Implementation of Facial Landmarks Detection Method for Face Follower Mobile Robot

Ahmad Zarkasi[1], Fachrudin Abdau[2], Agung Juli Anda[3], Siti Nurmaini[4], Deris Stiawan[5],
Bhakti Yudho Suprapto[6], Huda Ubaya[7], Rizki Kurniati[8]
[1,2,3,4,5,7,8]Faculty of Computer Science, Universitas Sriwijaya, Indonesia
[6]Faculty of Engineering, Universitas Sriwijaya, Indonesia

***Abstract*— This paper presents a new technique for facesrecognition based on auto-extracted facial marks. Our landmarks are those related to the outer corner of the nose. With extracted landmarks, a triplet of areas and their associated geometric invariance are formed. Where later the points on the outer corners of the eyes and nose will be connected with lines that will form a triangle. Later the line length will be calculated using the Euclidean Distance formula so that the area value of the triangle can be obtained. Then the data obtained will be trained using the Support Vector Machine algorithm so that they can recognize faces. And later the system will be implanted into a mobile robot with raspberry.**

Keywords*— *Face Detection, Face Recognition, Facial Landmarks, Mobile Robot

I. INTRODUCTION

Humans are living things that have characteristics that can be used as a differentiator from other humans. One of the differences between humans and other humans is the face. Every human being has a different face shape. Each face has its own unique characteristics/features. Each individual's face contains different information. Examples are expression, gender, age, and race. Therefore, in biometrics technology or biological data recognition technology, faces can be used as identification. In the research that will be carried out, the feature extraction process is carried out using the Facial Landmarks Ponts algorithm to obtain the value of the features that will be used [1]. After feature detection and extraction is complete, the next step is the feature data training process with the data set that has been previously taken using the Support Vector Machine method. The face that has been recognized will be processed by the mobile robot which will determine the motion of the mobile robot.

II. RESEARCH METHODOLOGY

The research methodology consists of three stages, software development, hardware development, and software & hardware integration. Step 1: Software development consists of face detection and face recognition programmed using the Python language. Step 2: Due to limited resources, the Raspberry Pi will be used to perform the image processing stages. Meanwhile, the Open CR Board will be used to control the actuator system and on the mobile robot. Step 3: Testing and validation are carried out in the previous two stages to ensure whether the system is functioning properly and correctly. Software testing is done by testing the facial recognition method that has been used. Then hardware testing is carried out by testing various devices contained in the Mobile Robot and ensuring that the devices used are functioning properly. After that analysis and conclusions are carried out from the system that has been running. Figure 1 shows how this face tracking system works.

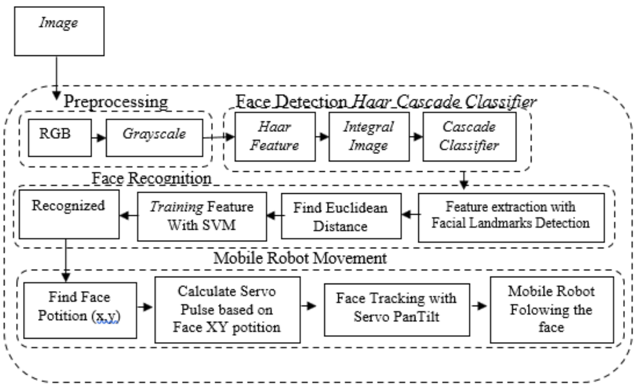

Figure 1. Block Diagram of the Face Tracking System

A. *Hardware Architechture*

In designing the hardware, the robot uses the RPi 3 Model B + with 1GB RAM capacity. The robot is designed to use 2 DC motors with the OpenCR Board as the motor controller. The robot camera is attached to a mounted PanTilt servo which is used as the neck of the robot and the camera used is the PiCamera V2.1 with a resolution of 8 megapixels. If there is a face on the camera, the robot will estimate where the robot will move by calculating the area of the detected face rectangle and the center point of the rectangle.

