Face Recognition Using Eigenface Algorithm on Laptop Camera

1st Rika Rosnelly
Department of Engineering and Computer Science
Universitas Potensi Utama
Medan, Indonesia
rikarosnelly@gmail.com

2nd Mutiara S. Simanjuntak
Department of Engineering and Computer Science
Universitas Potensi Utama
Medan, Indonesia
sarahwaty.mutiara@gmail.com

3rd Ade Clinton Sitepu
Department of Engineering and Computer Science
Universitas Potensi Utama
Medan, Indonesia
adeclintonsitepu@gmail.com

4th Mulkan Azhari
Department of Engineering and Computer Science
Universitas Potensi Utama
Medan, Indonesia
mulkanazhari@gmail.com

5th Sandy Kosasi
Department of Engineering and Computer Science
STMIK Pontianak
Pontianak, Indonesia
sandykosasi@yahoo.co.id

6th Husen
Department of Engineering and Computer Science
Universitas Potensi Utama
Medan, Indonesia
husinsariangsah1@gmail.com

***Abstract*—The eigenface algorithm is a collection of eigenvectors used for face recognition through computers. The face recognition system is part of image processing that recognizes faces based on imagery that is stamped and stored in an image file in JPEG format. Face recognition problems can be solved through the implementation of an algorithm. The algorithm used in this study is the eigenface algorithm. The input image is stamped through a laptop camera with a size of 320 pixels x 240 pixels and reduced to 100 pixels x 100 pixels to be saved as a master file of the face with various facial expressions is a forward-facing position without a smile, facing forward a thin smile, facing forward with a big smile, head tilted to the left, and head tilted to the right. The purpose of this research is to build face recognition software using eigenface algorithms. The results showed that faces could be recognized using the eigenface algorithm with an average accuracy rate of 85%.**

Keywords—Camera, Image, Eigenface, Pixel, Face

I. Introduction

The development of science and technology related to data and statistical analysis of biological data (biometrics) is growing rapidly. Much electronic equipment can help relieve human tasks. Refers to biometrics technology, the technology aims to measure and analyze human characteristics such as face patterns mainly used for face recognition processes. A person reading face characteristics needs reader equipment, as a database capable of storing face pattern data and software that analysis such data[1]. Face recognition system as identity verification has been developed and produced various algorithms for digital imagery process, and then it takes algorithms that can recognize faces.

Research on face recognition was conducted using Principal Component Analysis (PCA) with a successful percentage of face recognition processes of 82.81%. Some of the factors that influence the success of recognition are exposure to the face, the distance of the face with the webcam, the number of images of people's faces stored, and the performance of the computer used [2]. Other research on face recognition using the eigenface algorithm yielded an average match percentage of 88% when the database contained 10 face data, results, while at the time the database amounted to 20 face data, the average match percentage result reached 52%. The cause of the difference in results is due to exposure factors, distance, face shape, and the amount of data available [3]. Other research on face recognition using the Hidden Markov Model (HMM) and Fast Fourier Transform (FFT) methods with test results by recalling the accuracy rate is 41%, and overall testing is 36%. Accuracy depends heavily on clustering capabilities [4]. Other research on face recognition uses the deep neural networks method Convolutional Neural Networks (CNN) as real-time face recognition that has proven to be highly efficient in face classification[5]. The results of the trial using the construction of the Convolutional Neural Networks model into 7 layers with inputs from extended local binary pattern extraction results with a radius of 1 and neighbor 15 showed face recognition performance achieved an average accuracy rate of more than 89% in ∓ 2 frames per second [6]. Research on face recognition algorithms is quite numerous and varied, but they all have the same three basic stages the face detection, test face feature extraction, and face recognition[4].

A biometric verification system has two kinds of errors, namely errors in receiving a false acceptance rate (FAR) and errors in rejecting a false rejection rate (FRR). FAR and FRR are functions of the threshold value (t) [7]. In general, face recognition systems are divided into two types, namely feature-based systems and image-based systems [8]. In the feature-based system, features are used to extract from face imagery components such as the eyes, nose, mouth, etc. [9], which are then geometrically modeled on the relationship between them. In contrast, the image-based system uses raw information from image pixels, which is then represented in the Principal Component Analysis (PCA) method or wavelet transformation[10].

