I.J. Image, Graphics and Signal Processing, 2019, 11, 1-9
Published Online November 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2019.11.01

Real-Time Face Recognition with Eigenface Method

Ni Kadek Ayu Wirdiani
Departement of Information Technology, Udayana University, Bali, Indonesia
Email: ayuwirdiani@unud.ac.id

Tita Lattifia, I Kadek Supadma, Boy Jehezekiel Kemanang Mahar, and Dewa Ayu Nadia Taradhita
Departement of Information Technology, Udayana University, Bali, Indonesia
Email: {titalattifia26, kadeksupadma91, boyjehezekiel, taradhita}@gmail.com

Adi Fahmi
Departement of Information Technology, Udayana University, Bali, Indonesia
Email: adifahmi@student.unud.ac.id

Received: 15 June 2019; Accepted: 25 September 2019; Published: 08 November 2019

***Abstract*—**Real-time face image recognition is a face recognition system that is done directly using a webcam camera from a computer. Face recognition system aims to implement a biometrics system as a real-time facial recognition system. This system is divided into two important processes, namely the training process and the identification process. The registration process is a process where a user registered their name in a system and then registers their face. Face data that has been registered will be used for the next process, namely the identification process. The face registration process uses face detection using the OpenCV library. The feature extraction process and introduction to the recognition system use the Eigenface method. The results of this study found that, the Eigenface method is able to detect faces accurately up to 4 people simultaneously. The greater the threshold value will result in a greater value of FRR, while there isn't any FAR value found from different thresholds. The level of lighting, poses, and facial distance from the camera when training and testing the face image heavily influences the use of the eigenface method.

***Index Terms*—**Biometrics, real-time recognition, face recognition, eigenface method, Euclidean distance.

I. Introduction

The rapid development of technology influences the need for self-recognition systems. One of the rapidly developing self-recognition technologies is face recognition. The researchers were motivated to develop a face recognition system because of its wide application opportunities in the commercial and government fields. Face recognition technology can be applied in various fields, such as surveillance systems, crime recognition systems, and authentication systems.

Several studies in the field of facial recognition have been conducted. Most studies focused on detecting facial features such as the eyes, nose and mouth. In addition, most also defined facial images based on size, position, and the relationship between these facial features. Most of these studies had difficulty in recognizing faces in various poses, even though they are easy to identify.

Eigenface method has so far succeeded in providing high accuracy and efficiency for facial recognition. Eigenface is a set of eigenvectors that represent the characteristic space of face images in the database. Eigenface will project a face image vector from a higher dimensional space to a lower dimensional space. The level of similarity between the input face image and the face image in the similarity function database is calculated after the face image is projected using Eigenface.

Various studies on the Eigenface method have been conducted to design a face recognition system based on Android smartphones [1,2]. Face recognition applications that have been designed using Eigenface are able to recognize faces with accuracy up to 94.4% [1]. The similarity function that can be used in the Eigenface method is Euclidean distance [3]. The smaller the Euclidean distance between the two face image vectors, the greater the level of similarity. The vector of a face image that has the smallest Euclidean distance with an input face image vector will be the output of the face recognition process.

In this paper, we proposed a real-time face recognition system using Eigenface method. This paper is organized into several sections. In Section II, we discussed related works in real-time face recognition methods. Section III described the methodology of proposed real-time face recognition system. Section IV explained about the image acquisition steps taken in the proposed system. Section V described the processes taken in the preprocessing step. Section VI explained the calculations in each stages of

Eigenface method. The experiment process and results involving various amount of faces detected, threshold, and the distance from face to camera are explained thoroughly in Section VII. Lastly, Section VII discusses the conclusion from this paper.

II. Related Works

Many studies related to real time face recognition have been conducted. The real time GUI-based face recognition system is designed with a tool called Open Face, which is based on OpenCV. The HOG (Histogram of Oriented Gradients) method is used to detect the front part of the face image and reduce facial dimensionality. 128 facial features were extracted using the Deep Neural Network algorithm before the classification process with the SVM method was applied [4].

Zhang, Gonnot, and Saniie combined AdaBoost algorithms, Local Binary Pattern (LBP), and Principal Component Analysis (PCA) to design a real time face recognition system. AdaBoost is used in eye and face detection training. LBP is used in feature extraction. PCA is used to recognize faces efficiently. True positive results from the combination of algorithms reached 98.8% for face detection and 99.2% for proper facial recognition [5].

PCA as a feature extraction method was combined with Back Propagation Neural Network (BPNN) and Radial Basis Function (RBF) for classification process [6]. The face detection and cropping process from image input were implemented using Viola-Jones algorithm. The use of BPNN and RBF method together produced 90% acceptance ratio.