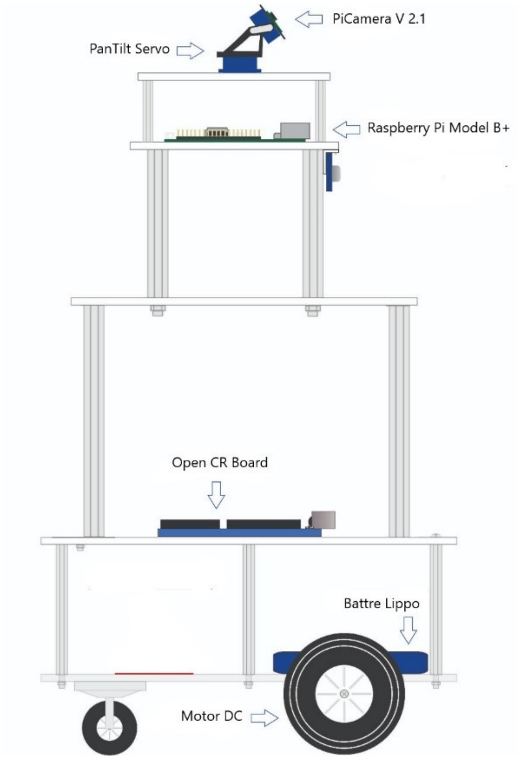

Figure 2. Mobile Robot Design

B. Algorithm Design

This section will describe the methods to be used in software development. The algorithm that will be used is to detect faces and recognize faces. Algorithm design can be seen in Figure 4. Input in the form of faces obtained on the camera. The image will be processed using the Haar Cascade method for face detection. After the face is found in the camera frame, the next process is feature extraction using the Facial landmarks detection method, after which the feature values obtained will be tested with a feature database that has been previously trained using the Support Vector Machine method. The output of this software development is a face that has been successfully recognized. After that, the next process will be carried out by the hardware.

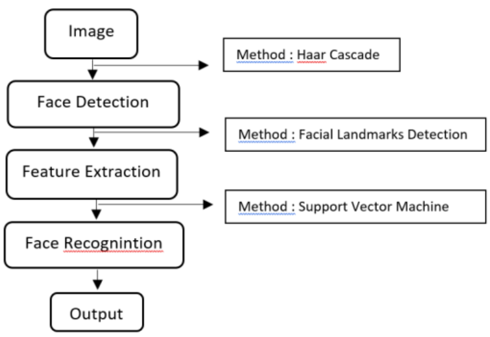

Figure 3. Algorithm Design

1. *Haar Cascade Calassifier*

Haar Cascade Classifier is an object recognition method that uses a feature from a digital image called the Haar-Like feature to detect an object. This method was adopted by Paul Viola and Michael Jones. This algorithm is known to provide results with very high speed and accuracy [2]. This algorithm uses 4 main factors, namely the Haar feature, Integral Image, Cascade Classifier, and Adaboost machine-learning. Figure 2.3 shows the Haar-wavelets work scheme

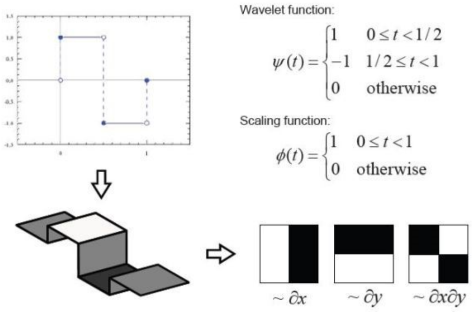

Figure 4. Skema kerja Haar-Wavelets

The Haar-Like feature processes the white box area and the black box area, where the boxes contain several pixels from a part of the image. The pixels in each region are added up and then the difference between the two regions is calculated [3]. A face image generally has areas with different light intensities, some dark areas, and some light areas, for example on the inside of the eye area with the cheeks. This Haar feature will detect and calculate the value between the eye area and the cheek area by calculating the pixel difference between dark and light colors as shown in Figure 2.4 below.

