

## Implementation of Weightless Neural Network in Embedded Face Recognition for Eye and Nose Pattern Mobile Identification

Ahmad Zarkasi<sup>1\*</sup>, Kemahyanto Exaudi<sup>2</sup>, Yoppy Sazaki<sup>3</sup>, Londa Arrahmando Romadhona<sup>4</sup>

<sup>1,2,3,4</sup>Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya  
ahmadzarkasi@unsri.ac.id

### ABSTRACT

The pattern of the human face is a form of self-identity and also a form of originality for each individual. The development of facial recognition technology impacts its application in various computing devices, both in computer vision and on single-chip processors. One of the continuously developed implementations is in the form of robot vision by identifying facial features. This research aims to develop a facial recognition system focusing on the identification of the eye and nose areas. This research utilizes the Weightless Neural Network (WNN) method with the Immediate Scan technique. The combination of methods allows for rapid and accurate pattern recognition, even when the face changes position. The detection process is carried out using the Haar Cascade Classifier algorithm, which functions to recognize faces and divides the area into nine different zones to ensure accurate identification. The hardware implementation was carried out on a Raspberry Pi for face detection and facial pattern recognition, as well as the data processor for the robot vision sensor and actuator on the microcontroller. The results of the robot's movement testing have worked well according to the calculation of GPS data values to determine the robot's last position. Then, in the face pattern recognition process, it shows that the proposed method can achieve a maximum accuracy level of up to 98.87% in testing with the internal data set, while testing under different conditions experiences a slight decrease in accuracy to 91.38%. The highest similarity percentage to the faces of other individuals reached 75.69%, indicating that this method is quite adaptive to various facial variations. The execution time of the identification process ranges from 11 ms to 17 ms, depending on the amount of data compared during the scanning. This research is expected to serve as a foundation for further development in robotics systems and embedded system-based facial recognition.

**Keywords:** Face Recognition, *Weightless Neural Network* (WNN), *Haar Cascade Classifier*, face detection, *robot vision*.

### 1. INTRODUCTION

Every person in the world uses their face as a means of self-identification and uniqueness. Faces come in a variety of shapes and configurations that combine to create a pattern that computers can recognize [1][2]. There are numerous ways that facial recognition technology is used in daily life thanks to current technological advancements. This is seen in the use of robotics, smart farming, smart houses, and other technologies [3][4]. The capabilities of robots in the modern era are truly remarkable. Robots autonomously carry out tasks based on human requirements. With the help of several sensors in its operation, and the program that has been embedded

in the robot [5][6]. In its development, it is not uncommon to find mobile robots already equipped with camera sensors. The camera sensor serves as the visual perception for the robot itself. With the presence of the camera sensor, the robot can now also process images obtained through the camera sensor and perform more complex tasks, such as object recognition and facial recognition [7][8]. This robot is known as a vision robot [9][10].

Robot vision implements continuous facial pattern recognition in identifying facial pattern features. Face identification is a form of implementation of detection and recognition in the process of face image processing [11][12][13]. sometimes in its application, robot vision is equipped with various navigation sensors, such as GPS, compass, and distance sensors. These navigation sensors are very important to ensure that the robot can operate safely and not lose direction in determining the destination location [14][15][16]. As in previous research, robots are used to deliver orders both indoors and outdoors. Therefore, navigation sensors are very much needed. For navigation, neural network and fuzzy logic methods are usually used, while for pattern recognition, the Weightless Neural Network Face Recognition (WNN-FR) is employed [17][18][19]. Weightless Neural Network Face Recognition Algorithm is a derivative of the WiSARD Algorithm that focuses solely on facial data processing, where the data is in the form of binary data sets grouped in several N-tuple memories [20][21][19].

The Weightless Neural Network Face Recognition method is reliable in single-chip processor device applications [22][23]. This is because pattern data is processed and recognized at the bit level. However, for the extraction of RGB pattern data in the preprocessor, a computer device is still used, in this case, a mini computer. The mini computer device works like a PC but on an embedded platform [24][25]. The processed data is selected in such a way using feature extraction methods. After that, it is sent to the single-chip processor unit for recognition. This research aims to develop a facial recognition system focusing on the identification of the eye and nose areas, using the Weightless Neural Network (WNN) method with the Immediate Scan technique. The combination of methods allows for rapid and accurate pattern recognition, even when the face undergoes positional changes. The detection process is carried out using the Haar Cascade Classifier algorithm, which functions to recognize faces and divide the area into several different facial areas, to ensure the accuracy of identification. The data processor for the robot vision sensor and actuator on the microcontroller, as well as the Raspberry Pi for face detection and facial pattern recognition, are used in the hardware implementation. The eye and nose image data from a dynamic facial image that has been positioned is the data that will be recognized.

## **2. METHODOLOGY**

This section aims to explain the methodology used and the process of conducting this research. Both from system design, hardware design, software design, and explanation of the methods used. Everything will be discussed in this section.

### **2.1 SYSTEM DESIGN**

This process is the initial stage in this research which will describe the entire system to be created in outline. The description will be loaded into the form of a flow chart which is more like a software design but without additional details, and focuses

on describing how the system will work. The following in Figure 1 will display a diagram that describes the design of the system to be created.

The system will start with the camera of the processor which is a Raspberry pi capturing the image, then the image will be processed from RGB format to Grayscale at the preprocessing stage. After that, detection will be carried out using the Haar cascades classifier method so that an image of the face area is obtained. The face image will be read the size and position so that it can be adjusted to the center of the frame. After that, facial feature extraction using *Resize* image is done in order to get an image of the eye and nose area. After getting an image of the eye and nose area, the image will be converted into binary form and will be sent to Microcontroller which is an Arduino Mega via serial communication.

