Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 10, No. 4, December 2022, pp. 955~969
ISSN: 2089-3272, DOI: 10.52549/ijeei.v10i4.3957

Weightless Neural Networks Face Recognition Learning Process for Binary Facial Pattern

Ahmad Zarkasi[1], Siti Nurmaini[2], Deris Stiawan[3], Bhakti Yudho Suprapto[4]
[1,2,3]Faculty of Computer Science, Universitas Sriwijaya, Indonesia
[4]Faculty of Engineering, Universitas Sriwijaya, Indonesia

Article Info

Article history:

Received Jun 29, 2022
Revised Nov 25, 2022
Accepted Dec 29, 2022

Keyword:

Facial recognition,
Face detection,
Embedded platform
Haar Cascade,
WNN-FRA,
learning process.

ABSTRACT

The facial recognition process is normally used to verify and identify individuals, especially during the process of analyzing facial biometrics. The face detection algorithm automatically determines the presence or absence of a face. It is, however, theoretically difficult to analyze the face of a system with limited resources due to the complex pattern of a face. This implies an embedded platform scheme which is a combination of several learning methods supporting each other is required. Therefore, this research proposed the combination of the Haar Cascade method for the face detection process and the Weightless Neural Network Face Recognition Algorithm (WNN-FRA) method for the learning process. The WNN-FRA uses facial data at the binary level and for binary recognition. Moreover, the sample face data in the binary were compared with the primary face data obtained from a particular camera or image. The parameters tested in this research include detection distance, detection coordinates, detection degree, memory requirement analysis, and the learning process. It is also important to note that the RAM node has 300 addresses divided into three face positions while the RAM discriminator has three addresses with codes (00), (10), and (10). Meanwhile, the largest amount of facial region of interest (ROI) data was found to be 900 pixels while the lowest is 400 pixels. The total RAM requirements were in the range of 32,768 bytes and 128 bytes and the execution time for each face position was predicted to be 33.3% which is an optimization because it is 66.67% faster than the entire learning process

Corresponding Author:

Siti Nurmaini,
Department of Computer System,
Universitas Sriwijaya, Indonesia.
Email: sitinurmaini@gmail.com

1. INTRODUCTION

The facial recognition process is normally used to verify and identify individuals [1][2]. The development of different methods for this purpose has generated a lot of studies [3][4], especially to analyze the questions related to facial biometry at the level of accuracy and error [5][6]. A face detection algorithm has also been developed based on pattern recognition and heuristics to automatically determine the presence or absence of a face in an image or video frame [7][8][9]. However, a combination of several methods is needed for images with a level of complexity, accuracy, and dynamic movement. This involves combining the template matching method with a neural network [10][11][12][13] which is possible through an embedded engineering scheme. It is important to note that image processing with embedded schemes is a new technique developed in recent years [14][15][16] for systems with specific and real-time applications as well as limited resources including hardware and software. Examples of the implementations of this technique are fingerprint recognition and target position [17][18].

The algorithm used in image processing for embedded schemes is required to be capable of supporting and running effectively on the hardware system to be used. This is a normal problem in terms of programming implementation as indicated in object image programming with complex computations, weights, classifications, and data patterns. Therefore, the weightless neural networks (WNN) method is considered to be applied due to its good ability in processing data on embedded systems [19][20][21]. A pattern recognition technique associated with this method is the Wilkes, Stonham, and Aleksander Recognition Device (WiSARD) [22][23][24] which has a classification organized by a set of discriminators used to correctly evaluate the observations that correspond to the class it represents. Moreover, WiSARD exhibits several generative capabilities such as the possibility of obtaining class prototypes from DRASIW [25]. This prototype is a reference to explicitly gather pieces of information from the training data for later use. The generative nature of WiSARD further motivated the analysis of its use in introducing open set recognition.

One of the things being developed in open set recognition research [26][27][28][29] is to compare different calculations used in measuring the closeness of observations to the data. It was discovered from the Gaussian blob that WiSARD reflects the best specificity of data due to the smoother contour features of the matching data. The WiSARD contour was observed in a two-circle analysis to follow the pattern provided clearly in order to represent the overall data precisely. Moreover, it was detected in the two moons characteristics to have the closest match to the proximity value and produced precise data but had certain irregularities. The studies showed that the use of the WiSARD algorithm in several cases of set recognition is very superior due to the fact that data were processed in RAM nodes on several discriminators. It is important to note that RAM is a separate architectural structure in artificial intelligence and a basic component of the WNN method.

The main idea in WNN architecture is that memory cells can maintain their state before being replaced by new data. This is because the data persists at a certain level for the required time in the RAM lookup table. Most of the studies on open set recognition incorporated several improvements in the multilayer architecture but WiSARD, which is one part of the WNN method, has not yet implemented a learning process for facial patterns, especially, those in a moving condition. Therefore, this study proposes the WNN as a method to be applied to the facial pattern learning process. It is recommended to be combined with the facial image processing method in the preprocessing process because explicit facial pattern retrieval cannot be achieved using the WNN method. This is a new technique in the facial pattern recognition process and it is proposed to be installed in a mobile device that can recognize facial patterns at the binary data level.

2. METHODOLOGY

This section focuses on the methods used in conducting this research which include the stages of image processing, memory requirements, and implementation of the WNN algorithm. The first stage (i) is image processing through a face detection system in the form of a Viola-Jones method. The second stage (ii) was used to calculate the memory requirements of the facial image and this is considered important in the WNN algorithm. The third stage (iii) was used to implement the learning process and test the facial patterns using the WNN algorithm.

2.1. Face Detection Algorithm

The face detection system was used to reduce the false positive rate and increase the accuracy of detecting the faces, especially in complex background images [4][5]. The main purpose of the algorithm was to determine the position of the face in the image. Face detection is one of the domains of object detection [30] and the initial step required before proceeding to the process of face recognition, facial poses, facial expression analysis, and detection of facial components such as eyes, mouth, and nose [31]. Haar cascade algorithm, presented in Figure 1, is a traditional method of detecting faces.

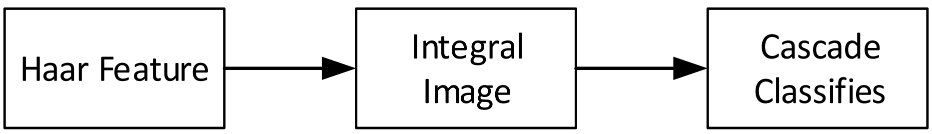

Figure 1. Face detection of Haar feature

Haar feature is based on Haar Wavelet while Haar waves are square waves with one high and one low interval which are further divided into one light field and one dark field. The Haar features can be determined by finding the difference between the average pixel value in the dark area and the light area, and the feature is confirmed to exist when the value of the difference between the two regions exceeds the specified threshold value. Moreover, the square in the Haar-like feature plane [32][33] can be calculated using the integral image which is a two-dimensional table in the form of a matrix of the same size as the original image. It is important to note that each element in the integral image is the sum of all pixel values located in the area to the left and above the elements in the original image. This makes it possible to count the number of rectangular areas at any position or scale in an image.