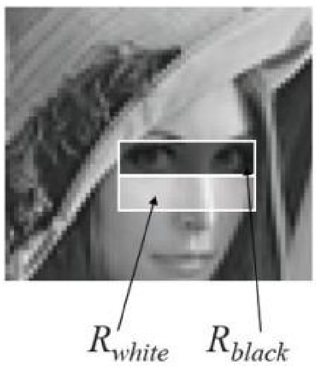

Figure 5. Skema kerja Haar-Like feature

2. *Facial Landmarks Detection*

The face landmark detection algorithm aims to automate the identification of the location of face points on face images. These points are dominant points that describe the unique location of facial components (eg, angles of the eyes, nose) or interpolation points that connect dominant points around facial components and facial contours [4]. To detect lock face structure on the person's face. This involves localizing the face in the image. We can do face detection in several ways. You can use the Haar Cascade XML file built-in OpenCV or even TensorFlow or use Keras. Here, in particular, We need to implement a special HOG (Histogram of Gradients) and Linear SVM (Support Vector Machines) object detector for face detection tasks. We can also do this by using a deep learning-based algorithm which is built for face localization. Algorithms will also be used to detect faces in the image. We can get the bounding box of the face through several methods that we use the respective (x, y) coordinates of the face in the image.

We do have a variety of face landmark detectors, but each method will try to localize and also label the face region the following will do.

- Nose
- Mouth
- Left eye
- Right eye
- Left eyebrow
- Mouth
- Right eyebrow

Face landmark detector pre-trained in the python dlib library to detect landmarks, is used to estimate the location of 68 points or (x, y) coordinates that map to facial structures. This index of 68 points or coordinates can be easily visualized in the image below.

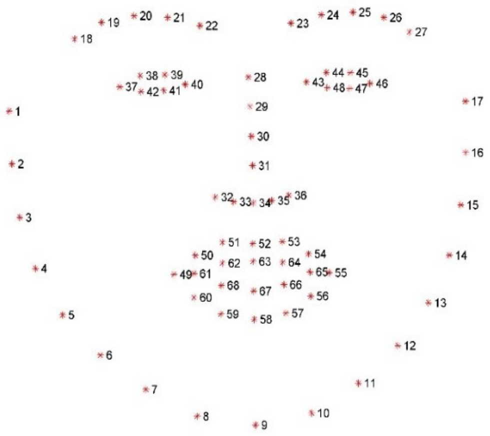

Figure 6. Facial Landmark Points

3. *Support Vector Machine*

SVM is a supervised learning set regarding the methods used for classification and regression. They belong to the general linear classification family. SVM simultaneously minimizes empirical misclassification and maximizes geometric margins. So SVM is called the Maximum Margin Classifiers. SVM is based on Structural Risk Minimization (SRM) [5].

Support Vector Machine (SVM) is a technique for making predictions, both in classification and regression cases. SVM has a basic principle of linear classifier, namely classification cases that can be separated linearly, but SVM has been developed to work on non-linear problems by incorporating concepts kernel in a high-dimensional workspace. In high-dimensional space, a hyperplane (hyperplane) will be sought which can maximize the distance (margin) between data classes[6].

Problems in classification for non-linear data are solved by kernel functions. Kernel functions in this method are often called kernel tricks. Kernel trick is a function that groups data from low dimension to high dimension. There are several kernel function options used in an application to solve the problem of non-linear SVM methods [5]. Where will be shown in the table below:

Tabel 1. Kernels Function

Kernel Name	Kernel Fuction
Linear	$K(x, y) = x.\, y$
Polynomial	$K(x, y) = (x.\, y + c)\, d$
Gaussian RBF	$K(x, y) = exp\ (-\|x - y\|\ 2\ 2.\ \sigma\ 2\ )$
Sigmoid	$K(x, y) = \tanh(\sigma(x.\, y) + c)$
Invers Multikuadrik	$K(x, y) = 1\ /\sqrt{\|x - y\|\ 2 + c\ 2}$