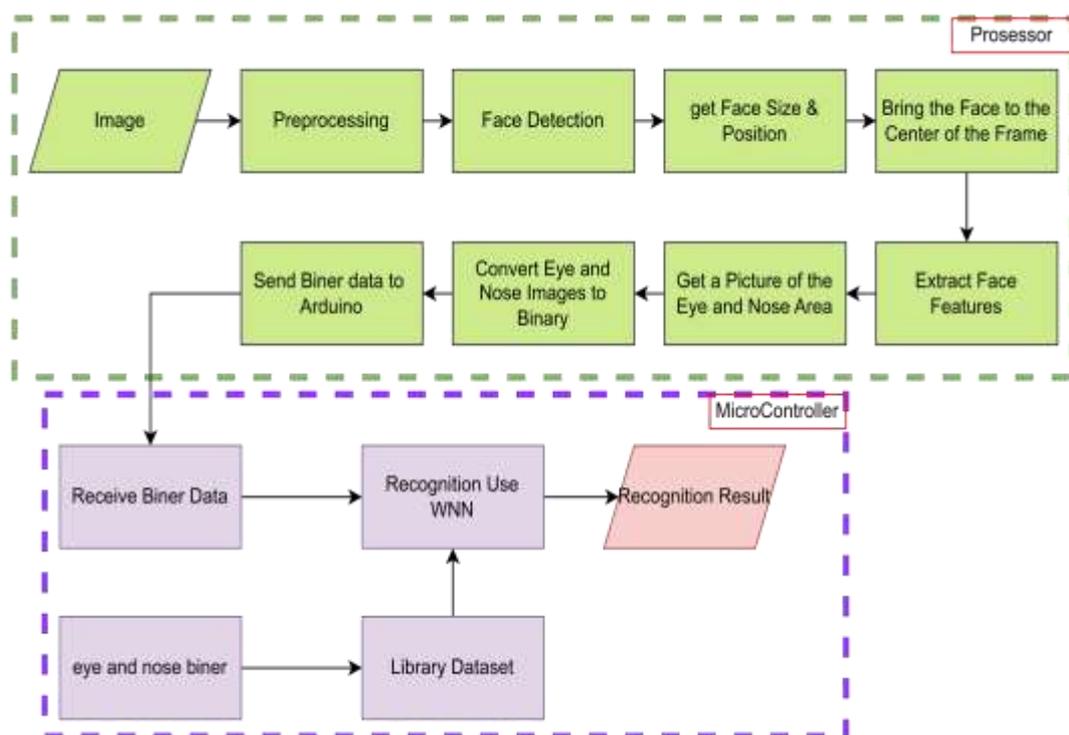


FIGURE 1. System design diagram

The binary data received by Arduino will then be identified. This identification process uses the *Weightless Neural Network (WNN)* method as its identification method. With the data set that has been created, the recognition process will be run and produce a percentage of similarity which, this level of similarity will be used as a reference in determining whether the image is recognized or not.

## 2.2 HARDWARE DESIGN

At this stage, the design of the hardware will be made to make the robot that will be used in this research. Starting with designing the robot frame and after the robot design is complete, it will be printed using a 3D printer. After the frame is printed, the frame will be put together so that the robot frame is finished. After the frame is printed, the frame will be assembled so that the robot frame is complete. Figure 2 shows the design results and the final outcome of the assembled robot.



FIGURE 2. Robot design

After the frame is completed, the robot frame is then assembled according to the required components, as shown in Table 1.

TABLE 1.  
 System requirements analysis

Item Name	Amount
Rspberry Pi 4B	1
Rspberry Pi Camera	1
Arduino Mega	1
Li-po baterai 12 volts	2
BTS-7960 Motor Driver	4
DC motor	4

Once the components have been collected. The components will be assembled so that they can function in accordance with the research objectives. The description of the wiring system in the robot can be seen in the Figure 3.