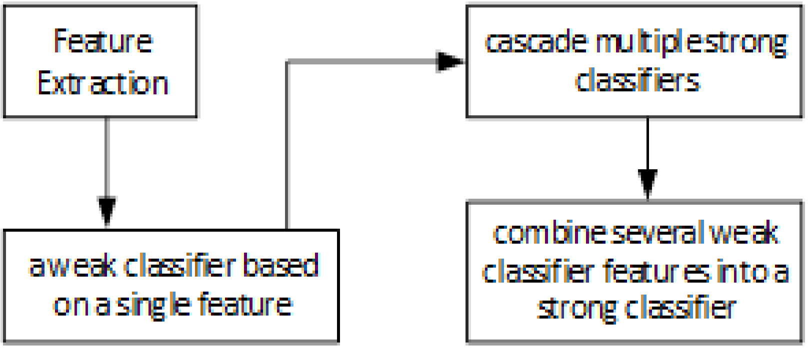

Figure 2. Block diagram of a Haar cascade classifier

Figure 2 indicates a face detection algorithm based on the Haar cascade classifier and it can be divided into three stages. The first is the extraction of the rectangular features of the face using an integral image. The second is to train the feature with the formation of a weak classifier followed by the combination of several accurate features to form a strong classifier. The goal is to be able to distinguish between faces and non-faces more accurately. The third is the placement of the strong classifier formed by important features and with a simpler structure in front in order to filter out a lot of non-faces in the sub-window and place the focus of detection on areas where there is little chance of human faces. This has the ability to greatly increase the speed of face detection.

2.2. WNN Face Recognition Algorithm (WNN-FRA)

WNN-FRA is an algorithm derived from the WiSARD algorithm to focus on the processing of only facial data which are binary data grouped in several N-tuple memory. The design of the algorithm has four process stages as indicated in Figure 3 and explained sequentially as follows:

(a) Preprocessor: This stage has two stages with the first being the environmental data collection process and the second being the face detection process. The environmental data obtained by the camera is the original image and this is followed by determining the existence of facial data in the image using the Haar Cascade algorithm. This is expected to produce face frames marked with square frames which are to be processed for the next stage.

(b) Feature extraction: This stage aims to obtain facial data in binary data and this requires going through several data conversion processes. The first is to convert the face RGB data into Grayscale data and then to binary data which is to be used in RAM processes. The binary data is to be grouped into several RAM groups called N-Tuples. This means the focus of this stage is to obtain facial data in binary data.

(c) Learning process. This stage is focused on grouping several N-Tuples into RAM discriminators which is where facial binary data are taught to recognize certain facial patterns using the WNN algorithm. Each data in the RAM discriminator recognizes how much data 1 is obtained in the memory addresses. The RAM discriminators are further grouped into several classes.

(d) Winner Class: This is the stage to determine the face pattern closer to the reference face pattern through the comparison of new data with reference data using the comparator mechanism which has the same function as the Euclidean Distance. The application of the comparator can be used to evaluate the closeness value of the input data pattern to the reference data pattern. The class that has recognized facial patterns or with better affinity is the winner class.

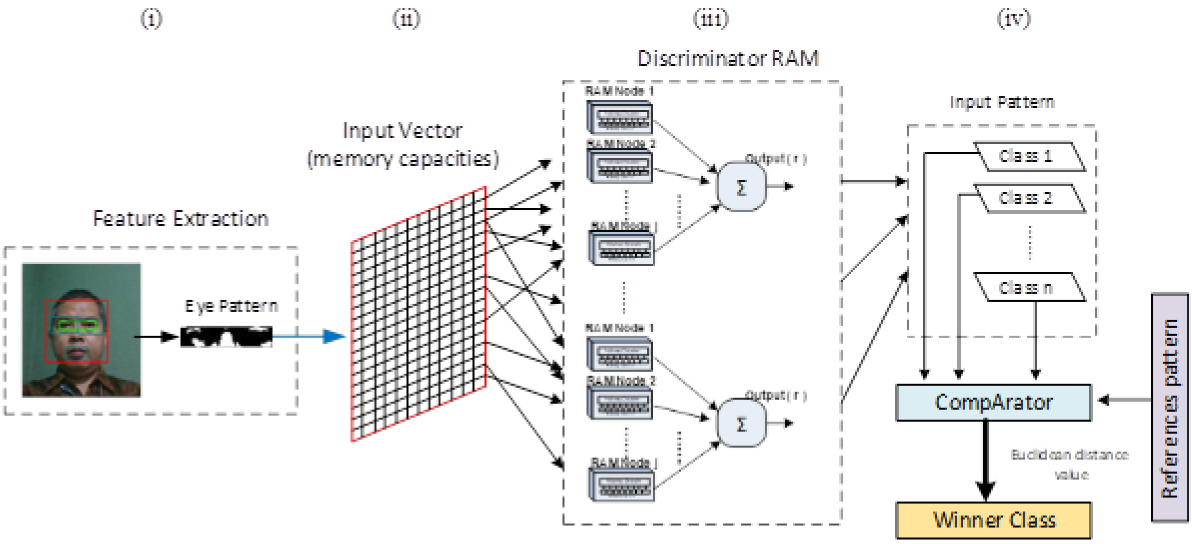

Figure 3. WNN-FRA algorithm

It is important to that the information stored in RAM during training is normally used to handle patterns that have not been seen before. The application of one of the invisible patterns as input makes it possible to read and sum up the contents of the RAM memory addressed by the input pattern using a summing device. This can be used to produce “r” which is the output having a value equal to the amount of logic 1 RAM also known as RAM discriminator response. This r reaches the maximum value of X when there is input to the training pattern and equals to 0 when no n-bit component of the input pattern appears in the training pattern. Moreover, the WNN-FRA architectural design is presented in Fig. 3 where the feature extraction section (i) was used to determine the facial pattern data in the form of binary data in certain pixel groups which include the position of the eyes. The data obtained from this section were used as input vector (ii) in the system. Furthermore, the RAM has several nodes to locate the memory of the pixel data obtained and the number was adjusted to the feature extraction data. The RAM node (iii) was later grouped into several RAM discriminators to process and study the data on each node. These discriminators were also grouped into several classes to train the data to recognize certain input patterns in the comparator(iv). The class with the largest data similarity value (Euclidean distance value) to the reference face pattern was decided as the winning class (winner class).

