Face Recognition Using Machine Learning Algorithm Based on Raspberry Pi 4b

Sunardi[a,1], Abdul Fadlil[a,2], Denis Prayogi[b,3,*]

[a]*Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia*
[b]*Master Program of Informatics, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia*
[1]*sunardi@mti.uad.ac.id;* [2] *fadlil@mti.uad.ac.id;* [3]*denis2107048007@webmail.uad.ac.id**
** corresponding author*

ARTICLE INFO

Article history:
Received 04 May 2022
Revised 30 May 2022
Accepted 15 June 2022

Keywords:
machine learning
Support vector machine
Face recognition
Raspberry pi
scikit-learn

ABSTRACT

Machine learning is one of artificial intelligence that is used to solve various problems, one of which is classification. Classification can separate a set of objects based on certain characteristics. The face is a part of the body with unique characteristics that can distinguish one person from another. For humans, recognizing someone from the face is easy, but for computers, it requires complex algorithms to solve this problem. In this paper, we propose a face image classification using a machine learning algorithm Support Vector Machine (SVM) with Principal Component Analysis (PCA) feature extraction which is implemented on a room security device with all processing using raspberry pi 4b. The dataset used is facial images collected from 15 employee respondents with 2100 training data and 525 test data. Image data is taken from the face with various pose variants with both eyes, nose, and ears visible. Training using raspberry pi 4b resulted in a model with the best score of 99% accuracy in 0.10 seconds, while testing of 525 data resulted in a model with a 99% precision score, 99% recall, and 99% f1 score. Testing the facial recognition device using the raspberry pi 4b with the SVM model can recognize facial images in real-time on the webcam and the sensors installed on the raspberry pi work according to their functions. The test results show that SVM can be applied properly to facial recognition devices as long as the facial features are still clearly visible.

I. Introduction

Machine learning is a part of Artificial Intelligence (AI) that is currently used to solve various kinds of problems. AI is a field of computer science whose goal is to make hardware and software function like humans who can think [1]. One of the fields in the computer world that currently applies machine learning to solve problems in computer vision. Computer vision is a combination of machine learning and AI [2] that focuses on the process of sensing computer-based vision. With computer vision, the computer seems to be able to see and identify objects that are around it.

Machine learning based on computer vision that is currently being researched and developed is biometric technology. Biometrics are human attributes that are not easy to duplicate and have distinctive characteristics. One of the biometrics that can be used for research and development is the face. Humans have faced with special characteristics that can distinguish them from other people. Current computer systems can classify human faces by utilizing machine learning. However, the problem with facial recognition today is how to get good accuracy in implementing it in various application platforms such as authentication, forensics, and security. To improve the accuracy of the facial recognition process, use machine learning-based algorithms. Some machine learning algorithms commonly used for classification include Principal Component Analysis (PCA), Support Vector Machine (SVM), Eigenface, and deep learning using neural network algorithms [3].

Face recognition to be able to recognize people is currently widely applied in various fields, one of which is security. The combination of automated embedded systems with machine learning

DOI: 10.29099/ijair.v7i1.321

produces smart devices that can work automatically with knowledge from machine learning widely applied to security. Research [4] designing a facial recognition system using the Principal Component Analysis-Genetic Algorithm (PCA-GA) algorithm for smart home door security. The PCA-GA algorithm can recognize faces with high accuracy of 90% which is built using raspberry pi processing. However, a study by Zhi and Liu [5] stated that the PCA algorithm is not optimal in performing classifications in facial recognition.

Nurhopipah and Harjoko [6] conducted facial recognition and motion detection research on 45 videos from CCTV. The method used for motion detection is Accumulative Difference Images (ADI), and face detection is the Haar Cascade Classifier (HCC) with Speeded-Up Robust Features (SURF) and PCA feature extraction. The late extracted features were trained using the Counter-Propagation Network (CPN). The results show a success rate of 76% for face detection and 60%, the combination of motion detection and face recognition causes a significant time delay.

Facial recognition can also be applied to forensic-based applications. In [7] examined face recognition by comparing facial sketches using the PCA method as feature extraction and using Euclidean to calculate the distance between the test image and the training image. Using a photo dataset and an image database from the Chinese University of Hong Kong (CUHK), sketch matching resulted in 76.14% accuracy, 91.04% precision, and 80.26% recall on training images. The sketch modification test got 95% recall accuracy and 100% precision.

Al-Aidid and Pamungkas [8] build a face recognition application using the HCC method to detect faces and Local Binary Pattern Histogram (LBPH) to recognize faces. LBPH is not machine learning, but an image matching process by matching the histogram values of the extracted images. The results of this study indicate that LBPH can recognize faces well on streaming video by comparing it with a stored image database.

In [9], build a facial recognition system for server room security using raspberry pi. The face recognition method used is Triangle Face, which is to calculate the distance between features such as eyes, nose, mouth, height, and face width. The resulting accuracy of this method is 75%, the positive error is 25%, and the negative error is 0%.

From the several studies presented, the comparison with previous research can be seen in Table 1.