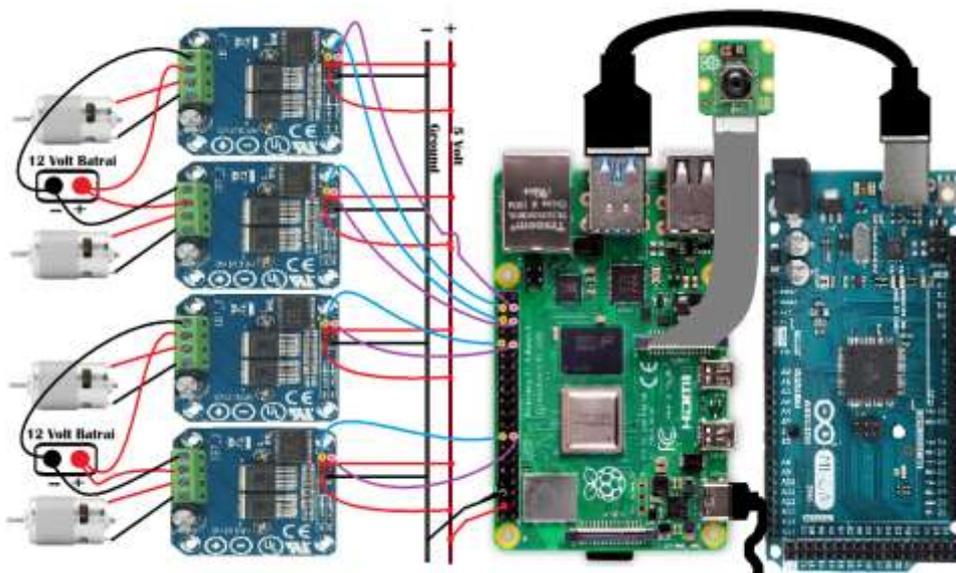


FIGURE 3. System Wiring Diagram

The basic idea behind this robot system is that it will move around freely in the space and use the pi camera as input to determine its direction. if a face is detected by the camera. With the aid of four motor drivers attached to four separate wheels, the robot will move in accordance with the face's location. Through pin input, the Raspberry, an image processing unit, will instruct the motor driver to approach the face up to a predetermined distance. When it's appropriate, the raspberry will snap an image of the face, process it, and send it to the Arduino so that it can be identified.

### 2.3 SOFTWARE DESIGN

The software design consists of several processes involving the face detection process, and the face area identification process. In the development of the process, the algorithm used to perform face detection is the Haar cascade classifier developed by Paul Viola and Michael Jones in 2001. Meanwhile, for the face area identification method using the Weightless Neural Network face Recognition Algorithm (WNN FRA) which is a derivative of the WiSARD Algorithm algorithm. The implementation of the WNN FRA algorithm is assisted by a direct recognition method called Immediate Scan which is based on a comparator that will compare the data in the WNN FRA directly from the point where the data is loaded. Figure 4 is an image of the Immediate Scan method.

In the Figure 4, the reference pattern has data with RAM node resolution  $(x*y)$ , while the input pattern has data with RAM node resolution  $(x*y)'$ . The data checking process is adjusted to the position of each RAM node data from the beginning to the end. The checking starts at the initial row RAM point  $R(1,1)$  to RAM point  $R(1,n)$  in the reference pattern, which is the same as point  $R(1,1)'$  to point  $R(1,n)'$  in the input pattern. Followed by the last row point in the reference pattern RAM  $R(m,1)$  to  $R(m,n)$ , which is the same as point  $R(m,1)'$  to  $R(m,n)'$  in the input pattern.

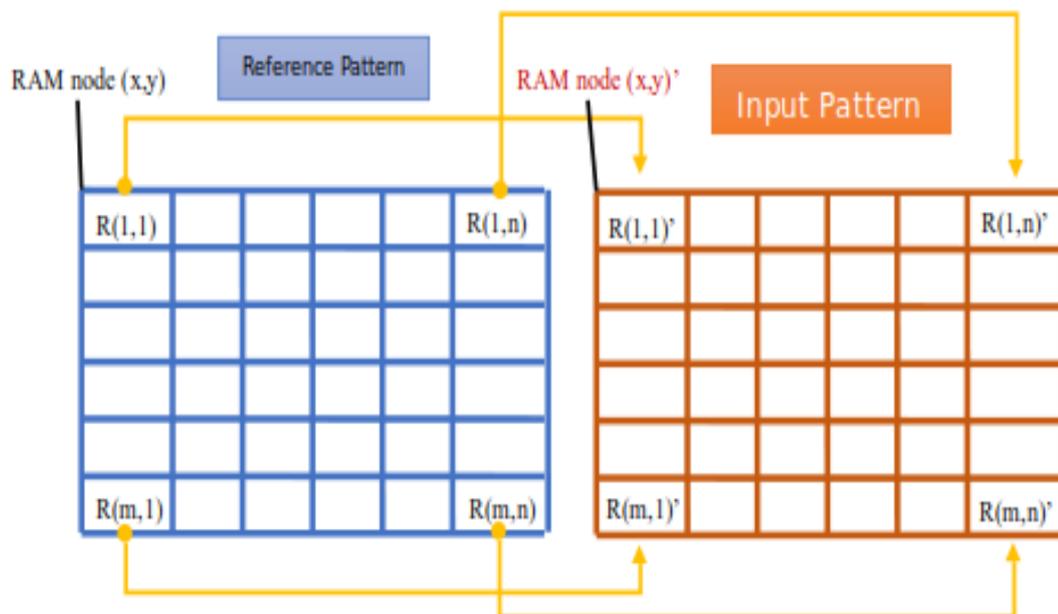


FIGURE 4. Immediate Scan method

**Ahmad Zarkasi, Kemahyanto Exaudi, Yoppy Szaki, Londa Arrahmando Romadhona**  
**Implementation of Weightless Neural Network in Embedded Face Recognition for Eye**  
**and Nose Pattern Mobile Identification**

The next step is the face detection procedure, which involves processing the camera's collected image data to determine the face region. Once this area is determined, the image is shrunk to just include the eyes and nose. In order for the face area to enter the shooting region in the camera frame, the face's coordinates will be read and adjusted in position. The process flow for face detection is shown in Figure 5.