Based on research and consideration of the causative factors of differences in previous match results, the authors conducted research on face recognition using the eigenface algorithm [11]. The eigenface algorithm is a collection of eigenvectors used for face recognition using the help of laptop cameras[12].

II. Overview

A. Face Recognition

The face recognition process consists of several components that can be described in a model as seen in Figure

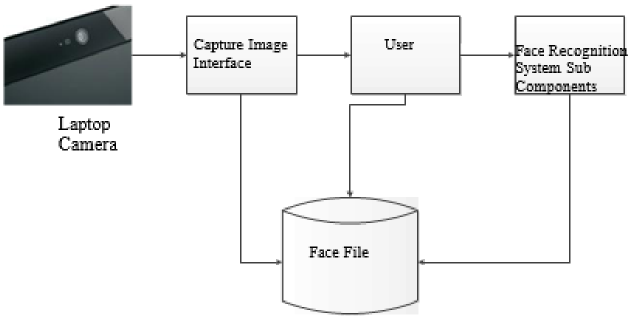

Fig. 1. System Component Analysis Model

Some face imagery will be retrieved and stored in the face database. To function to recognize the face, then the face that appears on the webcam is stamped through a designed application then processed in the application using the eigenface algorithm. After completion of the process, then the face image is stored in the folder. The stages of face recognition by the eigenface method can be seen in Figure 2.

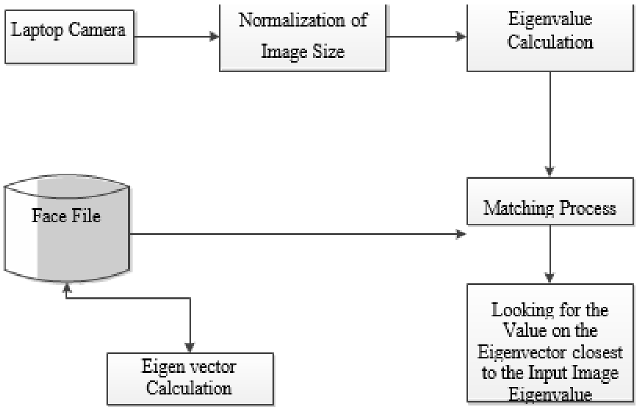
Laptop Camera
Normalization of Image Size
Eigenvalue Calculation
Face File
Matching Process
Eigen vector Calculation
Looking for the Value on the Eigenvector closest to the Input Image Eigenvalue

Fig. 2. Face Recognition Stages

Based on Figure 2 here are the stages of face recognition with the eigenface method:

- The laptop camera is available and active.
- Through the application designed face stamped then the size is normalized according to the size that has been set in the application
- Perform eigenface calculations and then process matching by taking an image of the face in the folder and then doing the eigenvector calculation
- After the calculation of the eigenvector is searched for the eigenvector value, that is, the eigenvalue of the image of the face inputted or stamped.

B. *Eigenface Calculation*

The face recognition algorithm starts by creating a matric of columns of faces that are input and stored in a file. The average vector image (mean) of the column matric is calculated by dividing it by the number of images stored in the file. The first step is to fill variable S with the input face image slice. Each image is transformed into a vector with an N size.

- Specify the middle or mean value.
- Then specify the difference between the input image and the mean image

III. Method

A. *Preparation of Flatvector Matric Imagery*

Face recognition measures with eigenvalue and eigenvector approaches represent all training matrices into a matric with Nx1 or linear matric as below:

$$\begin{bmatrix} a\ c\ d \\ x\ y\ z \end{bmatrix} \rightarrow [a\ b\ c\ x\ y\ z$$

Image training is done to shrink the size to make it easier to generate eigenvector calculations to 3 pixels x 3 pixels. Below are four face training matrices:

Face Image -1

Face matric -1

$$C_1 = \begin{bmatrix} 10 & 11 & 16 \\ 12 & 11 & 15 \\ 9 & 9 & 14 \end{bmatrix} \rightarrow [\ 10\ 11\ 16\ 12\ 11\ 15\ 9\ 9\ 14]$$