Table 1. Comparison research

Year	Method	Image	Hardware	Accuracy	Comparison
2017 [9]	Triangle Face	30 sample images from 9 respondents	Webcam, Raspberry pi	Accuracy 75%	The raspberry pi device used as a reference in this study was upgraded to the raspberry pi 4b
2018 [8]	HCC, LBPH	120 sample images from 6 respondents	Webcam	the system can recognize faces at an optimal distance of 50-150 cm	The HCC face detection algorithm is a reference for detecting faces in real-time on the webcam
2018 [6]	HCC, SURF, PCA, CPN	45 CCTV video	CCTV	92.65% motion detection, 76% face detection, 60% combination motion and face recognition	The PCA algorithm for feature extraction becomes a reference for looking for characteristics in facial images
2020 [4]	PCA-GA	240 sample images from 8 respondents	Raspberry pi 3b, webcam, Sensor ping, relay, solenoid door lock	Accuracy 90%	The devices and sensors used are the references for building machine learning-based security systems.
2020 [7]	PCA, Euclidean Distance	Dataset CUHK		Accuracy 76.14%, Precision 91.04%, recall 80.26%	The algorithm used is a reference for optimizing the classification by performing feature extraction.

The previous research that has been described has not used machine learning that is implemented directly on the raspberry pi starting from taking face images, training images, and testing the accuracy of the resulting model. This study aims to develop a facial recognition system that is implemented on a room security camera for computer laboratory access at STMIK PPKIA Tarakanita Rahmawati using a raspberry pi 4b. All processes starting from taking people's face images, machine learning processes, and facial recognition system devices are controlled by the raspberry pi 4b as the data processing center. Furthermore, the development carried out is to increase the accuracy of facial recognition using machine learning algorithms Support Vector Machine (SVM) assisted by the sci-kit-learn library, python programming language, and Open CV for digital image processing. SVM is an algorithm that can be used for digital image classification by providing higher accuracy than other conventional methods even though the training data is small, efficient classifier in high-dimensional spaces, and the most memory-efficient algorithm [10], so it is very suitable if implemented on raspberry pi. The next development detects the presence of people in front of the camera by adding motion detection sensors based on infrared, using magnetic sensors to detect open doors, and buzzers for warning alarms. The development of this raspberry pi-based facial recognition system device can control who is allowed and not allowed to enter the laboratory room.

II. RESEARCH METHOD

A. Face Recognition

Face detection is the process of recognizing the shape of facial images in humans through matching existing faces such as curvature textures with digital images stored in a database. Face recognition is a computer vision technology in the field of biometrics that is used to recognize a person from a digital image or video and is an actively researched topic [11][12][13]. Facial recognition is a more versatile biometric technology than other options, which can be used in cases involving multiple people at once, such as in the case of a missing persons search or in the case of a wanted list [14].

B. Machine Learning

Machine learning is artificial intelligence that is implanted in the system with a focus on development to be able to learn automatically without having to be programmed by humans [13]. Machine learning consists of algorithms and statistical models that computer systems use to perform specific tasks by relying on patterns built from mathematical models based on data or samples from the training process [15].

Several important terms need to be known in learning machine learning, namely dataset, training, validation, and testing [16]. A dataset is a collection of samples used to create and evaluate machine learning models. Training is a term for a data set used for training a model. Validation is a collection of data that can be used to optimize the model while the training process is being carried out. Testing is a term for a set of data used to test a trained model.

Machine learning has three categories, namely supervised learning, unsupervised learning, and reinforcement learning [17]. This study uses the SVM algorithm which is included in the category of supervised learning.

C. Support Vector Machine

SVM was developed by Boser, Guyon, and Vapnik in 1992 and is a machine learning data classification technique that is guided through a training process called supervised learning. This algorithm compares the candidate set of a standard parameter of discrete values. SVM defines the boundary between two classes with the maximum distance from the closest data. SVM is usually used to solve prediction cases, both classification, and regression cases. SVM consists of a training and testing process to test how accurate the learning process and the resulting model are [18]. The characteristic of SVM is the dividing line between classes called hyperplane which can be seen in Fig. 1 [10].

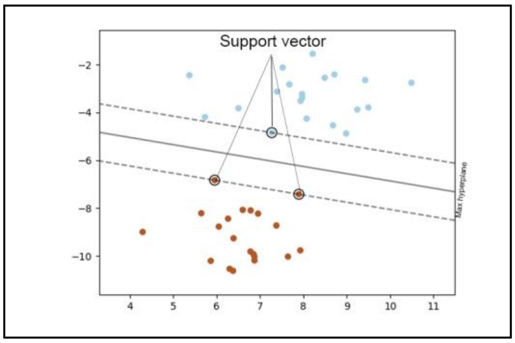


Fig. 1. Hyperplane in SVM

Hyperlane is between two different classes (integer and positive) and adjacent to the outermost object in each class. In SVM, the outermost object closest to the hyperplane is called a support vector. The support vector object is the most challenging object to classify because of its almost overlapping position with other classes. Because of its position, only support vector objects are taken into account to find the most optimal hyperplane in SVM. From Fig. 1, SVM uses a linear model with a general form using equation (1). Parameter y(x) is the output result, x is the input vector, w is the weight parameter, ∅(x) is the basic function, and b is the bias [19].

$$y(x) = w^T \emptyset(x) + b \tag{1}$$

The library used for the SVM algorithm in python is scikit-learn with the Support Vector Classifier (SVC) technique. SVC allows some data to be classified incorrectly [8] due to the complexity of the data (multi-class classification) so perfect vector separation cannot be performed. In this study, SVC uses a kernel based on Radial Basis Function (RBF), parameter cost (C), and gamma. The RBF kernel is suitable for use in image-based classification because the classification of data sets cannot be separated by straight lines. Cost (C) is a penalty parameter to minimize misclassification. While the gamma controls the distance between vector points during training.