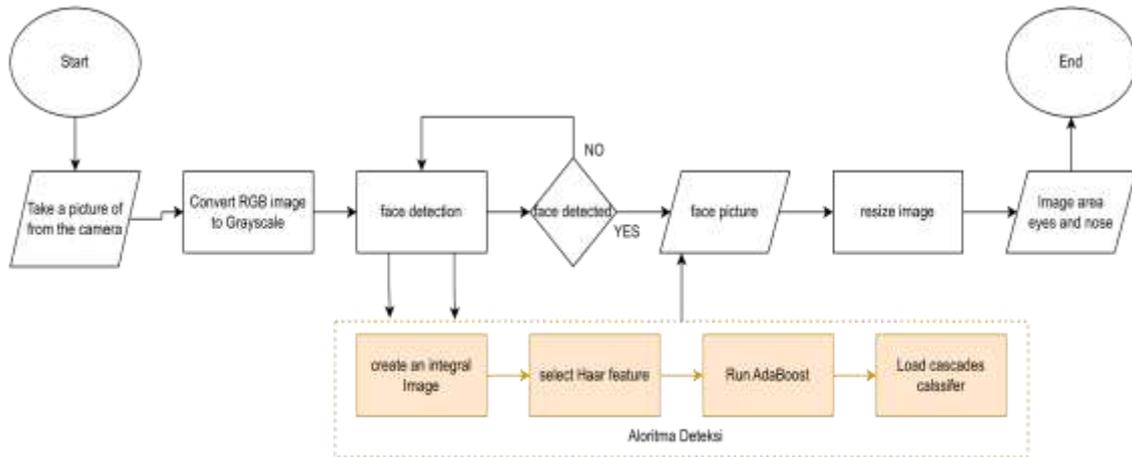


FIGURE 5. Face Detection Flow

Adjusting the picture frame's position to the shooting region comes next once data is obtained from the image frame (Figure 6). The following firing scheme image shows the shooting area. The image capturing area is separated into nine distinct areas according to their respective Regions Of Interest (ROI), as can be seen in the below image. The image will be processed once it is taken in order to be used in the following step, which is the identification process utilizing the WNN FRA method. Analysis of system needs the frame has seven image regions, each with a pixel width of 641 x 481. The facial image will be recorded and processed for detection and recognition if it fills one of the positions

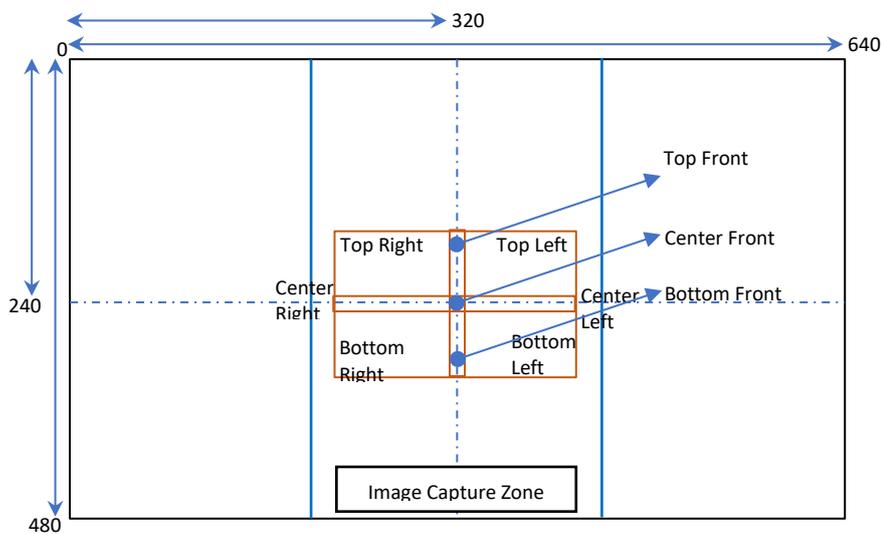


FIGURE 6. Image Capture Scheme

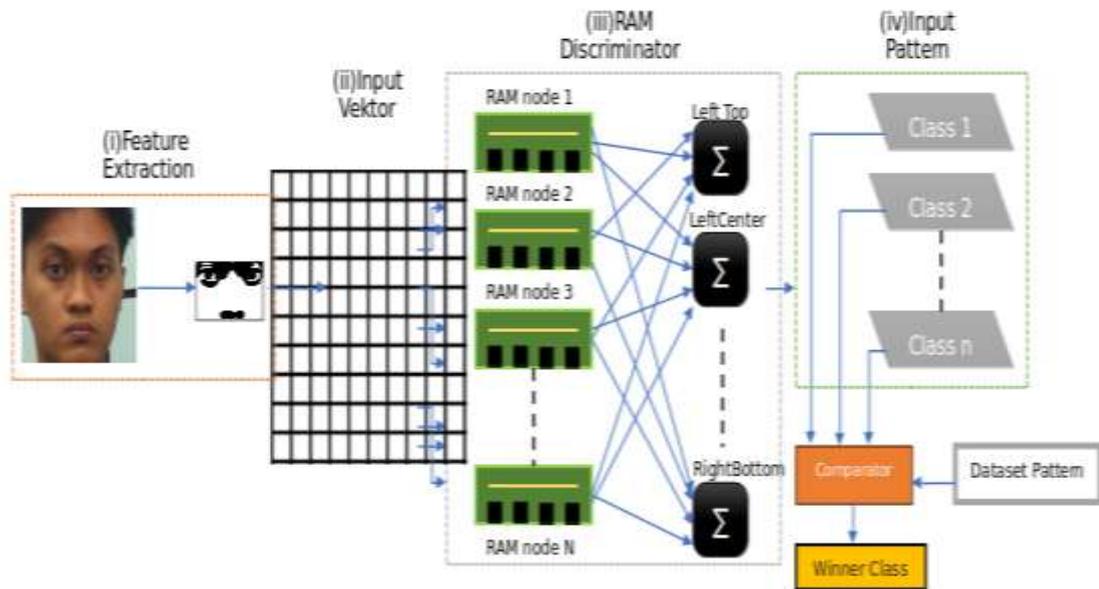


FIGURE 7. Face Recognition Architecture

The next stage is to design a WNN RFA architecture that meets the goals of this research. Figure 7 is an image of the WNN FRA architecture that will be used in this research. After getting an image of the face area, the data is extracted and adjusts the eye and nose area which will be converted to binary form and will enter into the input vector, in the input vector, each bit that is formed will be addressed to the same node which will be grouped into Ram discriminator, then this discriminator will be tested using comparator against each class which contains the previously created data set. The highest test result against the class will be determined on the comparator, the class that has the highest similarity to the input data will be determined as the Winnerclass.