Face Image -2

Face matric -2

$$C_2 = \begin{bmatrix} 8 & 10 & 8 \\ 8 & 12 & 11 \\ 5 & 10 & 10 \end{bmatrix} \rightarrow [8\ 10\ 8\ 8\ 12\ 11\ 5\ 10\ 10]$$

Face Image -3

Face matric -3

$$C_3 = \begin{bmatrix} 10 & 8 & 15 \\ 8 & 12 & 13 \\ 7 & 10 & 14 \end{bmatrix} \rightarrow [\ 10\ 8\ 15\ 8\ 12\ 13\ 7\ 10\ 14]$$

Face Image -4

Face matric -4

$$C_4 = \begin{bmatrix} 8 & 7 & 13 \\ 8 & 13 & 13 \\ 7 & 11 & 14 \end{bmatrix} \rightarrow [\,8\ 7\ 13\ 8\ 13\ 13\ 7\ 11\ 14\,]$$

Of the four face training, matric was obtained 1 x (H x W) from matric C1, matric C2, matric C3, and matric C4 as follows:

$$C_1 + C_2 + C_3 + C_4 = \begin{bmatrix} 10 & 11 & 16 & 12 & 11 & 15 & 9 & 9 & 14 \\ 8 & 10 & 8 & 8 & 12 & 11 & 5 & 10 & 10 \\ 10 & 8 & 15 & 8 & 12 & 13 & 7 & 10 & 14 \\ 8 & 7 & 13 & 8 & 13 & 13 & 7 & 11 & 14 \end{bmatrix}$$

B. Selecting a Template Calculation of Flatvector Flattening (Mean) Matric Imagery

From the flatvector obtained, sum the entire row so that it gets a matric measuring 1 x (H x W). After that, divide the matric by the number of imagery (N), which in the example is two, to get flatvector flattening (mean):

$$\Psi = \frac{C_1 + C_2 + C_3 + C_4}{4}$$

$$\Psi = \begin{bmatrix} 10 & 11 & 16 & 12 & 11 & 15 & 9 & 9 & 14 \\ 8 & 10 & 8 & 8 & 12 & 11 & 5 & 10 & 10 \\ 10 & 8 & 15 & 8 & 12 & 13 & 7 & 10 & 14 \\ 8 & 7 & 13 & 8 & 13 & 13 & 7 & 11 & 14 \end{bmatrix} \div 4$$

$$= \left[\frac{36\ 36\ 52\ 36\ 48\ 52\ 28\ 40\ 52}{4} \right]$$

= [9 9 13 9 12 13 7 8 13]

So mean flatvector is = (9 9 13 9 12 13 7 8 13)

C. Calculation of eigenface values

The mean value of the eigenface image for the arranged flatvector matric can be calculated by subtracting the rows on the flatvector matric with the mean flatvector value. If a negative value is obtained, then replace the value with 0. The calculation of eigenface values is as follows:

$$\left(\begin{array}{ccccccccc} 10 & 11 & 16 & 12 & 11 & 15 & 9 & 9 & 14 \\ 9 & 9 & 13 & 9 & 12 & 13 & 7 & 8 & 13 \\ \hline 1 & 2 & 3 & 3 & 0 & 2 & 2 & 1 & 1 \end{array} - \right) \begin{array}{l} \rightarrow \text{Face Image Matric} - 1 \\ \rightarrow \text{Mean Flatvector} \\ \rightarrow \text{Matric x} - 1 \end{array}$$

$$\left(\begin{array}{ccccccccc} 8 & 10 & 8 & 8 & 12 & 11 & 5 & 10 & 10 \\ 9 & 9 & 13 & 9 & 12 & 13 & 7 & 8 & 13 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \end{array} - \right) \begin{array}{l} \rightarrow \text{Face Image Matric} - 2 \\ \rightarrow \text{Mean Flatvector} \\ \rightarrow \text{Matric x} - 2 \end{array}$$

$$\left(\begin{array}{ccccccccc} 10 & 8 & 15 & 8 & 12 & 13 & 7 & 10 & 14 \\ 9 & 9 & 13 & 9 & 12 & 13 & 7 & 8 & 13 \\ \hline 1 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 1 \end{array} - \right) \begin{array}{l} \rightarrow \text{Face Image Matric} - 3 \\ \rightarrow \text{Mean Flatvector} \\ \rightarrow \text{Matric x} - 3 \end{array}$$

$$\left(\begin{array}{ccccccccc} 8 & 7 & 13 & 8 & 13 & 13 & 7 & 11 & 14 \\ 9 & 9 & 13 & 9 & 12 & 13 & 7 & 8 & 13 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 1 \end{array} - \right) \begin{array}{l} \rightarrow \text{Face Image Matric} - 4 \\ \rightarrow \text{Mean Flatvector} \\ \rightarrow \text{Matric x4} \end{array}$$