D. *Raspberry Pi Model 4b*

Raspberry pi is a mini-computer or so-called Single Board Computer in which there is a System-on-a-Chip (SoC) Advanced RISC Machine (ARM) integrated with input-output ports. Raspberry pi is usually used for various functions such as microcontrollers, programming, digital image-based processing, server devices, and multimedia entertainment [9]. Fig. 2. is an example of the Raspberry pi model 4b board used in the research [20].

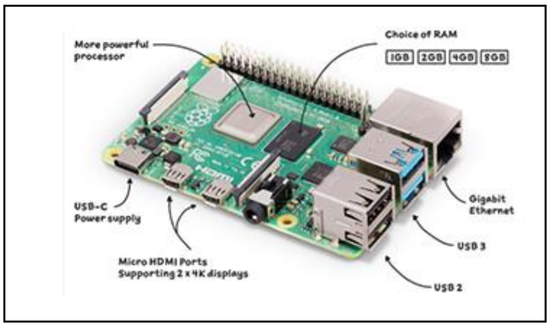


Fig. 2. Raspberry pi 4 model b

Raspberry pi 4b is one of the newest types that already have memory variants ranging from 1 GB to 8 GB and already supports 4K display output. Raspberry pi which is an embedded system similar to microcontrollers such as Arduino, Zigbee, and ESP8266. However, there is a major difference from the development system, namely the raspberry pi requires an operating system, and the programs are built to run on the operating system. While microcontrollers generally contain programs that run sequentially, there is no operating system [21].

E. Supporting Hardware

In addition to using raspberry pi, some components support this research to complete the security system created. Some components and their functions can be seen in Table 2. Seven main components support research. Raspberry pi 4b with 2 GB memory specifications is enough to process data on all security devices. The webcam is used to capture the face object, then a comparison process is carried out with the dataset from the model derived from machine learning. A Solenoid door lock is an automatic electronic locking device for security systems [22]. Passive Infra-Red (PIR) is a sensor to detects infrared radiation that is around it [23]. This sensor detects human objects' presence based on infrared emissions in front of the webcam camera.

Table 2. Research Hardware

No	Component Name	Type/Specification
1	Raspberry pi	4b, RAM 2 GB
2	Webcam	USB 720 HD
3	Selenoid door lock	12 volt
4	Passive infrared	HC-SR501
5	Magnetic sensor	MC-38
6	Relay Module	5 volt
7	Buzzer	5 volt

The MC-38 magnetic sensor is a sensor consisting of two modules that work using the electromagnetic principle, that is if they are close together, the switch is in a closed circuit condition, and vice versa [24]. This sensor is suitable when used to detect open doors or drawers. Buzzers are used for warning alarms if the door is open because they can produce sound vibrations from the electrical signal it converts [24]. The relay is a connecting switch and circuit breaker [25] that can be used to drive the door lock solenoid in this study because it requires 12 volts of external power from the adapter.

F. Design System

This research requires a face dataset that is used to carry out the training and testing process on machine learning to be able to classify faces. The data used are facial images taken directly from the camera using the Haar Cascade Classifier (HCC) algorithm. This algorithm can detect objects quickly and in real-time, including the image of a person's face, because it only depends on the number of pixels in the square of an image[11]. All training and test face images are sized at 200x200 pixels so that they are the same [26] so that they can be compared with each other [27]. then the image with the dimensions of the RGB layer (Red, Green, Blue) is labeled to distinguish each class and stored in a folder according to the class name. Table 3 describes the complete details of the number of facial datasets used in the study.

Table 3. Face Dataset

No	Description	Quantity
1	Data class	15
2	Face sample	2625
3	Training Data	2100
4	Testing Data	525

Based on Table 3, there are 15 different respondents used for the data class. Then, 175 facial images were taken for each sample, so the total face sample was 2625 data. Fig. 3 is a sample of each image from the respondent taken using capture from a webcam.

Fig. 3. Workflow of the training process

From the facial image data, it is made using a split technique for training by 80% or as much as 2100 data, and testing to test the training carried out is 20% or 525 data.

G. Software Design

The training and testing process uses a raspberry pi to obtain a dataset model that is tested to measure its accuracy. To build a facial recognition system using a raspberry pi, supporting software is needed as shown in Table 4.

Table 4. Supporting Software

No	Software	Version
1	Raspberry pi OS	11 Bullseye 64bit
2	Jupyter notebook	6.4.10
3	Python	3.9.2
4	Open CV	4.5.5
5	Scikit-learn	1.0.2
6	Joblib	1.1.0

The software in Table 4 must be prepared so that machine learning processing for the facial recognition system on the raspberry pi can work optimally. In general, the training and testing process on the raspberry pi can be seen in Fig. 4.