2.3. RAM Memory Requirement Analysis

The WNN algorithm cannot be separated from the process of optimizing resource usage in the devices owned due to the possibility of implementing the algorithm platform on simple embedded devices. Therefore, it is necessary to analyze the quantity of memory needed in the facial pattern recognition process. It is pertinent to state that the calculated data can be lower or higher depending on the image being processed. Meanwhile, the pattern in this algorithm was processed using the N-tuple technique which is a group of memory to be used together and an n-tuple usually consists of 8 memories. The example of a face image in pixels is presented in Figure 3 (i) and (ii).

The pattern of the processed face image in the picture is marked with a rectangle which indicates the number of pixels of the face pattern. A pixel is a collection of data addresses of face number values and it can be used to determine the memory needed. It was assumed that the memory pixel has L = x and H = y and this implies the total pixels used are X*Y pixels while the usable n-tuples is 8 and these were used to calculate the amount of memory required for the face pattern as indicated in Table 1. The inputs, x and y, are the size of the specified face image pixel, and the image was processed to obtain the face area, N-tuple (n) is a method of grouping RAM nodes into one group which is 8 nodes, p is the total number of pixels which was determined by multiplying x and y, P/n is the number of all n-tuples and obtained by dividing the number of pixels with the number of RAM nodes used, 2^n is the number of data bit widths used, where n is the number of RAM nodes in the n-tuple, r is the amount of memory used in bits, and total RAM required (r/8) is the amount of memory used in bytes.

2.4. Discriminator Architecture

The process layers in the WNN algorithm are similar to conventional neural networks with each designed to have functions to support the expected results. The WNN algorithm designed in this research has several stages which include the input vector, number of RAM nodes, number of RAM discriminators, training process, testing process, and determination of the output pattern which is the closest pattern to the existing pattern at the training stage. Moreover, one layer has three main discriminators which are the front, right, and left layers in the proposed WNN architecture as indicated in Figure 4. It is important to note that the input vector is the output of image processing in the form of facial pattern images converted into binary images. The next step was to divide the image into three positions which are right, front, and left with the input vector occupying the address on the RAM node according to the face position discriminator. The data were later trained and tested to obtain a pattern which was subsequently compared with the pattern data in a certain class in order to determine the one having the best closeness to the thresholding value. Furthermore, the design and process of the WNN method applied in database extraction provided for 3 RAM discriminators which include the Front, Right, and Left discriminators with each having a two-bit code, 01, 10, and 11 respectively, according to the face position.

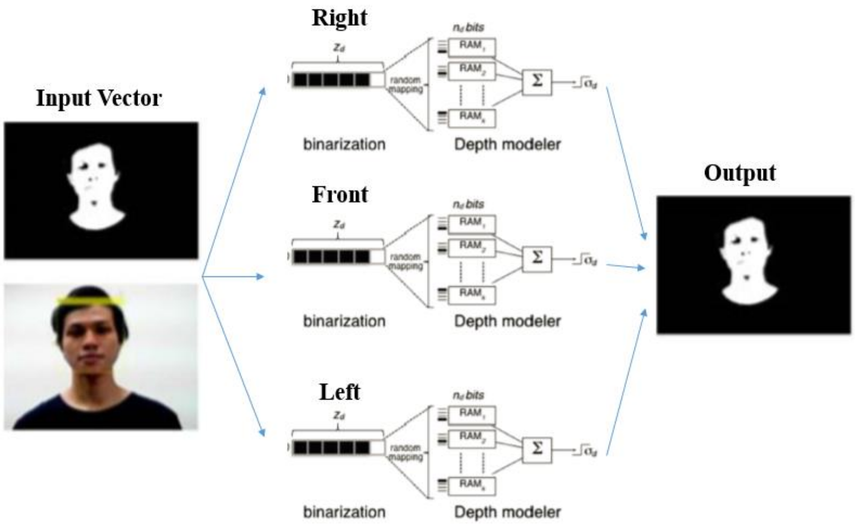

Figure 4. Discriminator Architecture

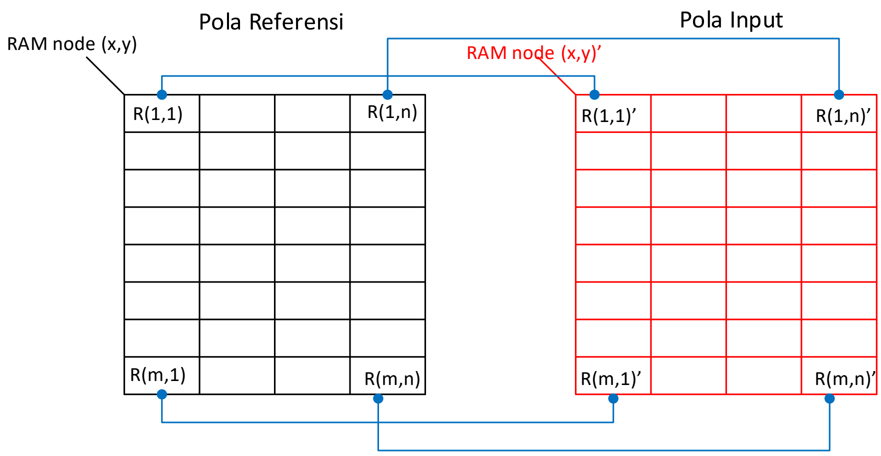

Figure 5. Immediate Scan Algorithm scheme (i)

2.5. Immediate Scan Algorithm

The direct recognition algorithm is a basic technique to check data on a comparator by comparing the input pattern data directly to the reference pattern at the bit level. The sequence of conditions in the process of identifying and checking the data are stated as follows

i. The input data with the same memory capacity (resolution) as the reference pattern were directly compared as indicated in Figure 5.

ii. The input data with a larger capacity than the reference pattern data required starting the checking process from the beginning to the end of the smallest pattern data as presented in Figure 6.
iii. The input data with a smaller capacity than the reference pattern data required initiating the checking process from the smallest initial data to the smallest final data according to the input pattern in Figure 6.

Figure 5 shows 2 feature patterns with the same memory capacity. It was discovered that the reference pattern has RAM node data (x*y) while the input pattern has RAM node data (x*y)'. The data checking process was adjusted to the position of each data RAM node at the beginning and the end. The check started at the initial row RAM point R(1,1) to the RAM point R(1,n) and proceed with the last line point of RAM labelled R(m,1) to R(m,n).