Matric x-1 to x-4 are combined to obtain eigenface matric for training in the introduction process.

IV. Result

So to recognize the testface image, the identification step is to calculate the eigenface value for matric testface with how to determine the value of eigenface and flatvector image.

Testface

Correspondent Matric

$$\begin{pmatrix} 2 & 8 & 14 \\ 9 & 14 & 15 \\ 7 & 8 & 14 \end{pmatrix} \rightarrow (2\ 8\ 14\ 9\ 14\ 15\ 7\ 8\ 14)$$

Flatvector testface

Furthermore, the flatvector to be obtained testface is reduced by mean flatvector.

$$\begin{array}{c} \begin{pmatrix} 2\ 8\ 14\ 9\ 14\ 15\ 7\ 8\ 14 \\ 9\ 9\ 13\ 9\ 12\ 13\ 7\ 8\ 13 \end{pmatrix} \\ \hline 0\ 0\ 1\ 0\ 2\ 2\ 0\ 0\ 1 \end{array} -$$

$$\begin{array}{ll} [1\ 2\ 3\ 3\ 0\ 2\ 2\ 1\ 1] \rightarrow & Eigenface\ face\ image - 1 \\ [0\ 1\ 0\ 0\ 0\ 0\ 0\ 2\ 0] \rightarrow & Eigenface\ facel\ image - 2 \\ [1\ 0\ 2\ 0\ 0\ 0\ 0\ 2\ 1] \rightarrow & Eigenface\ facel\ image - 3 \\ [0\ 0\ 0\ 0\ 1\ 0\ 0\ 3\ 1] \rightarrow & Eigenface\ facel\ image - 4 \end{array}$$

So the eigenvalue of testface is 001022001. The eigenvalue of the testface is used for identification by determining the absolute value of the line 1 reduction in the image training eigenface matric with the eigenface of the testface and sum it with the vector constituent element resulting from the reduction and obtained distance d index 1 and look for the least d value.

Calculation of distance between the face-1 image and testface calculation:

$$\begin{array}{ccccccccc} 1 & 2 & 3 & 3 & 0 & 2 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 2 & 2 & 0 & 0 & 1 \\ \hline 1 & 2 & 2 & 3 & -2 & 0 & 2 & 1 & 0 \end{array} -$$

$$\begin{array}{c} \downarrow \\ Absolute\ (positive\ value) \\ 1\ \ 2\ \ 2\ \ 3\ \ 2\ \ 0\ \ 2\ \ 1\ \ 0 \\ = 13 \end{array}$$

Calculation of distance between face image -2 and testface:

$$\begin{array}{ccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 2 & 2 & 0 & 0 & 1 \\ \hline 0 & 1 & -1 & 0 & -2 & -2 & 0 & 2 & -1 \end{array} -$$

$$\begin{array}{c} \downarrow \\ Absolute\ (\text{positive value}) \\ 0\ \ 1\ \ 1\ \ 0\ \ 2\ \ 2\ \ 0\ \ 2\ \ 1 \\ = 9 \end{array}$$

Calculation of distance between face image -3 and testface:

$$\begin{array}{ccccccccc} 1&0&2&0&0&0&0&2&1\\ 0&0&1&0&2&2&0&0&1\\ \hline 1&0&1&0&-2&-2&0&2&0 \end{array}$$

$$\downarrow$$

$$\textit{Absolute}\ \text{(positive value)}$$

$$\begin{array}{ccccccccc} 1&0&1&0&2&2&0&2&0 \end{array}$$

$$= 8$$

Calculation of distance between face image -4 and testface:

$$\begin{array}{ccccccccc} 0&0&0&0&1&0&0&3&1\\ 0&0&1&0&2&2&0&0&1\\ \hline 0&0&-1&0&-1&-2&0&3&0 \end{array}$$

$$\downarrow$$

$$\textit{Absolute}\ \text{(positive value)}$$

$$\begin{array}{ccccccccc} 0&0&1&0&1&2&0&3&0 \end{array}$$

$$= 7$$

Description:

- Image spacing -1 *with testface* = 13
- Image spacing -2 *with testface* = 9
- Image spacing -3 *with testface* = 8 d
- Image spacing -4 *with testface* = 7

Fig. 3. Face image similar to testface image

V. Conclusion

After testing and analysis of the research, the following conclusions can be obtained:

- Face recognition with the eigenface algorithm can identify faces with a variety of different facial expressions.
- Face recognition test results using the eigenface algorithm.

Acknowledgment

We would like to extend our gratitude for the support from STMIK Pontianak and Universitas Potensi Utama for funding this research fully.

References

[1] R. Rosnelly, "Combination of Thresholding and Otsu Method in Increasing Results of Identification of Malaria Parasite Type in Thin Blood Smear Image," *Int. J. Psychosoc. Rehabil.*, vol. 24, no. 4, pp. 3807– 3818, 2020, doi: 10.37200/ijpr/v24i4/pr201494.

[2] M. R. Muliawan, B. Irawan, and Y. Brianorman, “Implementasi Pengenalan Wajah DenganMetode Eigenface Pada Sistem Absensi,” *J. Coding, Sist. Komput. Untan*, vol. 03, no. 1, pp. 41–50, 2015.

[3] R. Arlando Saragih, “Pengenalan Wajah Menggunakan Metode Fisherface,” *J. Tek. Elektro*, vol. 7, no. 1, pp. 158–170, 2007, doi: 10.9744/jte.7.1.50-62.

[4] A. Tahmasebi and H. Pourghassem, "Signature identification using dynamic and HMM features and KNN classifier," *Proc. - 2013 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2013*, pp. 201–205, 2013, doi: 10.1109/CSNT.2013.51.

[5] W. Wanayumini, O. S Sitompul, M. Zarlis, S. Suwilo, and A. M H Pardede, "A Research Framework for Supervised Image Classification For Tornado Chaos Phenomena," *Int. J. Eng. Technol.*, vol. 7, no. 4.15, p. 447, 2018, doi: 10.14419/ijet.v7i4.15.25254.

[6] T. S. Gunawan *et al.*, "Development of video-based emotion recognition using deep learning with Google Colab," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 5, pp. 2463–2471, 2020, doi: 10.12928/TELKOMNIKA.v18i5.16717.

[7] A. N. Azmi, D. Nasien, and F. S. Omar, "Biometric signature verification system based on freeman chain code and k-nearest neighbor," *Multimed. Tools Appl.*, vol. 76, no. 14, pp. 15341–15355, 2017, doi: 10.1007/s11042-016-3831-2.

[8] M. S. Simanjuntak, R. Wijaya, J. T. Informatika, and P. I. Komputer, “Melalui Gejala Dan Pola Hidup Menggunakan Metode,” vol. 3, no. 3, pp. 122–125, 2019.

[9] R. Rosnelly, L. Wahyuni, and J. Kusanti, "Optimization of Region of Interest (ROI) Image of Malaria Parasites," *J. Appl. Intell. Syst.*, vol. 3, no. 2, pp. 87–95, 2018, doi: 10.33633/jais.v3i2.2060.

[10] Y. Fang and Z. Wang, Improving LBP features for gender classification, vol. 1. 2008.

[11] Wanayumini, O. S. Sitompul, M. Zarlis, and S. Suwilo, "Automatic Detection of Chaos Phenomenon in Tornadoes Prediction Using Edge Detection," *2019 3rd Int. Conf. Electr. Telecommun. Comput. Eng. ELTICOM 2019 - Proc.*, pp. 48–52, 2019, doi: 10.1109/ELTICOM47379.2019.8943845.

[12] M. S. Simanjuntak, R. Wijaya, and P. I. Komputer, “APLIKASI PENCARIAN GAMBAR DENGANALGORITMA CONTENT-,” vol. 3, no. 3, 2019.