Zhang, Wen, Shi, Lei, Lyu, and Li proposed a CNN-based Single-Shot Scale-Aware Face Detection (SFDet) model. This model uses large-scale layers that are related to the number of different anchors to handle various face sizes. Scale compensation operations are used to improve recall. Every loss function from the training sample is added IoU-aware weight to increase accuracy. The SFDet method is able to recognize faces in real-time [7].

Sayeed, Hossen, Jayakumar, Yusof, and Samraj designed a student attendance system using the PCA method for face recognition in real time. The matching process is done using Euclidean distance [8].

Mahdi, et al. designed a real-time face- based surveillance system. The methods used include Viola-Jones for face detection, Kanade-Lucas-Tomasi algorithm for tracking features and PCA for face recognition. The main factors that influence face recognition by systems are lighting [9].

Widiakumara, Putra, and Wibawa designed an Android-based application with the Eigenface method to obtain the weight value of face images. The results of image identification are seen from the smallest difference in Eigenface weight of the test image and database image. The success rate of the application testing is 68% with a FAR value of 32% [2].

Oka Sudana, Darma Putra, and Arismandika tested the Eigenface method for face images taken directly with the Android smartphone camera. The face image is converted to grayscale then the eigenvalue and threshold values are calculated. The difference in eigenvalue of the test image with the image in the database is compared to the threshold value. The highest accuracy is achieved by a threshold value of 44, which is 94.48%, with an FMR value of 2.52% and a FNMR of 3% [1].

Astawa, Putra, Sudarma and Hartanti compared the performance of YIQ, YCbCr, HSV, HSL, CIELAB, CIELUV, and RGB color space images in face detection. Intensity normalization is added to one channel of color spaces before being converted back to RGB. The percentage of face detection in each color space increases after intensity normalization [10].

The transform-invariant PCA (TIPCA) method is used to identify face characters by aligning the image and making optimal eigenspace to minimize the mean square error [11]. The face images processed by TIPCA are then tested by LBP, HOG, and GEF methods.

De, Saha, and Pal use the Eigenface-based method to recognize facial expressions (sad, happy, angry, scared, and surprised). This method uses the HSV color space in the face detection process before being converted to grayscale and applied the PCA process. Happy facial expressions were recognized by 93.1%, making them the most recognizable facial expression [3].

Kas, El Merabet, Ruichek, and Messoussi proposed a face description method called Mixed Neighborhood Topology Cross Decoded Patterns (MNTCDP). MNTCBDP is an LBP-like method that combines the neighborhood topology and pattern encoding as a feature extraction method [12]

ICA-based facial recognition method combines two different architectures with novel classification schemes that use mode values [13]. This method uses various types of preprocessing, which are Mirror Image Superposition (MIS), Histogram Equalization (HE) and Gaussian Filtering (GF) to overcome poses, expressions, and lighting variations. MIS is used to neutralize variations in expressions and poses, HE is used to increase contrast and GF helps eliminate noise. The accuracy recognition results has increased by around 14% compared to the original ICA method.

Huang and Yin introduced binary gradient patterns (BGP) methods to represent facial features. Image gradients are calculated from various directions and inserted into a series of binary strings to get the local structure on the pattern gradient [14]

Roy and Bhattacharjee used the Local Gravity Face method for invariant illumination and heterogeneous face recognition (HFR). LG-face uses Local Gravitational Force Angle (LGFA), which is the direction of gravitational force that is given by the middle pixel on other pixels in the local environment. This method has consistent performance in testing towards variations in noise [15].

III. Methodology

This system consists of two types of processes. The first process is the training process, which is used to store the weight value of training results (which were extracted from face images) into the database. The second process is the recognition process, which covers real-time image retrieval and comparing the weight values of the test image with the weight values of images stored in the database. The system main overview is illustrated in Fig. 1.

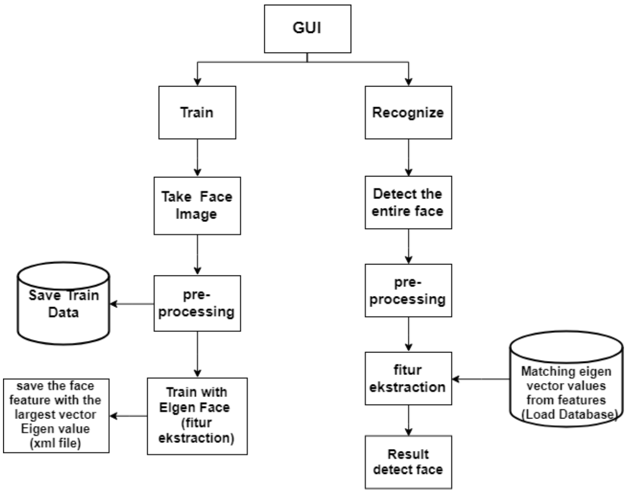
GUI
Train
Recognize
Take Face Image
Detect the entire face
Save Train Data
pre-processing
pre-processing
save the face feature with the largest vector Eigen value (xml file)
Train with Eigen Face (fitur ekstraction)
fitur ekstraction
Matching eigen vector values from features (Load Database)
Result detect face

Fig. 1. System main overview.