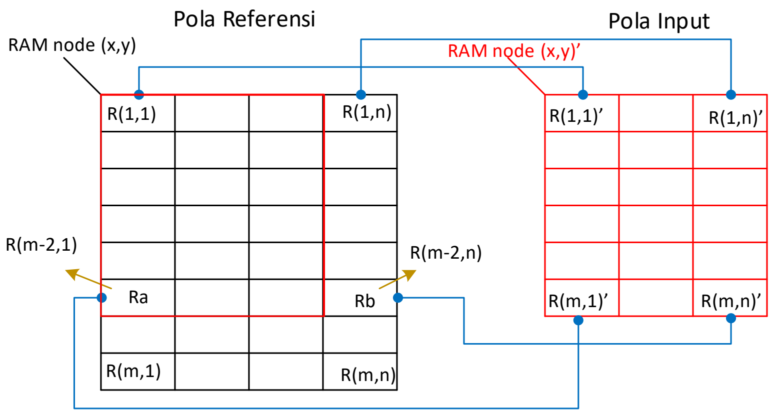

Figure 6. Immediate Scan Algorithm scheme (ii) and (iii)

Figure 6 shows that the reference pattern data is smaller than the input pattern data. This problem required that the check be completed based on the smallest pattern which is the reference pattern in this case. The check for the initial line was at the point RAM node R(1,1) to RAM node R(1,n) which is equivalent to RAM node R(1,1)' to RAM node R(1,n-2). Meanwhile, for the last row, it was in the position of RAM node R(m,1) and RAM node R(m,n) which is equivalent to RAM node R(m-1,1)' to RAM node R(m,n-2)' and the same trend was observed for the reverse data.

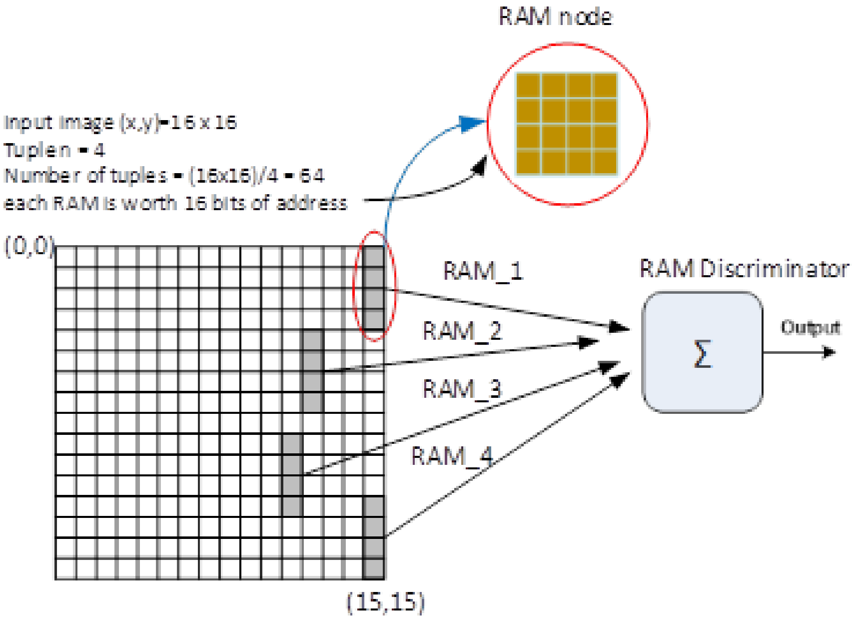

Figure 7. Memory requirement mapping

3. RESULTS AND DISCUSSION

This section explains the process of analyzing and thoroughly validating the data obtained to obtain the best results. The analysis was generally conducted to produce the RAM requirements, facial image processing, and pattern recognition using the WNN method.

3.1. Memory Requirement Analysis

This test was used to obtain a predictive value of RAM requirements from each detected facial pattern as one of the important parameters in the WNN algorithm. It was conducted after the pixel coordinate prediction value has been determined. The memory mapping structure of a frame is, therefore, presented in Figure 7.

The input in the picture has pixel coordinates (x,y) up to (16,16) while the number of input images obtained was 256. Moreover, the bit width of the n-tuple used was 4 and this provided 16 data memory addresses on the RAM node. The total number of n-tuples was 64 which was obtained by dividing the input image by the N-tuple bit width and this means the total number of RAM nodes of the pattern was 64. This was followed by the determination of the number of RAM discriminators. The sample showed that one RAM discriminator has 4 RAM nodes and this indicates the number of RAM discriminators in this research was 16 pieces. The result of the memory requirement formula analysis is presented in the following Table 1.

Table 1. Memory requirement formula

x	y	N-tuple	number of pixels	number of tuples	number of bits	Memory size	Total RAM required
		(n)	(P)	(P/n)	(2^n)	r = (P*2^n/n) bits	(r/8) bytes
256	256	4	65536	16384	16	262144	32768
16	16	4	256	64	16	1024	128

Figure 8 is an example of an analysis. It was discovered from the table that the largest pixel size was 55,720 pixels at resolution (280x199) and the smallest pixel was 20,164 pixels, at the resulution (142x142). This indicates the total RAM requirements are in the range of 32,768 bytes while the image analysis results showed that the amount of RAM required is 32 kbytes. Meanwhile, the face used in the coordinate test is presented in Figure 9 as primary image information.

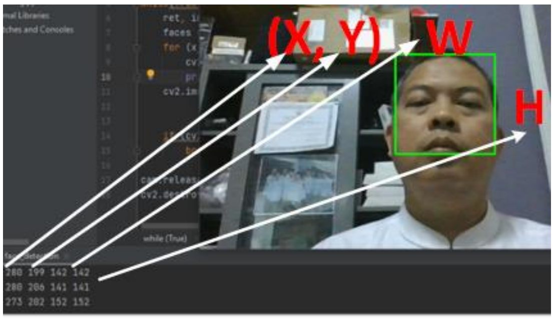

Figure 8. Coordinate test

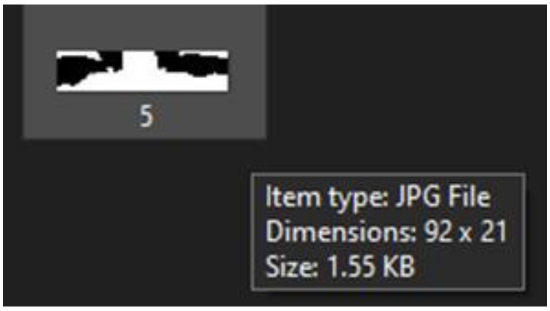

Figure 9. Front position reference pattern

The (x* y) data in Table 2 were obtained from the facial recognition test and the results of the analysis in Figure 8. The memory sizes obtained for facial feature patterns are not the same and this is associated with the continuous change in the position of the face when the camera captures images in real time. For example, Figure 7 shows that the pixel range, x, obtained for front facing pattern was 92 while y had 21 and the required memory capacity is 1932 bits. The memory requirements for the front, left and right face features are, therefore, presented in the following Table 2.