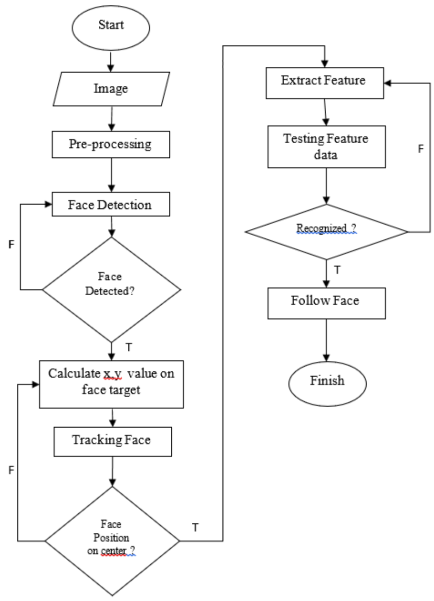

Figure 7. Software Development Flowchart

From the results of feature extraction, the x and y values of the outer blade points of the eyes and nose will be obtained. the point will be connected with a line to form a triangle. Then display each side of the triangle will be searched using the Euclidean distance formula, after that the area of the triangle is calculated which will be used as a feature. So that 4 features are obtained, namely the width of the detected face frame, the distance between the two corners of the eye, the length of the point from the nose to the midpoint of the connecting line of the eyes, and the area of the triangle formed by the outer point of the corner of the eye and nose.

$$E = \sqrt{|x_1 - x_2|^2 + |y_1 - y_2|^2}$$

III. EXPERIMENTAL RESULTS

In this section, the test results consisted of facial recognition testing, face tracking for Mobile robot movement.

A. *Facial Recognition*

This test aims to see if the system can recognize faces based on the 4 features described in the previous section. The test was conducted with 2 people, where each person was given 10 facial images. Where the feature value of the captured face will be tested with a database that has been previously trained. The results of facial recognition can be seen in table 2. Figure 6 is a test for facial recognition images.

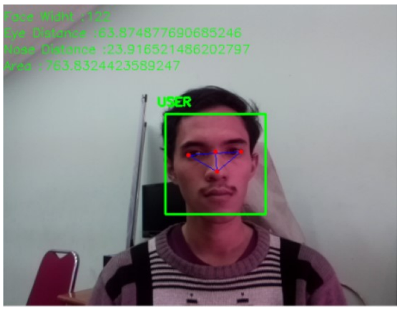

Figure 8. Facial Recognition with User Face

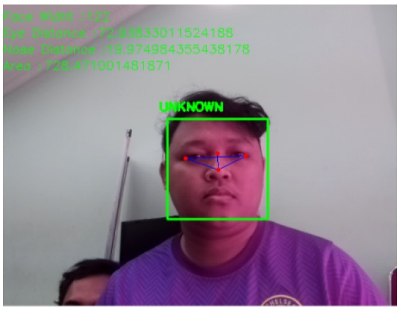

Figure 9. Facial Recognition with UNKNOWN Face

Table 2. Result of facial recognition testing

Tester	Accuration		Role
Agung	**True**	**False**	
1	I		USER
2	I		USER
3	I		USER
4	I		USER
5	I		USER
6	I		USER
7		I	UNKNOWN
8	I		USER
9	I		USER
10	I		USER
Yusuf			
1	I		UNKNOWN
2	I		UNKNOWN
3	I		UNKNOWN
4	I		UNKNOWN
5	I		UNKNOWN
6	I		UNKNOWN
7	I		UNKNOWN
8	I		UNKNOWN
9	I		UNKNOWN
10	I		UNKNOWN

From the results of facial recognition testing above, 19 face samples were successfully recognized, and there was 1 sample that was not recognized. Thus, the percentage of successful facial recognition can be calculated as follows:

$$Success\ Rate = \frac{recognized\ face}{all\ sample\ faces} \text{ x } 100\%$$

$$= \frac{19}{20} \text{ x } 100\%$$

$$= 95\%$$

It can be seen from the table that there is an introduction error in User data, there is a possibility that the error is caused by the similarity of User data with UNKNOWN data contained in the database or caused by an error in detecting landmarks that can occur due to lack of lighting or shaking or vibration which affects the detection process facial landmarks.