### 3. METHODOLOGY

The robotic system's distance calculation is the first analysis phase in this test. A key element of the GPS-based navigation system is the computation of the bearing and distance between the robot's current location and the destination coordinates. In order to determine the shortest distance between two geographical coordinate points, the Haversine formula is utilized, which accounts for the Earth's curvature. The latitude and longitude difference between the robot's current location and the destination point—which is first translated into radians—is used in this formula. The following formula is used to compute the distance.

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

...(1)

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad \dots(2)$$

$$d = R \cdot c \quad \dots(3)$$

Where  $\phi$  is latitude,  $\lambda$  is longitude,  $R$  is the radius of the Earth (approximately 6371 km), and  $d$  is the distance between two points in kilometers (or converted to meters). Meanwhile, the bearing or direction is calculated to determine the angle between the north line and the imaginary line from the robot's position to the destination point. The bearing calculation uses the following trigonometric formula:

$$y = \text{atan2}(\sin(\Delta\lambda) \cdot \cos(\phi_2), \dots(4)$$

$$x = \cos(\phi_1) \cdot \sin(\phi_2) - \sin(\phi_1) \cdot \cos(\phi_2) \cdot \cos(\Delta\lambda) \dots(5)$$

The result of the atan2 function is then converted to degrees and adjusted within the range of  $0^\circ$ – $360^\circ$ . The bearing value will be compared with the compass heading to determine the robot's movement direction towards the target coordinates accurately. Here is the calculation to measure the distance between Unsri Bukit and Unsri Indralaya using that formula.

Unsri Indralaya = **3.216749°S, 104.648751°E**

Unsri Bukit = **2.985692°S, 104.732406°E**

To convert coordinates from degrees to radians, the formula used is:

$$\text{radian} = \text{derajat} \times \left(\frac{\pi}{180}\right) \dots(6)$$

Point 1:

- **lat1** = 3.216749°S =  $-3.216749 \times \frac{\pi}{180} \approx -0.056141$  rad
- **lon1** = 104.648751°E =  $104.648751 \times \frac{\pi}{180} \approx 1.826517$  rad

Point 2:

- **lat2** = 2.985692°S =  $-2.985692 \times \frac{\pi}{180} \approx -0.052141$  rad
- **lon2** = 104.732406°E =  $104.732406 \times \frac{\pi}{180} \approx 1.826822$  rad

To convert coordinates from degrees to radians, the formula is used: Calculate the difference in latitude ( $\Delta\text{lat}$ ) and longitude ( $\Delta\text{lon}$ ).

- $\Delta\phi = \text{lat2} - \text{lat1} = -0.052141 \text{ rad} - (-0.056141 \text{ rad}) = 0.004 \text{ rad}$
- $\Delta\lambda = \text{lon2} - \text{lon1} = 1.826822 \text{ rad} - 1.826517 \text{ rad} = 0.000305 \text{ rad}$

Calculate the value of  $a$  using the Haversine formula to calculate  $a$ ,

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \dots(7)$$

Substitute the values that have already been calculated:

$$a = \sin^2\left(\frac{0.004}{2}\right) + \cos(-0.056141) \cdot \cos(-0.052141) \cdot \sin^2\left(\frac{0.000305}{2}\right)$$

$$a \approx (8.0 \times 10^{-6}) + (0.9984 \cdot 0.9986 \cdot 1.16 \times 10^{-8}) \approx 1.92 \times 10^{-10}$$

Calculate the value of c (angle)

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \approx 2 \cdot \text{atan2}(1.38 \times 10^{-5}, 0.999999)$$

$$\approx 2.77 \times 10^{-5} \text{ rad}$$

Calculate the distance (d), The distance is calculated by multiplying the value of c by the radius of the Earth (R), which is assumed to be 6,371 km (or 6,371,000 meters):

$$d = R \cdot c = 6371000 \cdot 2.77 \times 10^{-5} \approx \mathbf{27319.8 \text{ meter}}$$

The Haversine formula is used to calculate the distance between Unsri Layo and Unsri Bukit, and the result is  $\pm 27.319$  meters. This figure shows a very close or exact match to the estimated distance of about  $\pm 27.3$  km found using digital mapping tools like Google Earth. The distance from Google Earth between Unsri Layo and Unsri Bukit is shown in Figure 8 below.