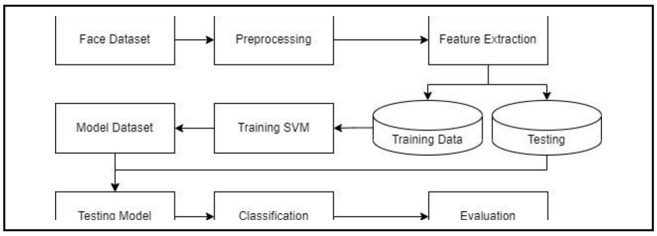


Fig. 4. Workflow of the training model testing process

Based on Fig. 4. The face dataset is processed at the preprocessing stage by changing the image size and converting the image from RGB to grayscale to reduce the dimensions of the data. The grayscale image is resized to a smaller size of 50x50 pixels to make the training process faster. The reduced grayscale image is then extracted to look for the characteristics of an image object that can distinguish it from the others [28]. Feature extraction uses PCA to reduce image dimensions into the Eigen feature space [29]. This is to display a significant variation of features from a known facial image called eigenface features [30].

The image resulting from feature extraction is divided into training data and test data with the percentage used being 80% for training data and 20% for test data. This data sharing is because machine learning requires data for training that will produce a model, then the model is tested to measure how well the resulting model is. The test results will see how well the model classifies the test data so that it can be evaluated using the evaluation method.

H. Evaluation

The problem that arises in machine learning is how to get a good and accurate data model from the training process. To get a model with good accuracy, an evaluation of the model [31] from the classification results is carried out using data testing [32]. Some terms in the evaluation process that need to be known are True Positive (TP) is positive data is predicted to be correct, True Negative (TN) is negative data is predicted to be correct, False Positive (FP) is negative data is predicted to be positive data, and False Negative (FN) is positive data predicted negative data.

Machine learning uses several tests to see the results of data testing such as accuracy, precision, recall, and F1-Score which is calculated using the following equations.

$$accuracy = \frac{TP}{Total\ Data} \tag{2}$$

$$precission = \frac{TP}{TP+FP} \tag{3}$$

$$recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1Score = 2\frac{Precission*recall}{precission+recall} \tag{5}$$

In addition to using the testing phase to see the results of the classification, an evaluation using a confusion matrix is also carried out. A confusion matrix is a method for evaluating classification methods [33] by seeing how much data is predicted based on TP, TN, FP, and FN.

I. Hardware Design

The design of the hardware used in the security system, in this case, exemplified room security based on face classification using a raspberry pi as a data processing center and assisted by sensors, which can be seen in the block diagram of Fig. 6.

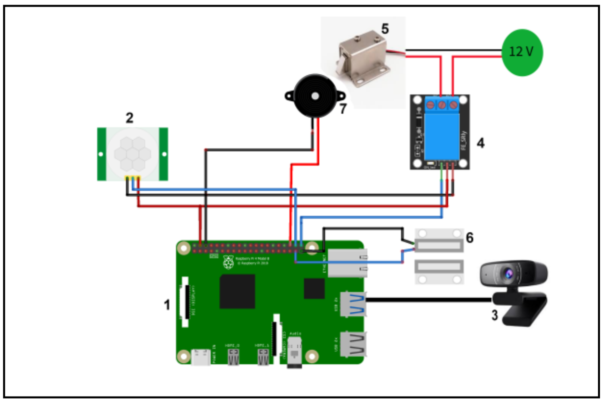


Fig. 5. Block diagram of the hardware design

Raspberry pi (1) as a security device processing center is connected to the PIR sensor (2) using a ground cable (GND) and digital pin number 38. Webcam (3) is connected to the USB port for capturing facial images. Relay (4) consists of three wires so that it can be activated, namely, the 5volt pin is connected to the 5volt raspberry pi, the GND pin, and the digital pin to connect and disconnect the electric current connected to digital pin number 40. A 12-volt electric current can come from the adapter and one of the wires is connected to a relay so that it can be activated and deactivated according to the program. The goal is to provide electricity to the door lock solenoid (5) so that the door lock can be opened. The door opens will make the magnetic sensor (6) which is connected to the GND pin and digital pin number 37 will be far apart so that it becomes an active buzzer trigger (7) whose cable is connected to the GND pin and digital pin number 40.

The hardware workflow diagram for the security system can be seen in Fig. 7. The process starts by loading the dataset generated from the training process. This process generates a face dataset that is stored in the application and can be used when needed. As long as the security device is operating, the initial process is to detect the presence of people around it using the PIR sensor. If someone is detected, the system starts to detect faces using the HCC algorithm.

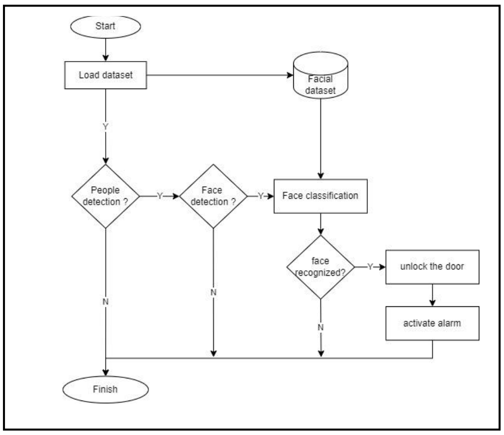


Fig. 6. Hardware workflow diagram

If a face is detected, the next process is to perform a face classification process based on a dataset that has been previously loaded. A suitable face will provide access to open the door of the room. Opening the door will activate the alarm as an indicator that the door has been opened. The alarm will not sound if the door is closed again.