In this facial recognition process, a data can be made by containing the processing time. From this data, a graph containing 200 processing time data can be made. The graph of the results of processing time can be seen in Figure 8 below:

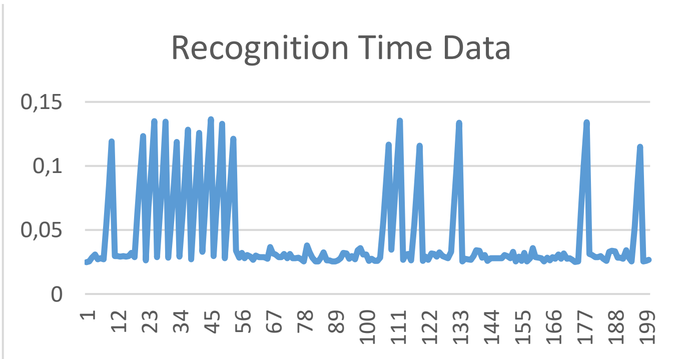

Figure 10. Face Recognition Processing Time Graph

B. *Face Tracking for Mobile Robot Movement*

After the system successfully recognized a face, the mobile robot will move based on the user's face position. In this study, the camera frame used was 480x368 pixels. There are upper, lower, right, and left frame. 2 servo motors mounted horizontally and vertically are expected to move up, down, right, and left based on the position of the face on the camera frame.

Servo movement will be determined by the calculation of the face position (x, y) plus half the pixel width of the face detected for the Pan servo, and added by half the pixel height of the face detected for the Tilt servo. The calculation of the change in Pulse Width is adjusted to the maximum Pulse Width and minimum Pulse Width on the servo. In the servo used, the maximum duty cycle is 2500, which means the servo will rotate to the right to a maximum of 180 degrees, and the minimum Pulse Width is 500, which means the servo will rotate to the left to the maximum reaching 0 degrees.

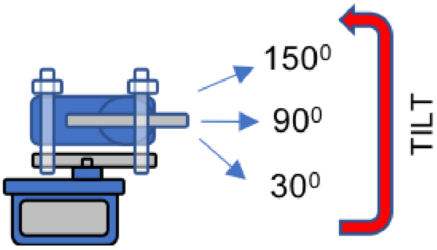

Figure 11. Motor Servo Position

If the face position is above of the camera frame, the Servo will move upwards, otherwise if the face position is below the camera frame, the Servo will move downwards. The result data for motor servo movement can be seen in table 3.

Table 3. Result of face tracking testing

No	Face Position		Servo Pluse	
	X	Y	Pan	Tilt
1	235	134	1683	1540
2	298	136	1626	1540
3	270	105	1568	1540
4	269	107	1510	1540
5	191	142	1510	1540
6	141	140	1610	1540
7	167	62	1610	1540
8	163	35	1610	1440
9	166	22	1610	1340
10	181	121	1610	1340

After the servo motor is able to follow the face position properly, the final step is to make the Mobile Robot move closer to the recognized face position. Two DC motors are used as Mobile Robot actuators. The movement of the Mobile Robot depends on the value of the Pan Servo pulse for right or left movement, then the face recognition results for moving forward to follow or stop for faces that are not recognized and the robot will start moving if the detected face width is less than 182 px if it exceeds then the robot will move backwards. The results of the Mobile Robot motion test can be seen in Table 4.

Table 4. Result of face tracking testing

No	Face position (*x*)	Detected face width	PWM Motor Right	PWM Motor Left	Robot Movement
1	124	81	0	0	Stop
2	344	122	0	0	Stop
3	217	54	0	0	Stop
4	242	81	75	75	Forward
5	244	81	75	75	Forward
6	238	81	75	75	Forward
7	371	81	15	-50	Turn Left
8	293	122	15	-50	Turn Left
9	71	122	75	75	Forward
10	161	122	75	75	Forward
11	145	182	75	75	Forward
12	150	182	75	75	Forward
13	152	182	75	75	Backward
14	107	273	-75	-75	Backward
15	110	273	-75	-75	Backward
16	42	273	-75	-75	Backward
17	11	273	-75	-75	Backward
18	140	122	-50	15	Turn Right
19	212	182	-50	15	Turn Right
20	240	122	-50	15	Turn Right