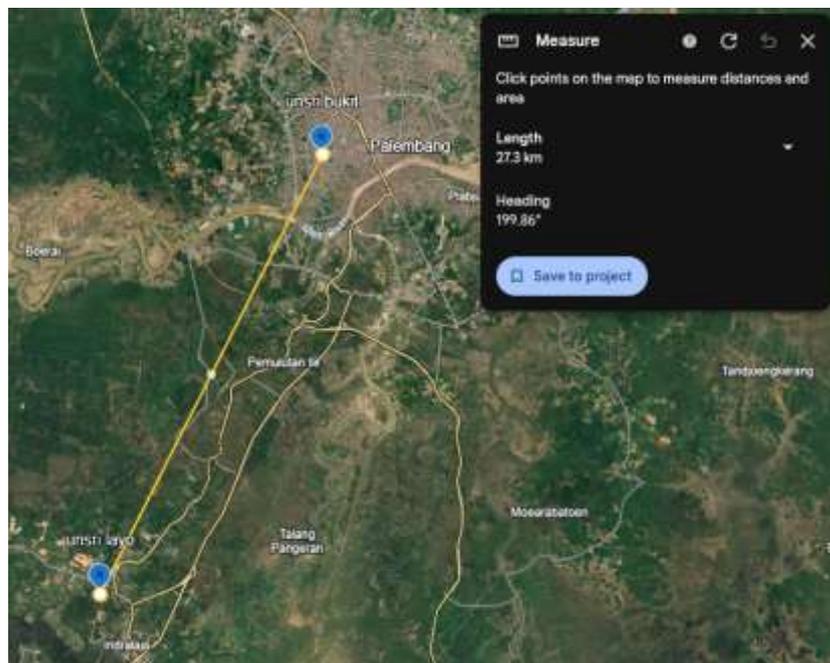


FIGURE 8. Distance between Unsri Layo and Unsri Bukit

Face detection comes next. The ability of the Viola-Jones algorithm to identify dynamic face regions will be evaluated in this test. The face's direction in this instance

**Ahmad Zarkasi, Kemahyanto Exaudi, Yoppy Sazaki, Londa Arrahmando Romadhona**  
**Implementation of Weightless Neural Network in Embedded Face Recognition for Eye**  
**and Nose Pattern Mobile Identification**

is dynamic, pointing in nine various directions: top left, center, bottom, top front, center, bottom, and top right, center, bottom. which nine directions are derived from the nine ROIs shown in Table 2 during the previous session. Top left, Center left, Bottom left, Top Center front, Lower front, Top right, Center right, and Bottom right are the ROI spots in the table, in that sequence.

TABLE 2.  
 Directions are based on the 9 ROI

Facing Direction	Top left	Center left	Bottom left	Top front	Center front	Lower front	Top right	Center right	Bottom right
Detection Result									
Resize Result									
	( 49x29 )	( 48x28 )	( 53x33 )	( 52x32 )	( 53x33 )	( 49x29 )	( 56x36 )	( 58x38 )	( 51x31 )

Following the acquisition of the face image. The image will then be resized to only show the area around the eyes and nose. We can refer to this as the resize image process, as shown in Table 3. An image of the eye and nose region with a resolution of 48x28 to 58x38 pixels is the end product of this procedure.

TABLE 3.  
 The result of Resize image

Facing Direction	Top left	Center left	Bottom left	Top front	Center front	Lower front	Top right	Center right	Bottom right
Raw Image									
Detection Result									

After receiving the face detection and resizing results. The robot motion system will use the value data on the frame based on these two outcomes as a reference. The robot will collect some data based on the face image capturing strategy, as illustrated in figure 9.



FIGURE 9. Example of Face Position Reading Result

The values of Inner, Inner\_y, Inner W, and Inner H are the data derived from the above image. This figure is derived from the results of resizing an image that only includes the nose and eyes and is indicated by a red box. The robot's movement will be based on this data, allowing the detected eye and nose area to reach the image capturing area. Following that, the robot will snap a picture in order to identify it.

Analyzing the facial recognition pattern test findings is the next stage. Two different kinds of test data are used to examine the identification (recognition) of the eye and nose region. Specifically, test data collected in identical conditions to those in which the data set was collected and test data collected in conditions that differed from those in which the data set was collected. The figure provides a graphic comparison of the highest and lowest accuracy. According to the test results above, Class left center 2 achieved the highest value through testing, with a similarity percentage of 98.87%. This class was the top winner in the test above, with an average similarity result of 92.98% for each class. The outcome of the pattern recognition accuracy test is shown in Figure 10.