Table 2. Facial pattern memory requirement analysis

Face position	x	y	N-tuple (n)	Memory Capacity (P)	Number of Tuple (P/n)	Number of bit (2^n)	Memory Size r = (P*2^n/n) bits	Total memory requirement (r/8) bytes	
Front	92	21	8	1932	241.5	256	123648	15456	15.5K
Right	107	24	8	2568	321	256	164352	20544	20.5K
Left	98	22	8	2156	269.5	256	137984	17248	17.5K

3.2. Face Detection Result

Several parameters were used to test the facial image pattern including the detection distance, detection coordinates, and detection degree. The detection coordinates were used to obtain the best coordinates in the face detection process in order to avoid data loss. This is very important in ensuring the system is calibrated before the detection process is conducted. Meanwhile, the detection degree focuses on obtaining the best degree position on the right and left sides of the face. The result of the face detection process is also presented in Figure 8. The coordinates of the face in the picture are symbolized by X and Y while the pixel coordinates of the face image are determined from the values of W (width) and H (height). The initial coordinates of the face image are found at points X = 280 and Y = 199 while the coordinate value is located at the point W = 142 and H = 142, implying the total number of pixels is 20,164. It is important to note that the coordinate value continues to change because the face image moves dynamically as indicated by the initial coordinate values in the next data line. Moreover, the results of the facial recognition parameters tested at three positions of the facial pattern are presented in Table 3. It was discovered that the face detection distance in the front position is in the range of 40 cm to 80 cm with a rotation angle of 00 to 100. The coordinate prediction pixel (x,y) are (100,100) to (50,50) respectively and this means the largest number of pixels is 10,000 pixels while the lowest is 2500 pixels. The findings from the right and left facial pattern positions showed that the detection distance also ranges from 40 cm to 80 cm but the coordinate prediction data is from (80, 80) to (50, 50) and this denotes the largest number of pixels in face pattern detection is 6,400 pixels while the smallest is 2,500 pixels.

Table 3. Face recognition parameter testing results

Pattern position	Distance detection	Coordinate prediction Pixels (w,h)	Number of predictions pixel (x * w)	Coordinate prediction Degree
Front	40	(100, 100)	10,000	0 until 10
	60	(80, 80)	6,400	
	80	(50,50)	2,500	
Right	40	(80,80)	8,100	10 until 40
	60	(70,70)	4,900	
	80	(50,50)	2,500	
Left	40	(80,80)	8,100	10 until 40
	60	(70,70)	4,900	
	80	(50,50)	2,500	

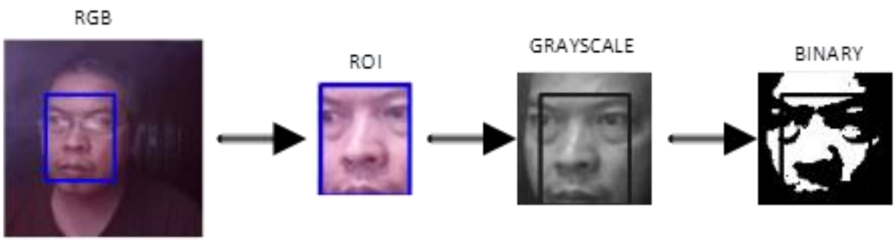

Figure 10. Facial extraction feature

The determination of the facial data was followed by the extraction of the facial features. According to the system design, the initial stage was to convert RGB data into grayscale data and then into binary data. The results of the processing are presented in the following Figure 10. The facial features obtained were reduced to only the eyes, mouth, and nose data. This area is called the region of interest (ROI) and is indicated by a blue rectangle in the figure. This was followed by conversion of the data from RGB to grayscale which can be used directly in the WNN process with the results compared to the binary data to determine the number of similar results. However, the memory required for each pixel is 8 bits which are considered to be higher than the capacity of the system. This is the reason it is necessary to optimize the memory in the future by dividing the memory capacity into 8 bits in order to ensure more efficiency and shorter execution time. Meanwhile, the grayscale data were converted into binary or black and white data as indicated in Figure 11.

Figure 11. ROI Binary data

The binary data were grouped into several N-tuple ram discriminators in order to be processed in the memory according to the design of the WNN-FRA algorithm. In this test, the division of the N-tuples was adjusted to the number of data bits to be analyzed. Moreover, the binary data obtained based on the conversion process in Figure 8 are 255 and 0 after which the 255 data were converted into number 1 in order to ensure the sequence of the face matrices is obtained in 1 and 0. The 1 data sequence was subsequently grouped into 8 data bits as indicated in the following Figure 12.

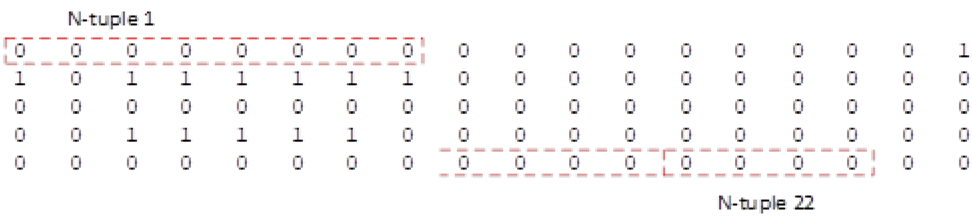

Figure 12. N-Tuple discriminator

Figure 13 shows that the data was grouped into a maximum of 112 N-tuples and each contains 8 bits of binary data. This was followed by the grouping of this data into a RAM discriminator to determine the data class as Front, Right, or Left position based on the binary code 01, 10, and 11, respectively. It is important to note that the code was also placed in the address bits in the MSB order of each RAM discriminator.