III. Results and Discussion

The first experiment was to conduct data training to get a model with good accuracy from the collected facial dataset. Feature extraction on each image uses PCA to reduce the data dimensions to one dimension. Fig. 7 shows the changes in the image dataset from the feature extraction process using eigenface-based PCA. The original image which was previously in the form of three-dimensional RGB with a size of 200x200 pixels was converted to a one-dimensional grayscale with a size of 50x50 pixels. From the grayscale image, then the feature extraction stage is carried out so that the image is reduced to a shape as in the third row of Fig. 7.

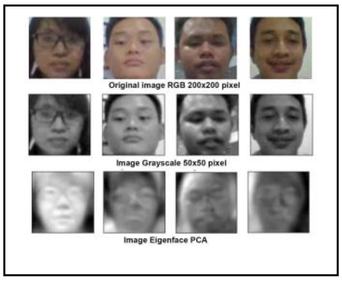


Fig. 7. original image samples and feature extraction

The extraction results divide the dataset into training data and test data used for the training process using SVM. The training process was carried out four times to find the best model with the RBF kernel, C parameter, and gamma from the SVM algorithm. The results of the data training can be seen in Fig. 8.

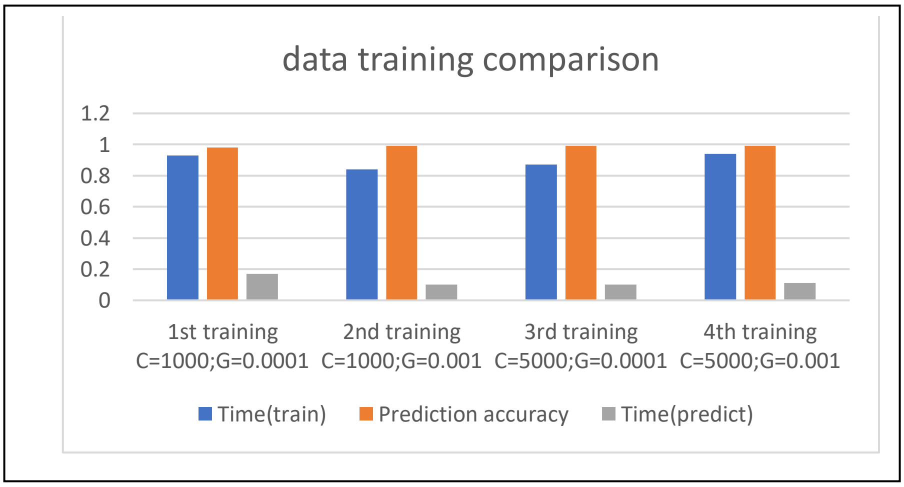
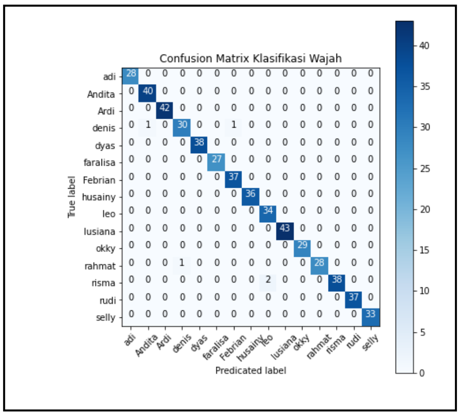


Fig. 8. Training and testing accuracy

Based on Fig. 8. The first training with settings C=1000 and gamma=0.0001 requires a training time of 0.93 seconds with a model accuracy of 98% in a test time of 0.17 seconds. the second training with C=1000 and gamma=0.001 requires a 0.84 second training time with 99% model accuracy in 0.10 second test time. the third training with C=5000 and gamma=0.0001 requires 0.87 seconds of training time with 99% model accuracy in 0.10 seconds of testing time. the fourth training with C=5000 and gamma=0.001 requires 0.94 seconds of training time with 99% model accuracy in 0.11 seconds of testing time. From the experiment, the second training resulted in the smallest time with 99% accuracy. These results show that a small C value and a larger gamma make the time shorter with high accuracy. 99% accuracy is obtained from the evaluation using equation (2), the classification that is predicted to be correct is divided by the total of all test data. The model test shows the number of correctly predicted data (TP)=520 data. More details about the results of TP predictions can be seen in the confusion matrix Fig. 9.



	adi	Andita	Ardi	denis	dyas	faralisa	Febrian	husainy	leo	lusiana	okky	rahmat	risma	rudi	selly
adi	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Andita	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0
Ardi	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0
denis	0	1	0	30	0	0	1	0	0	0	0	0	0	0	0
dyas	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0
faralisa	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0
Febrian	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0
husainy	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0
leo	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0
lusiana	0	0	0	0	0	0	0	0	0	43	0	0	0	0	0
okky	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0
rahmat	0	0	0	1	0	0	0	0	0	0	0	28	0	0	0
risma	0	0	0	0	0	0	0	0	2	0	0	0	38	0	0
rudi	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0
selly	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33