A. Training Process

The training process or storage of feature weights into the database consists of several steps. The first step in the training process is entering the name into the field available in the GUI then selecting Train menu. The face image will be taken by the camera 100 times and stored in the database. Face detection process is done before data is saved into the database. When the face is detected, then cropping process will be done to the face image. Cropped images will be converted to grayscale, and then histogram equalization is performed to grayscale images. The image from the histogram equalization is converted into a flat vector or one-dimensional array. The next step is eigenface extraction. This process is done to get the weight value or eigen weight of the image, and after all the processes are done, the data is stored into one between the identity and the value of the weight that has been obtained.

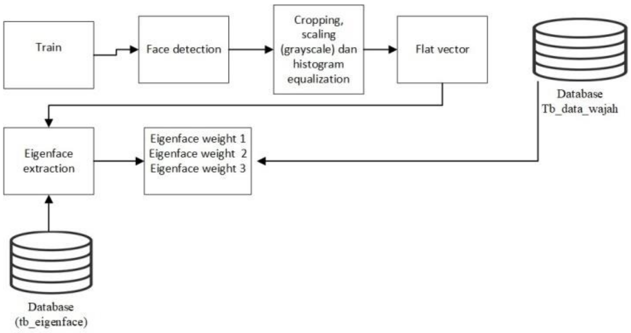

Fig. 2. Training process overview.

B. Recognition Process

Recognition process is done through several stages before getting the recognition results. The first stage in the recognition process is by selecting Testing menu while the face that will be detected is in front of the camera. The camera will take the images of the face to be used as input for face detection process. When the face is detected, preprocessing will be applied to the generated image and the image is converted to a flat vector. The next process is applying eigen extraction in the face image. This process is done to obtain the weight value of the image. The weight value of the image is used to compare the images stored in the database with the recognized image. The recognized image data is image data that has the minimum number of difference in error value.

IV. Image Acquisition

Image acquisition is a process to acquire an image that will be used for the recognition process. The image used is a face image. Image acquisition is done with utilizing a webcam camera directly from a computer. The webcam camera display will be activated when the user selects the Training or Testing menu.

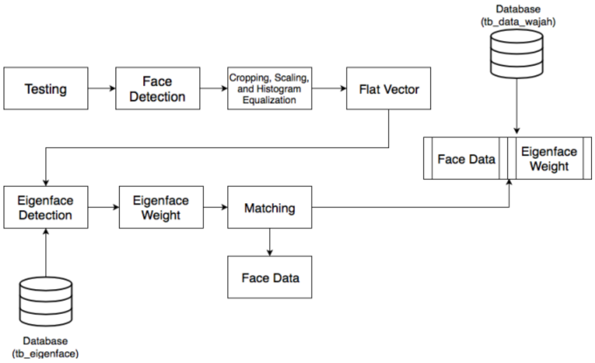

Fig. 3. Recognition process overview.

Image acquisition process utilized VideoCapture() feature found in the OpenCV library. The function will capture the face object from the camera into a RGB image. The application will detect faces on images displayed using the Haar Cascade Classifier method. This method is a cascade classifier method that is trained with the Haar feature to detect an object [5]. The Haar feature is shaped like a rectangle where each feature is a single value obtained by subtracting the number of pixels below the white box from the number of pixels below the black box [9].

The features of the image are grouped at various stages of the classifier and applied one by one with different sizes [5]. Each stage of the classifier is used to determine whether the image part includes the face or not. If part of the image fails to be detected as the face, then the certain part is immediately ignored.

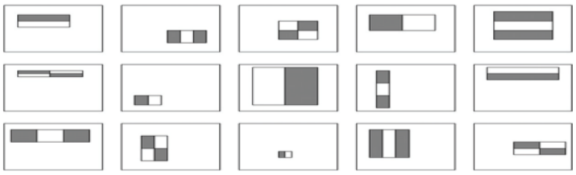

Fig. 4. Haar feature.