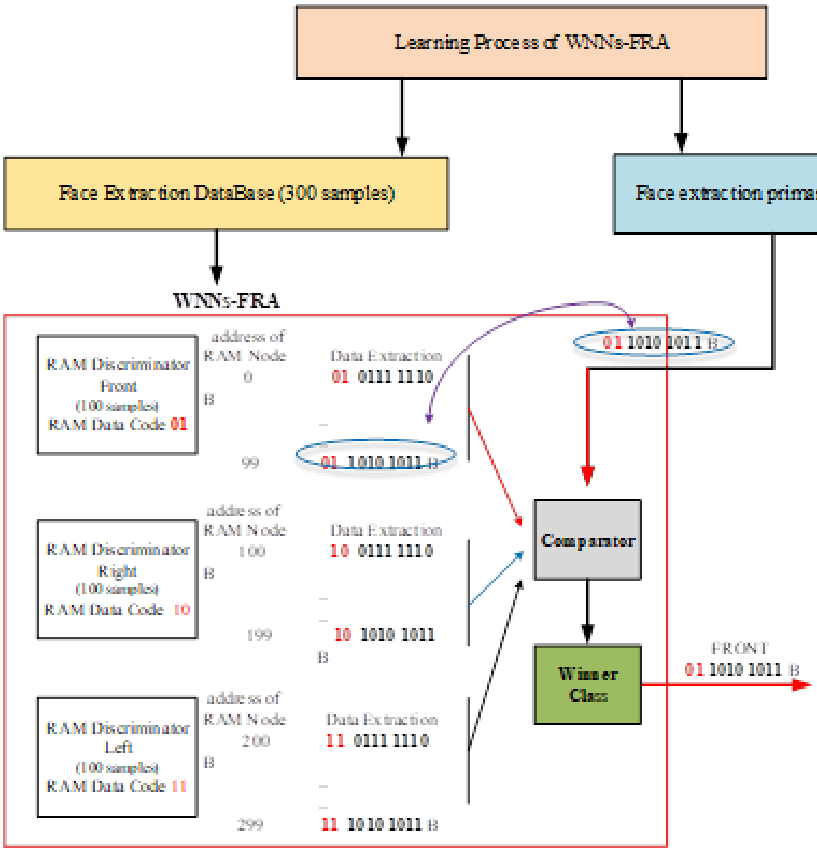

Figure 13. WNN-FRA learning process

3.3. WNN Face Recognition Result

The learning and testing process was initiated after the facial image has been processed using the WNN method as indicated in Figures 13. The process can be reduced to two sub-processes which include database extraction and primary data extraction. The extraction of the database has up to 300 samples of face images grouped into 3 face positions including front, right, and left with each having 100 face samples. Meanwhile, the primary data were adjusted to the position where the face image was obtained. It is important to note that all the data, both database and primary, were first converted into binary data. Moreover, the WNN method is in the extraction of the database according to the design and the process has 3 RAM discriminators which are the front, right, and left discriminators with each having a two-bit code, 01, 10, and 11 respectively, according to the position of the face. The position code for the facial pattern test is presented in Figure 11 and it was used to determine the face position being processed. It was adjusted to the code from the primary data which has the same code for the position of the facial image data. This led to the reduction of the 300 samples to 100, thereby, indicating the application of 1/3 of the samples. Therefore, the execution time became 66.67% faster when compared to the sequential learning process and this is considered to be very useful to process dynamic images or those with continuously changing positions.

The next process was to determine the RAM node address for each image position. It was discovered that each RAM discriminator has 100 addresses matched to the number of face samples and this signifies the front RAM discriminator addresses 0 to 99, the right addresses 100 to 199, and the left addresses 200 to 299. This was followed by the testing process which involves comparing the facial data from the primary and the database sources in a comparator block. It is, however, important to note that the binary data from facial ROI were used in this test with the face ROI coordinates recorded to be (25,25) to (30,30) for the front position and (20,20) to (25,25) for the right and left positions. This comparison was used to determine the primary data that has the closest match to the results in the face database and it was discovered from the two bits of MSB data that the front-facing data located in the front discriminator RAM has the closest match. Therefore, the front-end database was automatically compared with the primary data and the results were transferred to the winning class block to determine the data with the best closeness. The findings showed that the front face data with address 100 has this closeness and this was followed by the learning analysis on the WNN algorithm. Meanwhile, the results of the paw pattern tested in binary are presented in the following Table 4.

Table 4. Facial ROI pattern testing results in binary

Code	Face Position	Number of predictions ROI pixel (w * h)	Binary value of RAM Node (r = 10 bits)	The number of N-tuples Discriminator (r/8 bits)	Execution Time (%)
01	Front	625 to 900	10 0111 0001 to 11 1000 0100	75 to 112	33.3%
10	Right	400 to 625	01 1001 0000 to 10 0111 0001	50 to 75	33.3%
11	Left	400 to 625	01 1001 0000 to 10 0111 0001	50 to 75	33.3%

Table 4 presents the RAM nodes, N-tuple RAM discriminators, and the percentage of execution time for each group of face positions. The test was conducted on the face ROI area starting from the lowest pixel value to the highest to produce the detection result in binary data and it was discovered that the pixel data width is 10 bits for the RAM node such that the binary value for the front position (01) is 10 0111 0001 to 11 1000 0100 and the right (10) and left (11) positions have 01 1001 0000 to 10 0111 0001. Moreover, the number of N-tuples Discriminators for the front position is 75 to 112 N-tuples while the value for the right and left positions is 50 to 75 N-tuples. The face matching execution process is predicted to require 33.3% for each face position because the pattern has been classified based on the face position code and this infers the execution time can be optimized by 66.67% faster than the entire learning process. The next process was to test the pattern of facial features with the resolution or data capacity of the RAM node (x*y) set at (103x23) as shown in Figure 14. This means the entire input data (primary) for the face test in the bits was 2369, therefore, the data starts from bit 0 up to bit n to 2368.

Figure 14. Front position reference pattern (103x23)

Table 5. Result of Immediate Scan

Algorithm	reference pattern (facial features)	accuracy (%) (similarity)	execution time (s)
	Pattern_1	93.84	2.26
	Pattern_2	92.78	2.01
	Pattern_3	80.74	1.88
	Pattern_4	88.98	1.97
Immediate Scan	Pattern_5	90.06	1.77
	Pattern_6	91.82	2.17
	Pattern_7	89.21	1.86
	Pattern_8	90.21	1.97
	Pattern_9	87.75	3.03
	Pattern_10	85.61	1.68
average accuracy		89.1	
total time			20.6

Figure 15 shows the primary face pattern which was used as the input and tested against 10 references. The entire input or primary RAM node data was compared with the Front discriminator RAM dataset with code 01 using an immediate recognition algorithm technique. Moreover, the results of the direct pattern recognition test are presented in Table 5. It is important to note that the data were checked using the direct recognition technique which involves entering data directly without considering the size for both the input and reference data in the database. The reference data (dataset) include 10 facial feature patterns found in the database ranging from Pattern_1 to Pattern_10. A check was conducted to determine the level of accuracy or similarity of the input pattern data and the greatest accuracy was recorded in Pola_1 with 93.48% while the smallest was in Pola_3 with 80.74%. The average was found to be 89.1% which is higher than 75% and considered to be good.