Fig. 9. Evaluation of testing data using the Confusion matrix

The true label is the actual test data, while the predicated label is the predicted result from model testing. The top left to bottom right columns that are blocked is TP data which if added up result is 520 data. So based on equation (2), then:

$$accuracy = \frac{520}{525} = 0.99\ x\ 100\% = 99\%$$

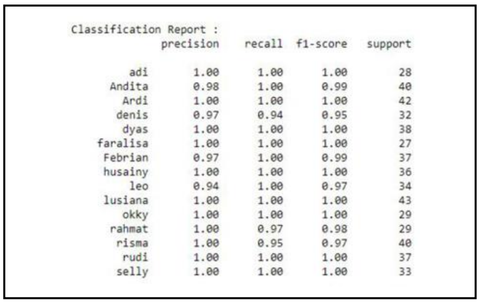
Based on Fig. 9, the evaluation also calculates precision, recall, and F1-Score using equation (3-5). For example, the confusion matrix with the name "denis" has 32 testing data. However, the prediction data is correct or TP = 30, there is data that is predicted incorrectly, namely "denis" data which is predicted "andita" and "febrian". In the evaluation method this is referred to as False Negative (FN), so FN = 2. Furthermore, there is data that is not “denis” (true label “Rahmat”) but is predicted to be “denis” which is termed False Positive, so FP=1. Evaluation of precision, recall, and F1-Score. So:

$$precission = \frac{30}{30+1} = 0.97\ x\ 100\% = 97\%$$

$$recall = \frac{30}{30+2} = 0.95\ x\ 100\% = 95\%$$

$$F1Score = 2\frac{0.97 * 0.95}{0.97 + 0.95} = 0.95\ x\ 100\% = 95\%$$

Fig. 10 shows the overall classification results where each class shows the value of precision, recall, and F1-score.



```
Classification Report :
              precision    recall  f1-score   support

         adi       1.00      1.00      1.00        28
      Andita       0.98      1.00      0.99        40
        Ardi       1.00      1.00      1.00        42
       denis       0.97      0.94      0.95        32
        dyas       1.00      1.00      1.00        38
    faralisa       1.00      1.00      1.00        27
     Febrian       0.97      1.00      0.99        37
     husainy       1.00      1.00      1.00        36
         leo       0.94      1.00      0.97        34
     lusiana       1.00      1.00      1.00        43
        okky       1.00      1.00      1.00        29
      rahmat       1.00      0.97      0.98        29
       risma       1.00      0.95      0.97        40
        rudi       1.00      1.00      1.00        37
       selly       1.00      1.00      1.00        33
```

Fig. 10. Evaluation results of precision, recall, and f1-score

After the dataset model is obtained, the next step is to implement it on a raspberry pi device. The model dataset is loaded together with the HCC algorithm to detect facial images on the camera and then classify these images with the dataset from SVM in real-time. Table 5 shows the results of the research experiments.

Table 5. Real-time Experiment Results

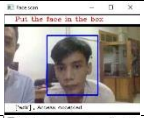
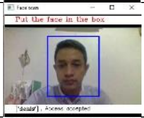
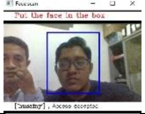
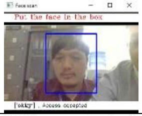
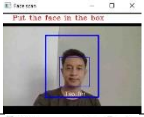
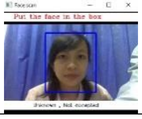
Experiment	**PIR**	**Condition**	**Description**	**Door Lock**	**Magnetic sensor**	**Alarm**
1	1	 Face scan Put the face in the box ['adi'] , Access accepted	Adi, Accepted	1	1	1
2	1	 Face scan Put the face in the box ['denis'] , Access accepted	Denis, Accepted	1	1	1
3	1	 Face scan Put the face in the box ['husainy'] , Access accepted	Husainy, Accepted	1	1	1
4	1	 Face scan Put the face in the box ['okky'] , Access accepted	Okky, Accepted	1	1	1
5	1	 Face scan Put the face in the box Too far	Too far	0	0	0
6	1	 Face scan Put the face in the box Unknown , Not accepted	Unknown, Not Accepted	0	0	0

Table 5 shows the experimental results in real-time using a raspberry pi with six randomized trials for respondents. In the first experiment, if the PIR sensor value is "0" it means that the sensor does not detect people, while the value "1" means it detects people, then the system scans the faces in the blue box. The face in the blue box indicates that the respondent named “adi” was detected and recognized as “adi” with the status “accepted” which means approved to enter. The status "accepted" makes the solenoid door lock value "1" which means the door is open, if "0" the status of the door is locked. The condition of the open door makes the magnetic sensor value "1" which means the sensor is far apart, indicating the door is open. A value of “0” on the magnetic sensor means the sensor is close together or the door is closed. The door opens and triggers the buzzer alarm with a value of “1” which means it is active, the value of “0” of the buzzer indicates that it is not active. This happened in the second, third, and fourth experiments. In the fifth experiment, the PIR sensor detects someone, but because the face is too far from the camera, it causes the solenoid door lock, magnetic sensor, and alarm to not activate. The position of the face too far is also a condition to prevent more than one face from being detected during the scanning process. In the sixth experiment, the system failed to recognize faces that should have been recognized because they were in the dataset and trained.

