EDUCATUM JSMT Vol. 3 No. 2 (2016)
ISSN 2289-7070 / eISSN 2462-2451 (1-7)

Application of Principal Component Analysis for Face Recognition Based on Weighting Matrix Using Gui Matlab

Y. A. Lesnussa[1], **N. A. Melsasail**[2] **& Z. A. Leleury**[3]
Mathematics Department, Faculty of Mathematics Natural Science, University of Pattimura
Jln. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon, Post 97233
e-mail: [1]yopi_a_lesnussa@yahoo.com; [3]zetharthur82@gmail.com

Abstract

The increasingly widespread of using computers in daily life has brought the piranti as assistant for human. One of the application computer in the field of security has increased it is role is in terms of facial recognition. Face recognition is the process of human identification with the face image. With the increasingly widespread use of computers, is expected to face recognition capabilities can be adapted on the smart device. The adaption process became possible with the discovery of variety of facial recognition methods, one of which is the Principal Component Analysis (PCA). Research began by designing a computer program using the programming language Matlab. The program is used to test the results of PCA with several facial image. At the end it can be concluded that PCA is quite worthy of a face recognition method. The research data shows recognition results were pretty good with quite small error rate.

Keywords PCA, weighting matrix, face recognition

INTRODUCTION

Technology development aims to meet human needs and provide easiness. One of them by creating a computer that has a similar intelligence or human intelligence approach, called Artificial Intelligence, such as: Expert System, Artificial Neural Network, Computer Vision, Fuzzy Logic, and etc. Computer Vision is a field of science that focuses on the field of artificial intelligence system and relate to the acquisition and images processing. One focus of computer vision is pattern recognition. The Data can be recognized by the computer device such as: a sound, image, text, and others. Today the application of pattern recognition is very important especially in the field of security, one of which is face recognition. Face recognition is a process that compares the face image data base entries with a face, and find the face that have best fits with the insert image. This process is widely adopted, for example: in building security, appliance identification, ATM, teleconferencing, tracking tools and other criminal.

There are many facial recognition techniques which can be used like as Histogram Equalization, Chi Square, Principal Component Analysis (PCA), etc. PCA also called Karhunen-Loeve transform introduced by M Turk & Pentland around 1991. The PCA procedure is basically aimed at simplifying the observed variables in a way to its dimensions. Face recognition is a problem in visual pattern recognition. Where in a face that represented into a three-dimensional image (3D) contained various levels of enlightenment, lighting, poses, expressions and others then performed the identification process based on the image information of two-dimensional (2D). A parameters used for face recognition process is through search a special feature location of the image, such as the eyes, nose and mouth, which then calculate the distance between its.

Today most researchers have write about the application of principal component analysis. For example, according to Hui Kong, 2005 publish a research about generalized 2D principal component analysis for face image representation and recognition, Aluko J. Olubunmi, 2015 write about performance evaluation of selected principal component analysis-based techniques for face image recognition. Furthermore, Madhuri M. Ghodake in 2013 publishes about Face Recognition Using Principal Component Analysis for Security Based System and also S. Thakur and friends publish about Face Recognition using Principal Component Analysis and RBF Neural Networks. Moreover, face recognition using principal component analysis and linear discriminant analysis on holistic approach in facial images database by Satonkar Suhas S, 2012.

PCA is a technique of dimension reduction that is commonly used in image processing applications. The basic principles of the PCA algorithm are to determine and maintain the components of a set of images that have a maximum distribution. The combination of the face image this is called the Eigenface. Eigenface-Eigenface this is a feature of a face image to be identified.

MATERIAL & METHODS

This research uses the primary data that obtained by taking images faces of several students and processed using PCA. This method begin with feature extraction based on PCA by calculate the matrix weight of the image and then perform image recognition test. The calculation using eigenface method (Holistic feature-based) by comparing the image that has been projected to be grayscale level has become image with a low-dimensional. The Common things of PCA technique are to devise a training image into a matrix and find the minimum distance of weighted matrix.

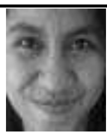
RESULTS & DISSUCSION

Facial images used in this research were obtained by taking photographs of students face both men and women in the Department of Mathematics. The amount of data used as many as 40 students with 10 different facial pose of each person. With 10 faces pose is then divided into two classes of which 4 face pose of each person is used as the test data, while 6 other poses are used as training data. The main thing to note is the entire digital image both training data and test data of the face taken in the same lighting conditions should be normalized to level grayscale and processed at the same resolution that is 92 × 112 pixels, then the image is converted to forms the vector with the dimensions 92 × 112 in which components are taken from the value of the pixel image.

Image Data Training

If known 8 sample image with a size of 8 x 10 pixels as shown in Table 1, and will be determined projection matrix and weight matrix of 8 sample images.