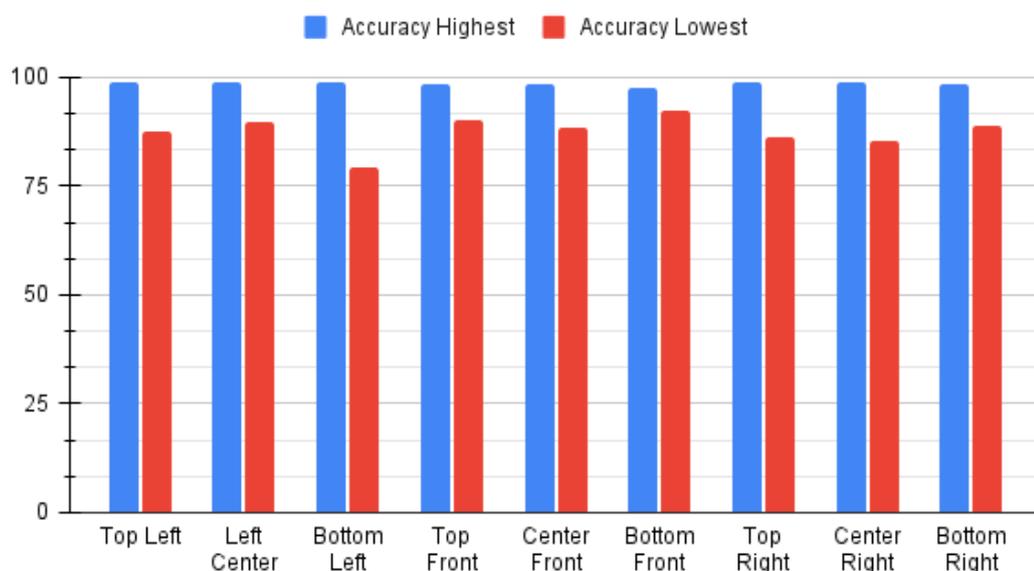


FIGURE 10. accuracy test

From the diagram Figure 10, it can be seen that the highest accuracy is stable near 100% in all face positions. While the lowest accuracy is only around 80% in all face positions. Furthermore, the comparison between the highest and lowest Ws at each face position can be seen in the following diagram:

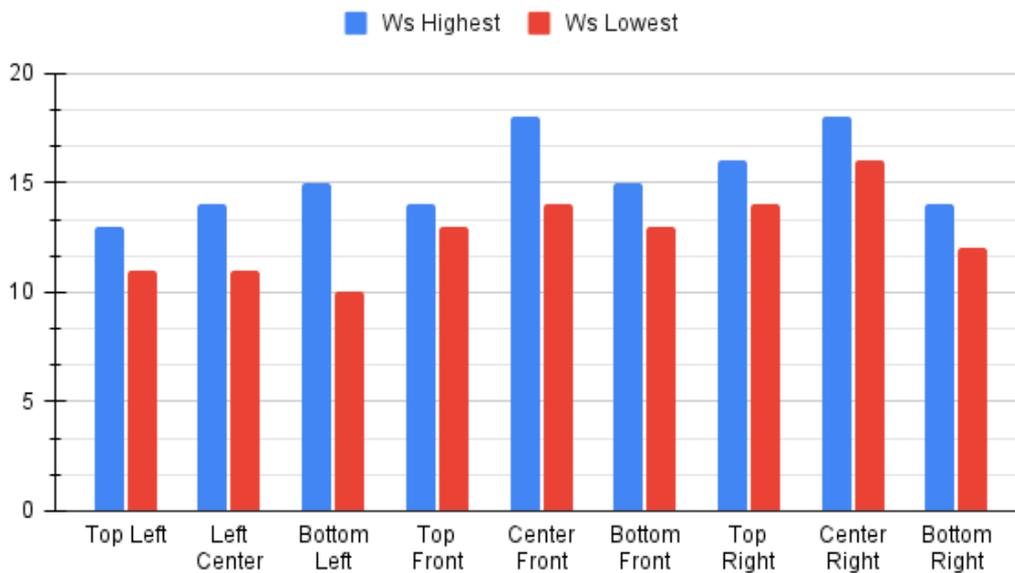


FIGURE 11. Execution time

From the figure 11, it can be seen that the highest Ws on each face is quite varied, ranging from 11 to 18, while the lowest Ws is also varied on each part of the face, ranging from 10 to 16. The next step is to evaluate the data in different conditions than when the data set was gathered. This results in a lower percentage of similarity because the environment varies, which affects the photos' output. Consequently, during testing, the percentage of similarity decreases. Table 5 presents the test results for the various conditions.

TABLE 5.  
 The test results of the test data with different circumstances

Eyes And Nose	Position	Accuracy Highest	Accuracy Lowest	Highest (time)	Lowest (time)
	Left Center	73,89	72,93	11	10
	Bottom Left	90,87	68,53	13	10
	Top Right	75,69	68,44	16	11
	Center Right	91,38	65,52	17	13
	Bottom Right	73,07	67,49	13	9

Besides the two test data mentioned previously. To determine how well the method can distinguish the nose and eye regions on various faces, a similarity test was also performed on a number of distinct faces that were selected from the author's friends' faces. The test results for faces that aren't used as data sets are listed here. It is evident from the aforementioned results that the percentage of similarity achieved when recognizing the author's friend's face is lower than that achieved when testing using the author's face test data. The test data from the author's face, which is 91.38% under various conditions, is significantly distant from the greatest percentage of similarity class, which is 75.69%.

#### 4. CONCLUSION

Overall, the research has produced positive findings. The robot's position was calculated by modeling and testing, and the results matched the calculated distance. The Weightless Neural Network approach has the potential to be effective. Under the same conditions, the test results with the author's face data set can achieve a similarity of 98.87%, and under other conditions, they can reach 91.38%. The author's friend's face data yielded the highest similarity value, 75.69%. When testing with data collected in the same state as the data set, the average class of test results is 92.98%; however, when testing with data collected under different conditions, it drops to 87.56%. Depending on the quantity of the data being compared, the duration per operation can vary from 11 ms to 17 ms.

#### ACKNOWLEDGEMENTS

The research publication of this article was funded by DIPA of Public Service Agency of Universitas Sriwijaya. Nomor SP DIPA-023.17.2.677515/2024, On November 24, 2023. In accordance with the Dean's Decree Number: 0154/UN9.1.9/LT/2024, On Juli 16, 2024," was supported by Universitas Sriwijaya. The first author would like to thank the Faculty of Computer Science, Strategic Pervasive Computing and Intelligent Embedded Systems Research Group (SPCIES-RG), and COMNETS RG, as well as Universitas Sriwijaya's Embedded Systems, Control System, and Robotic Laboratory.