The robot will not move when the Ultrasonic sensor detects an obstacle less than 30 cm, with PWM on the Right Motor and Left Motor being equal to 0. And when the Ultrasonic sensor detects an obstacle more than 30 cm, the Robot will move. When the face position (x) is between 90.0 to 145.0, the Robot will move forward with the PWM of the two motors is 75. When the face position is in a position less than 90.0, the Robot will turn right, with PWM Motor Right 50 and Left Motor 100. Meanwhile, when the face position is in a position more than 145.0, the Robot will turn left, with PWM Motor Right 100 and PWM Motor Left 50.Based on the data from the table above, a comparison chart can be made between the PWM values of the two motors. This graph can be seen in Figure 13.

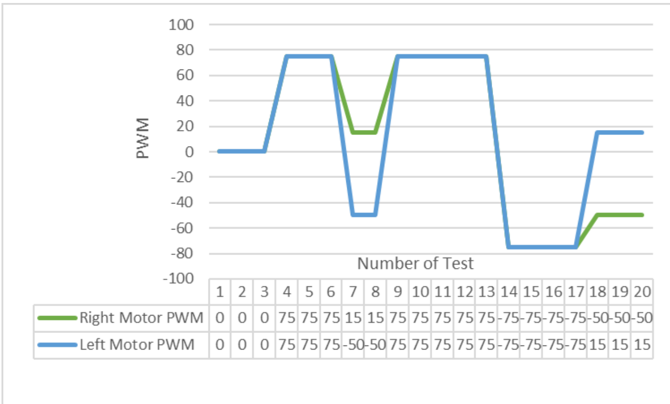

Figure 12. PWM Movement Graph

IV. CONCLUSION

From the tests that have been done, this system can extrac the feautre sucessly with Facial Landmarks Detection perform facial recognition using Support Vector Machine with a success rate of 95% with an error of 5%. The used servo motor can move from angle 0° to angle 180° based on the face position. And the DC motor move as which has ben programmed before where ther 5 movements namely forward, backward, stop, turn right and turn left.

ACKNOWLEDGMENT

The experiment to recognize faces gets a satisfactory result. The Mobile Robot used as hardware can also move properly when following faces. The weakness of this system is when the face gets insufficient lighting, the distance between the face and the camera is too far, and the complexity of the face. These three things are very influential when the system carries out the facial recognition process. Due to this, we recommend using another method which still has better accuracy. For further development, it is recommended to use

a camera with a higher resolution, so that the detection and face recognition results are better.

REFERENCES

[1] U. D. Rosiani *et al.*, “PENERAPAN FACIAL LANDMARK POINT UNTUK KLASIFIKASI JENIS KELAMIN BERDASARKAN CITRA WAJAH,” vol. 6, pp. 55–60, 2019.

[2] M. V. Gupta and D. Sharma, “A Study of Various Face Detection Methods,” vol. 3, no. 5, pp. 3–6, 2014.

[3] M. Margaretha, “OBJECT DETECTION USING VIOLA JONES AND TEMPLATE,” pp. 44–48, 2013.

[4] L. Wang, Y. Zhang, and J. Feng, “On the Euclidean Distance of Images.”

[5] J. Mase, M. T. Furqon, and B. Rahayudi, “Penerapan Algoritme Support Vector Machine (SVM) Pada Pengklasifikasian Penyakit Kucing,” vol. 2, no. 10, pp. 3648–3654, 2018.

[6] D. I. Pusphita Anna Octaviani, Yuciana Wilandari, “PENERAPAN METODE KLASIFIKASI SUPPORT VECTOR MACHINE (SVM) PADA DATA AKREDITASI SEKOLAH DASAR (SD) DI KABUPATEN MAGELANG,” vol. 3, no. 8, pp. 811–820, 2014.