1100	0010 0000	0100 0000	0100 0000	discriminator Right	0001 1100	0000 0100
ram node Left	0001 1100	0010 0000	0100 0000	0100 0000	discriminator Left	0001 1100	0000 0100
ram node Neutral	0000 1110	0001 0000	0000 0110	0100 0000	discriminator Neutral	0000 1110	0000 0110



Figure 10. Time response of the learning and training process

4. CONCLUSION

In this research, testing and analysis had been carried out on a dynamically moving system. The combination of face detection methods and ram based neural network algorithms have worked according to the given parameters, thus providing good response to the movement of the mobile robot. The first test was carried out by taking 63 face samples for the upper and lower area frames, resulting in an accuracy rate of 95%. Then proceed with testing for the Right and Left frame areas, the resulting accuracy rate is 93%. Furthermore, the process of testing the ram based neural network algorithm pattern is carried out. The efficiency of memory capacity on the Ram Discriminator is as much as 50%, assuming the 16-bit input pattern becomes 8 bits. While the execution time of the input vector until the winner class is millisecond (ms).

ACKNOWLEDGEMENTS

This research was funded by the Ministry of Research and Technology Republic of Indonesia for funding the research on "Penelitian Disertasi Doktor 2020" and Universitas Sriwijaya. The 1st author would like to thank the Faculty of Computer Science, The Faculty of Engineering, Intelligence System Research Group, and COMNETS Research Group and also Embedded Systems, Control System and Robotic Laboratory Universitas Sriwijaya for supporting this research.

REFERENCES

- [1] S. Nurmaini and A. Zarkasi, "Simple Pyramid RAM-Based Neural Network Architecture for Localization of Swarm Robots," *Artic. J. Inf. Process. Syst.*, vol. 11, no. 3, pp. 370-388 2015, doi: 10.3745/JIPS.01.0008.
- [2] A. Zarkasi, A. I. Wuryandari, and R. Fauzian, "Multilayer processing architecture of RAM based neural network with memory optimization for navigation system," in *Proceedings of the 2013 Joint International Conference on Rural Information and Communication Technology and Electric-Vehicle Technology, rICT and ICEV-T 2013*, 2013, doi: 10.1109/rICT-ICEVT.2013.6741494.
- [3] B. McElroy *et al.*, "Highly efficient Localisation utilising Weightless neural systems," 2012.
- [4] S. Nurmaini and B. Tutuko, "A New Classification Technique in Mobile Robot Navigation," *TELKOMNIKA*