#### REFERENCES

- [1] 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," vol. 12, no. 3, pp. 147–158, 2023..
- [2] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021, doi: 10.1016/j.neucom.2020.10.081.
- [3] 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,

doi: 10.1016/j.cose.2021.102416.

- [4] X. Zhang and H. Zhao, "Hyperspectral-cube-based mobile face recognition: A comprehensive review," *Inf. Fusion*, vol. 74, no. March, pp. 132–150, 2021, doi: 10.1016/j.inffus.2021.04.003.
- [5] F. Marcolin, E. Vezzetti, and M. G. Monaci, "Face perception foundations for pattern recognition algorithms," *Neurocomputing*, vol. 443, pp. 302–319, 2021, doi: 10.1016/j.neucom.2021.02.074.
- [6] H. Hosoya and A. Hyvärinen, "A mixture of sparse coding models explaining properties of face neurons related to holistic and parts-based processing," *PLoS Comput. Biol.*, vol. 13, no. 7, p. e1005667, Jul. 2017, doi: 10.1371/journal.pcbi.1005667.
- [7] C. Peng, W. Bu, J. Xiao, K. Wong, and M. Yang, "An Improved Neural Network Cascade for Face Detection in Large Scene Surveillance," *Appl. Sci.*, vol. 8, no. 11, p. 2222, Nov. 2018, doi: 10.3390/app8112222.
- [8] 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, doi: 10.1109/ICECOS.2018.8605215
- [9] Du W Z, Song Z H, Yan W B, Wu L G.: Fast image stitching under single camera rotate monitoring. *J. Journal of Image and Graphics*. 24--254(2016).
- [10] M. De Gregorio and M. Giordano, "Background estimation by weightless neural networks," *Pattern Recognit. Lett.*, vol. 96, pp. 55–65, Sep. 2017, doi: 10.1016/j.patrec.2017.05.029
- [11] 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.
- [12] Cardoso, D. O., França, F. M. G., & Gama, J. (2015). A bounded neural network for open set recognition. In *2015 International joint conference on neural networks, IJCNN 2015, Killarney*, July 12–17, 2015 (pp.1–7). IEEE.
- [13] Cardoso, D. O., França, F. M. G., & Gama, J. (2017). Weightless Neural Networks for open set recognition. In *Springer*, July 12, 2017.
- [14] Fischer, L., Hammer, B., & Wersing, H. (2016). Optimal local rejection for classifiers. *Neurocomputing*, 214, 445–457.
- [15] Fischer, L., Hammer, B., & Wersing, H. (2015). Efficient rejection strategies for prototype-based classification. *Neurocomputing*, 169, 334–342.

- [16] R. D. Cavalcanti, P. M. V. Lima, M. De Gregorio, and D. S. Menasche, "Evaluating weightless neural networks for bias identification on news," in *Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control, ICNSC 2017*, Aug. 2017, pp. 257–262, doi: 10.1109/ICNSC.2017.8000101.
- [17] Bartlett, P. L., & Wegkamp, M. H. (2008). Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9, 1823–1840.
- [18] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class Open Set Recognition Using Probability of Inclusion," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691 LNCS, no. PART 3, Springer Verlag, 2014, pp. 393–409.
- [19] W. Homenda, M. Luckner, and W. Pedrycz, "Classification with rejection based on various SVM techniques," in *Proceedings of the International Joint Conference on Neural Networks*, Sep. 2014, pp. 3480–3487, doi: 10.1109/IJCNN.2014.6889655.
- [20] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11. Elsevier Ltd, p. e00938, Nov. 01, 2018, doi: 10.1016/j.heliyon.2018.e00938.
- [21] M. A. Bhatti, R. Riaz, S. S. Rizvi, S. Shokat, F. Riaz, and S. J. Kwon, "Outlier detection in indoor localization and Internet of Things (IoT) using machine learning," *J. Commun. Networks*, vol. 22, no. 3, pp. 236–243, Jul. 2020, doi: 10.1109/jcn.2020.000018.
- [22] M. De Gregorio and M. Giordano, "CwisarDH+: Background Detection in RGBD Videos by Learning of Weightless Neural Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10590 LNCS, pp. 242–253, doi: 10.1007/978-3-319-70742-6\_23.
- [23] S. Nurmaini and A. Zarkasi, "Simple Pyramid RAM-Based Neural Network Architecture for Localization of Swarm Robots," *Artic. J. Inf. Process. Syst.*, 2015, doi: 10.3745/JIPS.01.0008.
- [24] Ahmad Zarkasi, Siti Nurmaini, Deris Stiawan, Bhakti Yudho Suprpto, "Weightless Neural Networks Face Recognition Learning Process for Binary Facial Pattern", *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, Vol. 10, No. 4, December 2022, pp. 955-969, ISSN: 2089-3272, DOI: 10.52549/ijeai.v10i4.3957.
- [25] Siti Nurmaini, Ahmad Zarkasi, Deris Stiawan, Bhakti Yudho Suprpto, Sri Desy Siswanti, Huda Ubaya, "Robot Movement Controller Based on Dynamic

**Ahmad Zarkasi, Kemahyanto Exaudi, Yopyy Sazaki, Londa Arrahmando Romadhona  
Implementation of Weightless Neural Network in Embedded Face Recognition for Eye  
and Nose Pattern Mobile Identification**

*Facial Pattern Recognition*". Indonesian Journal of Electrical Engineering and  
Computer Science (IJEECS), 2020, 22(2), pp. 125–135, ISSN: 2502-4752,  
DOI: 10.11591/ijeecs.v22.i2.pp733-743