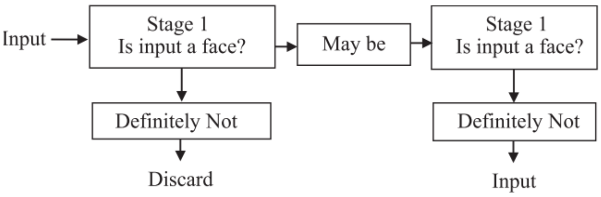

Fig. 5. Cascade classifier algorithm.

V. Preprocessing

A. Cropping and Scaling

The cropping process is the process of cutting off the acquired facial image into a specified size. Cropped image then undergo scaling process, which is the process of converting RGB images into greyscale images.

B. Histogram Normalization

Histogram Equalization is a technique to increase contrast in image processing [13]. Histogram Equalization flattens the intensity of pixels by non-linear mapping on the input image such that the Histogram Equalization result has a uniform intensity distribution.

VI. Eigenface

The word eigenface comes from the German language "eigenwert" where "eigen" means characteristics and "wert" means value. Eigenface is a face pattern recognition algorithm based on the Principle Component Analysis (PCA) [3]. The eigenface method extracts relevant information from a face image then converts it into a face code set called the eigenvector. This face code is then compared to the face database which stores face codes from previous training [2]. Eigenvector is also expressed as facial characteristics, therefore this method is called eigenface. Each face is represented in a linear eigenface combination.

The Eigenface calculation consists of two stages, namely the training stage and the face recognition stage. The training stage algorithms at Eigenface are as follows [2].

1. Preparing data with making a set *S* consists of training image Γ1, Γ2, ...,Γ*M*.
2.

$$S = \Gamma 1, \Gamma 2, \ldots, \Gamma M \qquad (1)$$

3. Calculating the average or mean value (Ψ). Γ in Equation 2 refers to training image. *M* refers to total amount of training image.

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma \qquad (2)$$

4. Calculating the difference (Φ) between training image value (Γi) with mean value (Ψ).

$$\phi i = \Gamma_i - \Psi . \qquad (3)$$

5. Calculating the value of the covariance matrix (*C*),

$$C = \frac{1}{M}\sum_{n=1}^{M}\phi^i \phi^n = AA^T \qquad (4)$$

$$L = A^T A = \phi_m^T \phi^n \qquad (5)$$

A is the matrix that consists of differences between each training image withmean value.

$$A = \{\phi 1, \phi 2, \phi 3, \ldots, \phi n\} . \qquad (6)$$

6. Calculating the eigenvalue (λ) and eigenvector (*v*) from the covariance matrix (*C*).

$$C \times vi = \lambda i \times vi . \qquad (7)$$

7. After the eigenvector (*v*) is obtained, the eigenface (μ) can be calculated by Equation 8.

$$\mu_i = \sum_{k=1}^{M} v_{ik}\phi_k \qquad (8)$$

The face recognition stage algorithm on Eigenface is as follows [2].

1. After the eigenvector *(v)* is obtained, then the eigenface (μ) can be found with Equation 9. Γ_{new} is the testing image and Ψ is the average value.

$$\mu_{new} = v \times (\Gamma_{new} - \Psi) \qquad (9)$$

2. Use euclidean distance method to find the shortest distance between the eigenvalue of the training image in the database and the eigenvalue of the testing image. *ε* is the Euclidean distance, *Ω* is the training image eigenvalue, and *Ω*$_k$ is the testing image eigenvalue.

$$\varepsilon_k = \left\| \Omega - \Omega_k \right\| \qquad (10)$$

VII. Experiments and Result

The real time facial recognition system with Eigenface method was created with the OpenCV 3 library and the Python 3.7 programming language. The interface display of the face recognition application system in real time using the Eigenface method is showed in Fig. 6.

Fig. 6 is the interface display of the face recognition system, where there is a Training menu to train image, Preprocessing menu to filter photos and Recognize menu for the testing process.

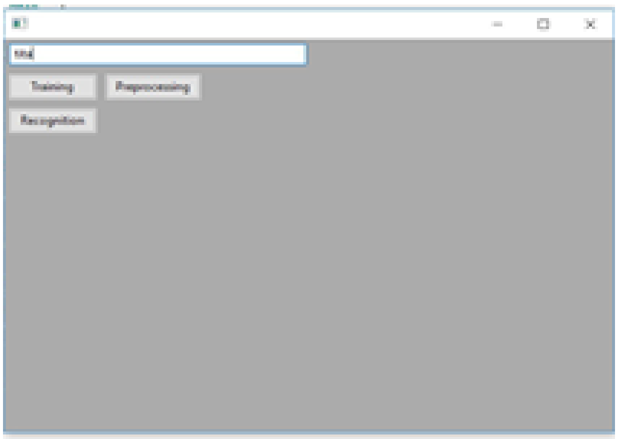

Fig. 6. Cascade classifier algorithm.