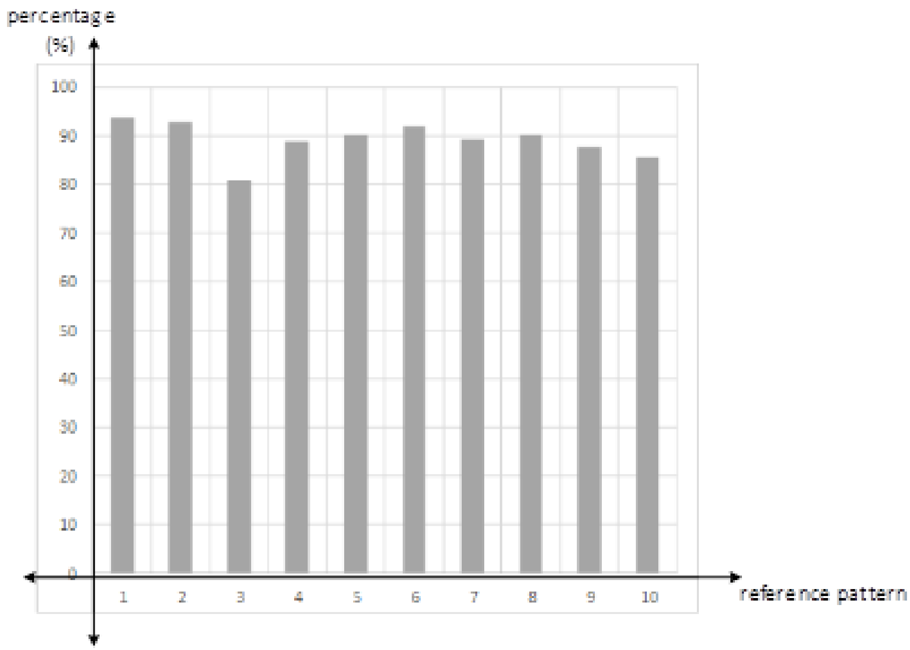

Figure 15. Data on the accuracy of the immediate scan algorithm

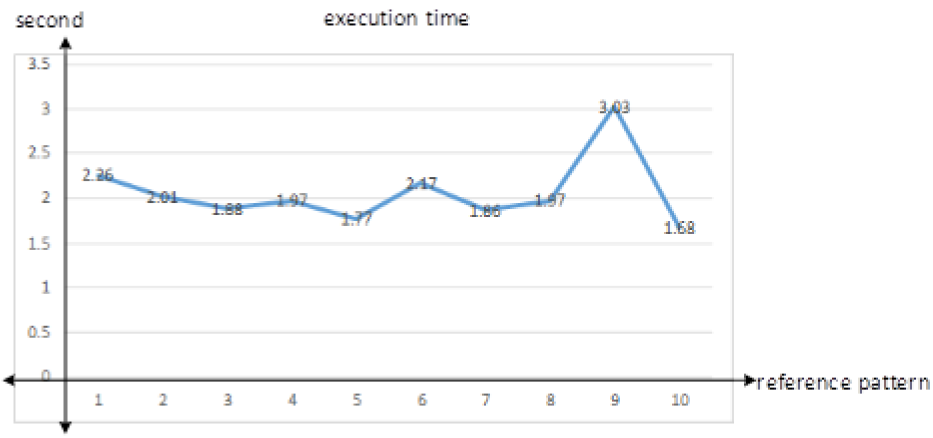

Figure 16. Execution time of the immediate scan algorithm

The execution time for each pattern checking process was also calculated in seconds and it was discovered that Pola_9 had the highest with 3.03 seconds while Pola_10 had the lowest with 1.68 seconds. The total execution time was recorded to be 20.6 seconds as indicated in Figure 16.

4. CONCLUSION

WNN FRA was developed from the WNN algorithm which is normally used to analyze the facial area learning process (ROI) at the binary level. The findings showed that the address required by the RAM node is 300 addresses which are divided into three face positions. The largest amount of facial ROI data was found to be 900 pixels while the least 400 pixels, thereby, making the total RAM requirements to be in the range of 32,768 bytes and 128 bytes. Furthermore, the execution time for each face position was predicted to be 33.3% and this indicates optimization of the execution time which is 66.67% faster than the entire learning process. It was discovered that Pola_1 had the highest accuracy with 93.48% while Pola_3 had the lowest withs 80.74% and the average was found to be 89.1% which was considered to be good because it is above 75%. Moreover, Pola_9 had the highest execution time with 3.03 seconds while Pola_10 had the lowest with 1.68 seconds and the total execution time was recorded to be 20.6 seconds.

ACKNOWLEDGMENTS

This research was funded by the Ministry of Research and Technology Republic of Indonesia through the Hibah Kompetitif 2019 Universitas Sriwijaya. The 1st author appreciates the Faculty of Computer Science, The Faculty of Engineering, Intelligence System Research Group, COMNETS Research Group, and also Embedded Systems, Control System, and Robotic Laboratory Universitas Sriwijaya for supporting this research.

REFERENCES

[1] P. Tome, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, “Facial soft biometric features for forensic face recognition,” *Forensic Sci. Int.*, vol. 257, pp. 271–284, Dec. 2015, doi: 10.1016/j.forsciint.2015.09.002.

[2] L. Sun, X. Liang, and Q. Zhao, “Recursive templates segmentation and exemplars matching for human parsing,” *Comput. J.*, vol. 57, no. 3, pp. 364–377, Mar. 2014, doi: 10.1093/comjnl/bxt100.

[3] J. Barbosa, W. K. Seo, and J. Kang, “ParaFaceTest: An ensemble of regression tree-based facial features extraction for efficient facial paralysis classification,” *BMC Med. Imaging*, vol. 19, no. 1, p. 30, Apr. 2019, doi: 10.1186/s12880-019-0330-8.

[4] 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.

[5] V. Maruthapillai and M. Murugappan, “Optimal Geometrical Set for Automated Marker Placement to Virtualized Real-Time Facial Emotions,” *PLoS One*, vol. 11, no. 2, p. e0149003, Feb. 2016, doi: 10.1371/journal.pone.0149003

[6] F. Fuentes-Hurtado, J. A. Diego-Mas, V. Naranjo, and M. Alcañiz, “Automatic classification of human facial features based on their appearance,” *PLoS One*, vol. 14, no. 1, p. e0211314, Jan. 2019, doi: 10.1371/journal.pone.0211314.