- (*Telecommunication, Computing, Electronics and Control*), vol. 9, no. 3, pp. 453-464, 2011.
- [5] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Rob. Res.*, vol. 35, no. 11, pp. 1289-1307, Sep. 2016, doi: 10.1177/0278364915619772.
- [6] V. Bonato, A. K. Sanches, M. M. Fernandes, J. M. P. Cardoso, E. D. V Simoes, and E. Marques, "A Real Time Gesture Recognition System for Mobile Robots," *ICINCO 2004, Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal, August 25-28, 2004.
- [7] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks." *Conference: Advances in Neural Information Processing Systems*, 2016.
- [8] "Modular Weightless Neural Network Architecture for Intelligent Navigation-Sriwijaya University Repository." [Online]. Available: <http://repository.unsri.ac.id/8839/>. [Accessed: 05-Mar-2020].
- [9] J. Li and F. Yang, "Research on multi-robot scheduling algorithms based on machine vision," *Eurasip J. Image Video Process.*, vol. 2018, no. 1, p. 137, Dec. 2018, doi: 10.1186/s13640-018-0355-x.
- [10] K. Solovey and D. Halperin, "K-color multi-robot motion planning," *Int. J. Rob. Res.*, vol. 33, no. 1, pp. 82-97, Jan. 2014, doi: 10.1177/0278364913506268.
- [11] Y. Chen, L. Song, Y. Hu, and R. He, "Adversarial occlusion-aware face detection," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems, BTAS 2018*, 2018, doi: 10.1109/BTAS.2018.8698572.
- [12] A. Barbu, N. Lay, and G. Gramajo, "Face Detection with a 3D Model," *Academic Press Library in Signal Processing Image and Video Processing and Analysis and Computer Vision*, vol. 6, pp 237-259, 2018.
- [13] J. Foytik, P. Sankaran, and V. Asari, "Tracking and recognizing multiple faces using Kalman filter and ModularPCA," in *Procedia Computer Science*, vol. 6, pp. 256-261, 2011, doi: 10.1016/j.procs.2011.08.047.
- [14] K. H. Bin Ghazali, J. Ma, and R. Xiao, "Driver's Face Tracking Based on Improved CAMShift," *Int. J. Image, Graph. Signal Process.*, vol. 5, no. 1, pp. 1-7, Jan. 2013, doi: 10.5815/ijjgsp.2013.01.01.
- [15] S. Zhang, L. Wen, H. Shi, Z. Lei, S. Lyu, and S. Z. Li, "Single-Shot Scale-Aware Network for Real-Time Face Detection," *Int. J. Comput. Vis.*, vol. 127, no. 6-7, pp. 537-559, Jun. 2019, doi: 10.1007/s11263-019-01159-3.
- [16] M. Berger, A. F. De Souza, J. de O. Neto, E. de Aguiar, and T. Oliveira-Santos, "Visual tracking with VG-RAM Weightless Neural Networks," *Neurocomputing*, vol. 183, pp. 90-105, Mar. 2016, doi: 10.1016/j.neucom.2015.04.127.
- [17] M. De Gregorio, M. Giordano, S. Rossi, and M. Staffa, "Experimenting WNN support in object tracking systems," *Neurocomputing*, vol. 183, pp. 79-89, Mar. 2016, doi: 10.1016/j.neucom.2015.09.117.
- [18] Z. Ghouse, N. Hiwrale, and N. Ranjan, "Military Robot for Reconnaissance and Surveillance using Image Processing," *Int. Res. J. Eng. Technol.*, 2017.
- [19] "A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection-IEEE Conference Publication." [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7013097>. [Accessed: 05-Mar-2020].
- [20] A. Zarkasi *et al.*, "Face Movement Detection Using Template Matching," in *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018*, pp. 333-338, 2019, doi: 10.1109/ICECOS.2018.8605215.
- [21] V. E. Dahiphale and S. R., "Real-Time Computer Vision System for Continuous Face Detection and Tracking," *Int. J. Comput. Appl.*, vol. 122, no. 18, pp. 1-5, Jul. 2015, doi: 10.5120/21797-5100.
- [22] A. Soetedjo and I. K. Somawirata, "Implementation of face detection and tracking on a low cost embedded system using fusion technique," in *2016 11th International Conference on Computer Science & Education (ICCSE)*, pp. 209-213, 2016, doi: 10.1109/ICCSE.2016.7581582.
- [23] C. Qin, W. Zhang, F. Cao, X. Zhang, and C.-C. Chang, "Accepted Manuscript Separable Reversible Data Hiding in Encrypted Images via Adaptive Embedding Strategy with Block Selection," *Signal Processing*, 2018, doi: 10.1016/j.sigpro.2018.07.008.
- [24] A. Kaczmarczyk and W. Zatorska, "Accelerating Image Fusion Algorithms Using CUDA on Embedded Industrial Platforms Dedicated to UAV and UGV," in *Advances in Intelligent Systems and Computing*, vol. 920, pp. 697-706, 2020, doi: 10.1007/978-3-030-13273-6_65.
- [25] Z. Shi, H. Yu, L. K.-2018 I. I. C. on, and undefined 2018, "Image Stitching Algorithm Based on Embedded System," *ieeexplore.ieee.org*, 2018.
- [26] C. Yu, C. Xie, and Y. Zhang, "A facial expression recognition strategy based on template matching," in *Proceedings - 2015 8th International Congress on Image and Signal Processing, CISP 2015*, pp. 835-840, 2016, doi: 10.1109/CISP.2015.7407993.
- [27] M. A. Berbar, H. M. Kelash, and A. A. Kandeel, "Faces and facial features detection in color images," in *Geometric Modeling and Imaging New Trends, 2006*, vol. 2006, pp. 209-214, 2006, doi: 10.1109/GMAI.2006.18.
- [28] V. Kirby Pascual German and B. Dioquino Gerardo, "Implementing Enhanced AdaBoost Algorithm for Sales Classification and Prediction eVoting System View project," *researchgate.net*, 2017, doi: 10.18178/ijtf.2017.8.6.577.
- [29] S. C. Brubaker, J. Wu, J. Sun, M. D. Mullin, and J. M. Rehg, "On the Design of Cascades of Boosted Ensembles for Face Detection," *International Journal of Computer Vision*, vol. 77, pp. 65-86, 2008, doi.org/10.1007/s11263-007-0060-1.
- [30] J. Huang, Y. Shang, and H. Chen, "Improved Viola-Jones face detection algorithm based on HoloLens," *Eurasip J. Image Video Process.*, vol. 2019, no. 1, p. 41, Dec. 2019, doi: 10.1186/s13640-019-0435-6.