We tested the face recognition system by using the webcam to take real-time screenshots of faces that will be recognized. The testing of real time face image recognition system is based on the number of faces identified, the number of thresholds, and the distance between face and the camera.

A. *Testing Based on The Number of Faces*

The testing of real time face image recognition system is performed on one face, two faces, three faces, and four faces.

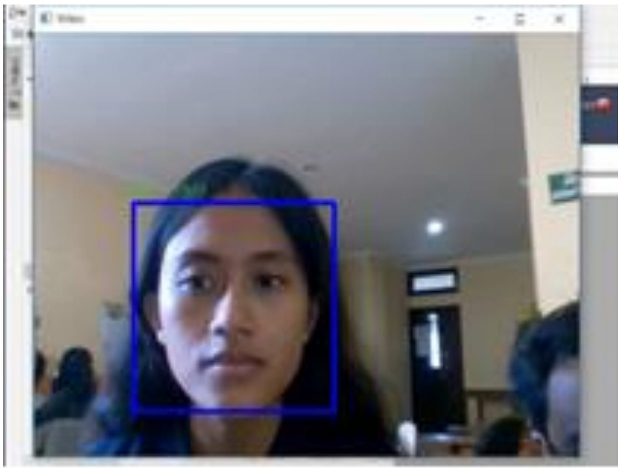

Fig. 7. One face testing.

Fig. 7. is an example of testing result using one face, where the system successfully recognizes the face as Tita.

From the results of the identification test, eight results were identified as identified objects and no results failed to recognize the object. The results above shown the success rate of identification was 100% from all 8 different faces tested and the rate of misidentification was 0% of the 8 times the total testing of 8 different faces.

Table 1. Results of One Face Identification

PERSON	RECOGNIZED AS	RECOGNIZED OR NOT RECOGNIZED
Tita Lattifia	Tita Lattifia	Recognized
Adi Fahmi	Adi Fahmi	Recognized
Boy Jehezkiel	Boy	Recognized
Tara Ditha	Tara	Recognized
Kadek supadma	Supadma	Recognized
Ryan Adi Wiguna	Ryan	Recognized
Bagus Miftah	Miftah	Recognized
Aldi pratama	Aldi pratama	Recognized

Table 2. Results of Two Faces Identification

PERSON	RECOGNIZED AS	RECOGNIZED OR NOT RECOGNIZED
Tita Lattifia and Dania Evandari	Tita Lattifia and Dania Evandari	Recognized
Miftah and Ryan	Miftah and Ryan	Recognized
Adi and Ryan	Miftah and Ryan	Recognized

Fig. 8. is an example of testing result using two faces, where the system successfully recognizes the faces as Miftah dan Ryan.

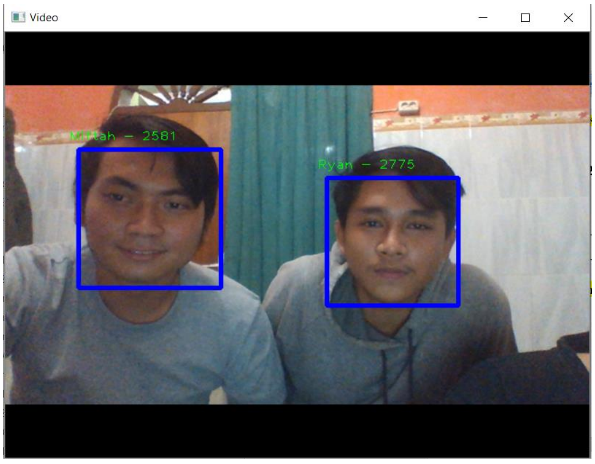

Fig. 8. Two faces testing.

The results of the tests on two faces resulted in three identifications as recognized objects and no results that failed to recognize the object. The identification success rate is 100% and the recognition error rate is 0% from 3 times the total trial to 5 different faces, with the identification of 2 face experiments at the same time in each experiment.

Fig. 9. is an example of testing result using three faces, where the system successfully recognizes the faces as Ryan, Adi and Miftah.