Several factors cause this in the next few experiments, such as the effect of lighting at the time of image capture for training data and facial scans that are too significantly different. Then the influence of image variations is not too much so that the resulting model is less accurate. At the time of the scan, the face should also be visible in both eyes, nose, and ears. Positions that are too tilted to the left or right cause the system cannot detect faces.

IV. Conclusion

The SVM machine learning algorithm was successfully implemented on a raspberry pi 4b-based recognition device with increased performance based on model testing that was trained with 99% accuracy with a training time of 0.10 seconds. SVM is suitable to be applied to mini-computers such as raspberry pi because this algorithm is efficient in image classification, fast in the training process, and real-time processing so that it saves resources such as memory. The dataset model implemented on the raspberry pi device can recognize faces in real-time, but there are still errors in recognizing faces. this is because the variation of sample data during training is less varied and the effect of lighting during the face image recognition process is in real-time.

Based on the results, it is hoped that further research can develop a system by increasing accuracy during real-time classification with data samples that can be more varied with various lighting conditions from each data class. Then the system can be developed by authenticating and distinguishing in real-time the real face on the camera from the face from the digital image.

References

[1] A. Roihan, P. A. Sunarya, and A. S. Rafika, “Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper,” *Indones. J. Comput. Inf. Technol.*, vol. 5, no. 1, pp. 75–82, 2019.

[2] S. Yulina, “Penerapan Haar Cascade Classifier dalam Mendeteksi Wajah dan Transformasi Citra Grayscale Menggunakan OpenCV,” *J. Politek. Caltex Riau*, vol. 7, no. 1, pp. 100–109, 2021.

[3] V. D. Win, “Pengenalan Wajah Menggunakan Convolutional Neural Network,” Institut Teknologi Sepuluh Nopember, 2018.

[4] S. Subiyanto, D. Priliyana, M. E. Riyadani, N. Iksan, and H. Wibawanto, “Sistem Pengenalan Wajah dengan Algoritme PCA-GA untuk Keamanan Pintu Rumah Pintar Menggunakan Rasberry Pi,” *J. Teknol. dan Sist. Komput.*, vol. 8, no. 3, pp. 210–216, 2020, doi: 10.14710/jtsiskom.2020.13590.

[5] H. Zhi and S. Liu, “Face Recognition Based on Genetic Algorithm,” *J. Vis. Commun. Image Represent.*, vol. 58, pp. 495–502, 2019, doi: 10.1016/j.jvcir.2018.12.012.

[6] A. Nurhopipah and A. Harjoko, “Motion Detection and Face Recognition for CCTV Surveillance System,” *Indones. J. Comput. Cybern. Syst.*, vol. 12, no. 2, pp. 107–118, 2018, doi: 10.22146/ijccs.18198.

[7] E. P. Purwandari, A. Erlansari, A. Wijanarko, and E. A. Adrian, “Pengenalan Sketsa Wajah Menggunakan Principal Component Analysis sebagai Aplikasi Forensik,” *J. Teknol. dan Sist. Komput.*, vol. 8, no. 3, pp. 178–184, 2020, doi: 10.14710/jtsiskom.2020.13422.

[8] S. Al-Aidid and D. S. Pamungkas, “Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram,” *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 62–67, 2018, doi: 10.17529/jre.v14i1.9799.

[9] I. D. Wijaya, N. Usman, and M. A. Barata, “Implementasi Raspberry Pi untuk Rancang Bangun Sistem Keamanan Pintu Ruangan Server dengan Pengenalan Wajah Menggunakan Metode Triangle Face,” *J. Inform. Polinema*, vol. 4, no. 1, pp. 9–15, 2017.

[10] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, “Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 6308–6325, 2020, doi: 10.1109/JSTARS.2020.3026724.

[11] S. Abidin, “Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam pada Matlab,” *J. Teknol. Elekterika*, vol. 15, no. 1, pp. 21–27, 2018, doi: 10.31963/elekterika.v15i1.2102.

[12] R. Purwati and G. Ariyanto, “Pengenalan Wajah Manusia Berbasis Algoritma Local Binary Pattern,” *Emit. J. Tek. Elektro*, vol. 17, no. 2, pp. 29–38, 2017, doi: 10.23917/emitor.v17i2.6232.

[13] Fahrizal, F. O. Reynaldi, and N. Hikmah, “Implementasi Machine Learning pada Sistem PETS Identification Menggunakan Python Berbasis Ubuntu,” *J. Inf. Syst. Informatics Comput.*, vol. 4, no. 1, pp. 86–91, 2020.

[14] I. I. Setiawan, A. Jaenul, and D. Priyokusumo, “Prototipe Sistem Keamanan Rumah Menggunakan Face Recognition Berbasis Raspberry Pi 4,” *SNITT- Politek. Negeri Balikpapan 2020*, pp. 496–501, 2020.

[15] R. R. Pratama, “Analisis Model Machine Learning Terhadap Pengenalan Aktifitas Manusia,” *J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 2, pp. 302–311, 2020, doi: 10.30812/matrik.v19i2.688.