[7] M. Kasar, B. Vidyapeeth, M. M. Kasar, D. Bhattacharyya, and T.-H. Kim, “Face Recognition Using Neural Network: A Review Analysis of Multi-View Face Images and Recognition Through ANN View project Face Recognition Using Neural Network: A Review,” *Artic. Int. J. Secur. its Appl.*, vol. 10, no. 3, pp. 81–100, 2016, doi: 10.14257/ijsia.2016.10.3.08.

[8] L. Li, X. Feng, Z. Xia, X. Jiang, and A. Hadid, “Face spoofing detection with local binary pattern network,” *J. Vis. Commun. Image Represent.*, vol. 54, pp. 182–192, 2018, doi: 10.1016/j.jvcir.2018.05.009.

[9] H. Kim, H. W. Kim, and E. Hwang, “Real-time shape tracking of facial landmarks,” *Multimed. Tools Appl.*, vol. 79, no. 23–24, pp. 15945–15963, 2020, doi: 10.1007/s11042-018-6814-7.

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

[11] D. Jia, J. Cao, W. D. Song, X. L. Tang, and H. Zhu, “Colour FAST (CFAST) match: Fast affine template matching for colour images,” *Electron. Lett.*, vol. 52, no. 14, pp. 1220–1221, Jul. 2016, doi: 10.1049/el.2016.1331.

[12] L. Sun, X. Liang, and Q. Zhao, “Recursive templates segmentation and exemplars matching for human parsing,” *Comput. J.*, vol. 57, no. 3, pp. 364–377, Mar. 2014, doi: 10.1093/comjnl/bxt100.

[13] U. Park and A. K. Jain, “Face matching and retrieval using soft biometrics,” *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 3, pp. 406–415, Sep. 2010, doi: 10.1109/TIFS.2010.2049842.

[14] H. Liu and L. Wang, “Gesture recognition for human-robot collaboration: A review,” *Int. J. Ind. Ergon.*, vol. 68, pp. 355–367, Nov. 2018, doi: 10.1016/j.ergon.2017.02.004.

[15] 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, doi: 10.11591/ijeecs.v22.i2.pp733-743

[16] C. Qin, W. Zhang, F. Cao, X. Zhang, and C. C. Chang, "Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection," *Signal Processing*, vol. 153, pp. 109–122, Dec. 2018, doi: 10.1016/j.sigpro.2018.07.008.

[17] 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*, 2020, vol. 920, pp. 697–706, doi: 10.1007/978-3-030-13273-6_65.

[18] Z. Shi, H. Yu, and L. Kong, "Image stitching algorithm based on embedded system," in *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*, 2018, pp. 1174–1178, doi: 10.1109/ICMA.2018.8484530.

[19] 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.

[20] 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, pp. 1–6, doi: 10.1109/rICT-ICeVT.2013.6741494.

[21] 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.

[22] F. Ghazanfar and N. G. Haider, "Memory efficient sign language recognition system based on WiSARD weightless neural network technique," in *2016 2nd International Conference on Robotics and Artificial Intelligence, ICRAI 2016*, 2016, pp. 17–22, doi: 10.1109/ICRAI.2016.7791221.

[23] F. Rangel, F. Firmino, P. Machado, V. Lima, and J. Oliveira, *Semi-Supervised C lassification of Social Textual Data Using WiSARD*. .

[24] M. De Gregorio and M. Giordano, "An experimental evaluation of weightless neural networks for multi-class classification," *Appl. Soft Comput. J.*, vol. 72, pp. 338–354, Nov. 2018, doi: 10.1016/j.asoc.2018.07.052.

[25] 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.

[26] 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.

[27] D. O. Cardoso, J. Gama, and F. M. G. França, "Weightless neural networks for open set recognition," *Mach. Learn.*, vol. 106, no. 9–10, pp. 1547–1567, Oct. 2017, doi: 10.1007/s10994-017-5646-4.

[28] M. De Gregorio and M. Giordano, "An experimental evaluation of weightless neural networks for multi-class classification," *Appl. Soft Comput. J.*, vol. 72, pp. 338–354, Nov. 2018, doi: 10.1016/j.asoc.2018.07.052.

[29] D. O. Cardoso, F. França, and J. Gama, "A bounded neural network for open set recognition," in *Proceedings of the International Joint Conference on Neural Networks*, 2015, vol. 2015-September, pp. 1–7, doi: 10.1109/IJCNN.2015.7280680.

[30] J. Li and F. Yang, "Research on multi-robot scheduling algorithms based on machine vision," *Eurasip J. Image Video Process.*, vol. 2018, no. 1, 2018, doi: 10.1186/s13640-018-0355-x.

[31] 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

[32] F. Fuentes-Hurtado, J. A. Diego-Mas, V. Naranjo, and M. Alcañiz, "Automatic classification of human facial features based on their appearance," *PLoS One*, vol. 14, no. 1, p. e0211314, Jan. 2019, doi: 10.1371/journal.pone.0211314.

[33] C. Y. Low, A. B. J. Teoh, and C. J. Ng, "Multi-Fold Gabor, PCA, and ICA Filter Convolution Descriptor for Face Recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 115–129, Jan. 2019, doi: 10.1109/TCSVT.2017.2761829

BIOGRAPHY OF AUTHORS

Ahmad Zarkasi
Ahmad Zarkasi, M.T., was born in Palembang, August 25, 1979. He received Master of Engineering from Institute Technology Bandung (ITB) in majoring of computer engineering. in 2013. His research interests are in the area of Microprocessors, SoC (System-On-Chip), embedded system, and robotics. They include topics such as WNN in robotic system, Pattern Recognition for robotic mapping.

Siti Nurmaini
Prof. Dr. Siti Nurmaini was born in Palembang, August 2, 1969. Currently, she is a lecturer in department of computer engineering, Faculty of Computer Science, University of Sriwijaya, Indonesia. She is graduated from department of electrical engineering-UNSRI, Master of Engineering from Institute Technology Bandung (ITB) in majoring of control system and computer. She received a Ph.D. degree from Universiti Teknologi Malaysia-UTM majoring of intelligent control. She is very interesting in research area of the soft computing and intelligent system.

Deris Stiawan
Deris Stiawan received the Ph.D. degree in computer science from the Universiti Teknologi Malaysia, Malaysia. He is currently an Associate Professor with the Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion detection/prevention systems, and heterogeneous networks

Bhakti Yudho Suprapto
Bhakti Yudho Suprapto received the Dr. degree in Electrical Engineering from the Universitas Indonesia, Indonesia. He is currently an Associate Professor with the Department of electrical Engineering, Faculty of Engineering, Universitas Sriwijaya. His research interests include comtrol system, and robotics.