Table 3. Results of Three Faces Identification

PERSON	RECOGNIZED AS	RECOGNIZED OR NOT RECOGNIZED
Tita, Dania, Alesia	Tita, Dania, Alesia	Recognized
Adi,Ryan, Boy	Adi, Ryan, Boy	Recognized
Miftah, Adi, Ryan	Miftah, Adi, Ryan	Recognized

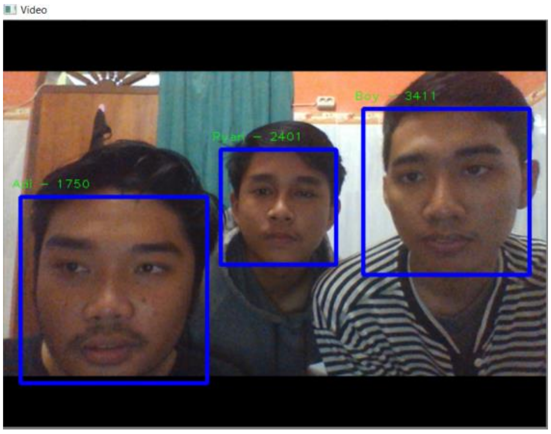

Fig. 9. Three faces testing.

Table 4. Results of Four Faces Identification

PERSON	RECOGNIZED AS	RECOGNIZED OR NOT RECOGNIZED
Tita, Dania, Alesia, Dode	Tita, Dania, Alesia, Dode	Recognized
Adi, Ryan, Boy, Miftah	Adi, Ryan, Boy, Miftah	Recognized

The results of the tests on three faces resulted in no results that failed to recognize the object. The identification success rate is 100% and the recognition error rate is 0% from 3 times the total trial to 7 different faces, with the identification of 3 face experiments at the same time in each experiment.

Fig. 10. is an example of testing result using four faces, where the system successfully recognizes the faces as Adi, Ryan, Miftah dan Boy.

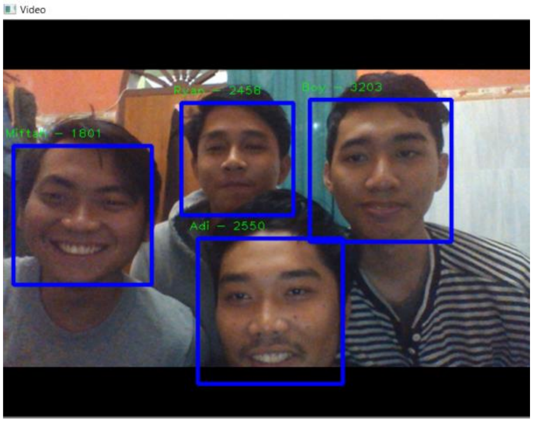

Fig. 10. Four faces testing.

The results of the tests on four faces resulted in no results that failed to recognize the object. The identification success rate is 100% and the recognition error rate is 0% from 2 times the total trial to 8 different faces, with the identification of 4 face experiments at the same time in each experiment.

B. Testing Based on Threshold

Threshold used in testing of real time face recognition system are 3500, 2500, 4000, and 1000.

The percentage of success of identification with a threshold value of 3500 is 100% and the percentage of failure of the identification process is 0% from 9 faces that were tested.

Table 5. Results of Identification in Threshold 3500

NAME	RECOGNIZED	
Tita Lattifia	Yes	
Adi fahmi	Yes	
Boy jehezkiel	Yes	
Supadma	Yes	
Tara	Yes	
Ryan	Yes	
miftah	Yes	
Putra	Yes	
Aldi	Yes	
FAR	**0**	
FRR	**0**	

Table 6. Results of Identification in Threshold 2500

NAME	RECOGNIZED	
Tita Lattifia	Yes	
Adi fahmi	Yes	
Boy jehezkiel		No
Supadma	Yes	
Tara	Yes	
Ryan	Yes	
Miftah	Yes	
Putra	Yes	
Aldi	Yes	
FAR	**0**	
FRR	**0.1111**	

The percentage of success of identification with a threshold value of 2500 is 88.8% and the percentage of failure of the identification process is 11.2%. The failure that occurred was due to an error in rejection of the face object named Boy Jehezkiel which was actually in the class but was rejected (not recognized).

Table 7. Results of Identification in Threshold 2000

NAME	RECOGNIZED	
Tita Lattifia	Yes	
Adi fahmi	Yes	
Boy jehezkiel		No
Supadma		No
Tara	Yes	
Ryan		No
Miftah	Yes	
Putra	Yes	
Aldi	Yes	No
FAR	**0**	
FRR	**0.444**	

The percentage of success of identification with a threshold value of 2000 is 88.8% and the percentage of failure of the identification process is 11.2%.