[16] J. W. G. Putra, “Pengenalan Konsep Pembelajaran Mesin dan Deep Learning,” in *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, 1.4., Tokyo, 2020, pp. 1–235.

[17] R. Santoso, Ramadhandi Resky Megasari and Y. A. Hambali, “Implementasi Metode Machine Learning Menggunakan Algoritma Evolving Artificial Neural Network pada Kasus Prediksi Diagnosis Diabetes,” *J. Apl. dan Teor. Ilmu Komput.*, vol. 3, no. 2, 2020, [Online]. Available: https://ejournal.upi.edu/index.php/JATIKOM/article/view/27885.

[18] I. M. Parapat, M. T. Furqon, and Sutrisno, "Penerapan Metode Support Vector Machine (SVM) pada Klasifikasi Penyimpangan Tumbuh Kembang Anak," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, pp. 3163–3169, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2577.

[19] S. Prangga, "Optimasi Parameter pada Support Vector Machine Menggunakan Pendekatan Metode Taguchi untuk Data High-Dimensional," Institut Teknologi Sepuluh November, 2017.

[20] T. Y. Hadiwandra and F. Candra, "High Availability Server Using Raspberry Pi 4 Cluster and Docker Swarm," *IT J. Res. Dev.*, vol. 6, no. 1, pp. 43–51, 2021, [Online]. Available: https://journal.uir.ac.id/index.php/ITJRD/article/view/5806.

[21] Friendly, Z. Sembiring, and H. R. Safitri, "Deteksi Wajah Bermasker Berbasis Tensorflow-Keras untuk Pengendalian Gerbang Akses Masuk Menggunakan Rasberry Pi4," *Jikstra*, vol. 02, no. 02, pp. 45–55, 2020.

[22] R. Suwartika and G. Sembada, "Perancangan Sistem Keamanan Menggunakan Solenoid Door Lock Berbasis Arduino Uno pada Pintu Laboratorium di PT. XYZ," *J. E-Komtek*, vol. 4, no. 1, pp. 62–74, 2020, doi: 10.37339/e-komtek.v4i1.217.

[23] R. Toyib, I. Bustami, D. Abdullah, and O. Onsardi, "Penggunaan Sensor Passive Infrared Receiver (PIR) untuk Mendeteksi Gerak Berbasis Short Message Service Gateway," *J. Pseudocode*, vol. 6, no. 2, pp. 114–124, 2019, doi: 10.33369/pseudocode.6.2.114-124.

[24] Siswanto, G. P. Utama, and W. Gata, "Pengamanan Ruangan dengan Dfrduino Uno R3, Sensor Mc-38, Pir, Notifikasi Sms, Twitter," *J. Resti (Rekayasa Sist. dan Teknol. Informasi)*, vol. 2, no. 3, pp. 697–707, 2018, doi: 10.29207/resti.v2i3.592.

[25] A. Yudhana, S. Sunardi, and P. Priyatno, "Perancangan Pengaman Pintu Rumah Berbasis Sidik Jari Menggunakan Metode UML," *J. Teknol.*, vol. 10, no. 2, pp. 131–138, 2018.

[26] A. B. Satriya and S. Agustini, "Face Recognition using Modified Triangle Method," *Integer J.*, vol. 2, no. 1, pp. 1–9, 2017.

[27] D. Devito, R. C. Wihandika, and A. W. Widodo, "Ekstraksi Ciri untuk Klasifikasi Gender Berbasis Citra Wajah Menggunakan Metode Histogram of Oriented Gradients," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 8, pp. 8002–8011, 2019.

[28] W. Fitriani, M. Zidny Naf'an, and E. Usada, "Ekstraksi Fitur pada Citra Tanda Tangan sebagai Ciri Identitias Pemiliknya Menggunakan Discrete Fourier Transform," in *Prosiding Sendi_u*, 2018, pp. 978–979.

[29] H. Ai and X. Cheng, "Research on Embedded Access Control Security System and Face Recognition System," *Meas. J. Int. Meas. Confed.*, vol. 123, no. April, pp. 309–322, 2018, doi: 10.1016/j.measurement.2018.04.005.

[30] N. Kustian, "Analisis Komponen Utama Menggunakan Metode Eigenface Terhadap Pengenalan Citra Wajah," *J. Teknol.*, vol. 9, no. 1, p. 43, 2017, doi: 10.24853/jurtek.9.1.43-48.

[31] R. Armandhani, R. C. Wihandika, and M. A. Rahman, "Klasifikasi Gender Berbasis Wajah Menggunakan Metode Local Binary Pattern dan Random KNN," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 8, pp. 7575–7582, 2019.

[32] R. Munarto and A. Darma, "Klasifikasi Gender dan Usia Berdasarkan Citra Wajah Manusia Menggunakan Convolutional Neural Network," *Setrum Sist. Kendali Tenaga Elektron. Telekomun. Komput.*, vol. 10, no. 2, pp. 30–43, 2021, doi: 10.36055/setrum.v10i2.12991.

[33] C. Kurniawan and H. Irsyad, "Perbandingan Metode K-Nearest Neighbor Dan Naïve Bayes untuk Klasifikasi Gender Berdasarkan Mata," *J. Algoritm.*, vol. 2, no. 2, pp. 82–91, 2022.