Table 8. Results of Identification in Threshold 4000

NAME	RECOGNIZED	
Tita Lattifia	Yes	
Adi fahmi	Yes	
Boy jehezkiel		No
Supadma	Yes	
Tara	Yes	
Ryan	Yes	
Miftah	Yes	
Putra	Yes	
Aldi	Yes	
FAR	**0**	
FRR	**0.1111**	

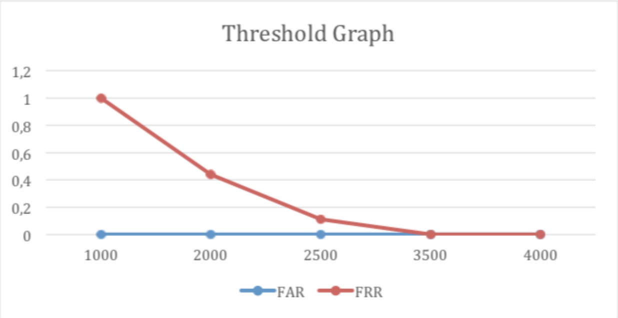

Fig. 11. Threshold Graph.

The percentage of success of identification with a threshold value of 4000 is 88.8% and the percentage of failure of the identification process is 11.2%.

The percentage of success of identification with a threshold value of 1000 is 88.8% and the percentage of failure of the identification process is 11.2%.

Table 9. Results of Identification in Threshold 1000

NAME	RECOGNIZED	
Tita Lattifia	Yes	
Adi fahmi	Yes	
Boy jehezkiel		No
Supadma	Yes	
Tara	Yes	
Ryan	Yes	
Miftah	Yes	
Putra	Yes	
Aldi	Yes	
FAR	**0**	
FRR	**0.1111**	

Based on the threshold test, the smaller the threshold value results in the increase of FRR (False Rejection Rate). The FAR (False Acceptance Rate) values were not found in this experiment.

C. Testing Based on Face to Camera Distance

Table 10 explains the results of testing process with the distance of the camera to the face, where the application can properly recognize the face with a distance of up to 100 cm, the application cannot properly recognize faces with distances above 100cm. This is caused by the inability of the camera to take clear images of the facial features if the subject is far from the camera.

Table 10. Results of Identification Based on Face to Camera Distance

DISTANCE	RECOGNIZED	
30cm	Yes	
40cm	Yes	
50cm	Yes	
70cm	Yes	
80cm	Yes	
90cm	Yes	
100cm	Yes	
130 cm		No
150 cm		No
FAR	**0**	
FRR	**0.1111**	

D. Face Training Time

We measured the amount of time taken during training process. The training process is shown in Fig. 12.

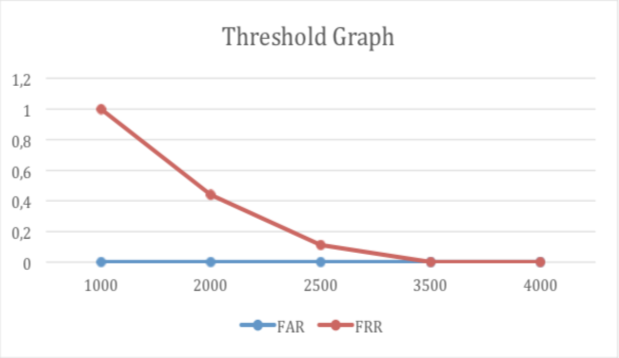

Fig. 12. Calculating Time of Face Training

Fig. 12. is an example of face data training result. Application could train the face well with the average time of 3-4 minutes needed.

VIII. Conclusion

The results of this study found that Eigenface method can be applied and used to recognize up until four person's faces directly from a live camera. The lighting that is done when training and testing the image heavily influences the use of the eigenface method. Different lighting levels, facial position, and facial distance with the camera can result in reduced accuracy of recognition. Future works could investigate other methods to overcome reduction of recognition accuracy caused by the surrounding conditions during image acquisition.

References

[1] A. A. K. Oka Sudana, I. K. G. Darma Putra, and A. Arismandika, “Face Recognition System on Android Using Eigenface Method,” *Journal of Theoretical and Applied Information Technology*, vol. 61, no. 1, pp. 128–135, 2014.

[2] I. K. S. Widiakumara, I. K. G. D. Putra, and K. S. Wibawa, “Aplikasi Identifikasi Wajah Berbasis Android,” *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 8, no. 3, pp. 200, 2018.

[3] A. De, A. Saha, and M. C. Pal, “A human facial expression recognition model based on eigen face approach,” *Procedia Computer Science*, vol. 45, no. C, pp. 282–289, 2015.

[4] D. M. Prasanna and C. G. Reddy, “Development of Real Time Face Recognition System Using OpenCV,” *International Research Journal of Engineering and Technology*, vol. 4, no. 12, 2017.

[5] X. Zhang, T. Gonnot, and J. Saniie, “Real-Time Face Detection and Recognition in Complex Background,” *Journal of Signal and Information Processing*, vol. 08, no. 02, pp. 99–112, 2017.

[6] N. H. Barnouti, S. S. M. Al-Dabbagh, and M. H. J. Al-Bamarni, “Real-Time Face Detection And Recognition Using Principal Component Analysis (PCA) – Back Propagation Neural Network (BPNN) And Radial Basis Function (RBF),” *Journal of Theoretical and Applied Information Technology*, vol. 91, no. 1, pp. 28–34, 2016.

[7] S. Zhang, L. Wen, H. Shi, Z. Lei, S. Lyu, and S. Z. Li, “Single-Shot Scale-Aware Network for Real-Time Face Detection,” *International Journal of Computer Vision*, 2019.

[8] S. Sayeed, J. Hossen, V. Jayakumar, I. Yusof, and A. Samraj, “Real-Time Face Recognition for Attendance,” *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 1, 2017.

[9] F. P. Mahdi, M. M. Habib, M. A. R. Ahad, S. McKeever, A. S. M. Moslehuddin, and P. Vasant, “Face recognition-based real-time system for surveillance,” *Intelligent Decision Technologies*, vol. 11, no. 1, pp. 79–92, 2017.

[10] I. N. G. A. Astawa, I. K. G. D. Putra, I. M. Sudarma, and R. S. Hartati, “The Impact of Color Space and Intensity Normalization to Face Detection Performance,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 15, no. 4, pp. 1894, 2018.

[11] W. Deng, J. Hu, J. Lu, and J. Guo, “Transform-invariant PCA: A unified approach to fully automatic face alignment, representation, and recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1275–1284, 2014.

[12] M. Kas, Y. El merabet, Y. Ruichek, and R. Messoussi, “Mixed neighborhood topology cross decoded patterns for image-based face recognition,” *Expert Systems with Applications*, vol. 114, pp. 119–142, 2018.

[13] V. R. Peddigari, P. Srinivasa, and R. Kumar, “Enhanced ICA based Face Recognition using Histogram Equalization and Mirror Image Superposition,” *2015 IEEE International Conference on Consumer Electronics, ICCE 2015*, pp. 625–628, 2015.

[14] W. Huang and H. Yin, “Robust face recognition with structural binary gradient patterns,” *Pattern Recognition*, vol. 68, pp. 126–140, 2017.

[15] H. Roy and D. Bhattacharjee, “Local-Gravity-Face (LG-face) for Illumination-Invariant and Heterogeneous Face Recognition,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1412–1424, 2016.

Authors’ Profiles

Ni Kadek Ayu Wirdiani was born on 1981. She received her Bachelor’s of Engineering (S.T.) degree in Communication Electronics, Electrical Engineering Department from Udayana University, Indonesia on 2004. She received her Master’s of Engineering (M.T) degree in Information Technology Management, Electrical Engineering Department from Udayana University, Indonesia on 2011.

In 2012, she joined Departement of Information Technology, Udayana University, Bali, Indonesia as a Lecturer. She joined STMIK STIKOM Indonesia in Bali, Indonesia as a Lecturer from 2012 until 2013. Her representative published articles lists as follows: Rural Road Mapping Geographic Information System Using Mobile Android (International Journal of Computer Science Issues), Herbs Recognition Based on Android using OpenCV (International Journal of Image, Graphics and Signal Processing), Color Image Segmentation

Using Kohonen Self-Organizing Map (IACSIT International Journal of Engineering and Technology). Her research interests are Biometrics and Image Processing.

Tita Lattifia is currently an undergraduate student in Information Technology Departement from Udayana University, Indonesia.

I Kadek Supadma is currently an undergraduate student in Information Technology Departement from Udayana University, Indonesia.

Boy Jehezekiel Kemanang Mahar is currently an undergraduate student in Information Technology Departement from Udayana University, Indonesia.

Dewa Ayu Nadia Taradhita is currently an undergraduate student in Information Technology Departement from Udayana University, Indonesia.

Adi Fahmi is currently an undergraduate student in Information Technology Departement from Udayana University, Indonesia.

How to cite this paper: Ni Kadek Ayu Wirdiani, Tita Lattifia, I Kadek Supadma, Boy Jehezekiel Kemanang Mahar, Dewa Ayu Nadia Taradhita, Adi Fahmi, " Real-Time Face Recognition with Eigenface Method", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.11, No.11, pp. 1-9, 2019.DOI: 10.5815/ijigsp.2019.11.01