Table 1 Sample image of eight facial as training data

Class	Members 1	Members 2	Class	Members 1	Members 2
1	*f(1)* 	*f(5)* 	3	 *f(3)*	*f(7)* 
2	*f(2)* 	*f(6)* 	4	*f(4)* 	*f(8)* 

By using Matlab program, can obtained value of each gray level image can be shown in the matrix $f(1)$ to $f(8)$.

$$f(1) = \begin{bmatrix} 103 & 100 & 102 & 97 & 105 & 95 & 79 & 63 & 35 & 12 \\ 116 & 109 & 71 & 78 & 114 & 87 & 71 & 65 & 66 & 13 \\ 130 & 93 & 67 & 51 & 111 & 74 & 61 & 53 & 61 & 28 \\ 127 & 120 & 94 & 110 & 116 & 84 & 79 & 95 & 95 & 54 \\ 118 & 149 & 101 & 95 & 124 & 81 & 69 & 102 & 108 & 60 \\ 95 & 146 & 81 & 113 & 94 & 64 & 75 & 57 & 104 & 42 \\ 16 & 150 & 114 & 83 & 97 & 87 & 56 & 90 & 73 & 4 \\ 10 & 17 & 133 & 143 & 108 & 90 & 97 & 64 & 4 & 11 \end{bmatrix}$$

$$\vdots$$

$$f(8)=\begin{bmatrix} 42 & 45 & 51 & 63 & 59 & 61 & 57 & 45 & 44 & 47\\ 39 & 25 & 23 & 35 & 53 & 44 & 20 & 22 & 36 & 53\\ 43 & 28 & 28 & 33 & 43 & 41 & 38 & 34 & 42 & 54\\ 51 & 55 & 56 & 41 & 53 & 60 & 52 & 63 & 65 & 61\\ 56 & 62 & 48 & 43 & 44 & 42 & 44 & 57 & 65 & 63\\ 61 & 38 & 29 & 35 & 36 & 45 & 41 & 37 & 55 & 62\\ 90 & 46 & 47 & 41 & 65 & 87 & 49 & 52 & 57 & 71\\ 101 & 60 & 43 & 46 & 44 & 41 & 58 & 55 & 55 & 74 \end{bmatrix}$$

Transform two-dimensional matrix into one dimension, then for the entire training data compiled sorted. Results for 8 training data transformation can be seen in the matrix $f_{8\times 80}$.

$$f_{8\times 80}=\begin{bmatrix} f(1)_{1,1}=103 & f(1)_{1,2}=100 & \cdots & f(1)_{1,80}=11\\ f(2)_{1,1}=98 & f(2)_{1,2}=102 & \cdots & f(2)_{1,80}=91\\ f(3)_{1,1}=144 & f(3)_{1,2}=161 & \cdots & f(3)_{1,80}=13\\ f(4)_{1,1}=93 & f(4)_{1,2}=110 & \cdots & f(4)_{1,80}=109\\ f(5)_{1,1}=92 & f(5)_{1,2}=138 & \cdots & f(5)_{1,80}=147\\ f(6)_{1,1}=108 & f(6)_{1,2}=114 & \cdots & f(6)_{1,80}=136\\ f(7)_{1,1}=111 & f(7)_{1,2}=120 & \cdots & f(7)_{1,80}=89\\ f(8)_{1,1}=42 & f(8)_{1,2}=45 & \cdots & f(8)_{1,80}=74 \end{bmatrix}$$

The result of the transformation matrix is averaged from column 8 to column 80, so the average number of elements is 80 or has a dimension of 1 x 80. The result of the calculation can be written as a matrix of rows μ. For ease the calculation, can be arranged matrix such as the size of the training data as the original matrix size of 1 x 80 converted into 8 x 10. The average value is then rounded as seen in the matrix μ as following:

$$\mu=\begin{bmatrix} 99 & 111 & 110 & 110 & 99 & 94 & 81 & 65 & 63 & 48\\ 104 & 81 & 64 & 83 & 115 & 90 & 62 & 65 & 71 & 60\\ 116 & 83 & 57 & 70 & 113 & 78 & 63 & 62 & 78 & 67\\ 125 & 116 & 103 & 111 & 119 & 94 & 86 & 101 & 104 & 86\\ 128 & 135 & 100 & 106 & 118 & 97 & 83 & 102 & 116 & 89\\ 122 & 113 & 96 & 110 & 96 & 86 & 84 & 77 & 104 & 76\\ 112 & 121 & 104 & 100 & 97 & 91 & 80 & 89 & 87 & 76\\ 116 & 100 & 120 & 123 & 108 & 99 & 104 & 93 & 73 & 84 \end{bmatrix}$$

The next step is to double the average value of which has been obtained by the amount of training data. If the dimensions of the average value obtained is 1 x 80, then the result of doubling the number of training data as much as 8 x 80 dimensions, and generated the zero mean matrix, shown in the following:

$$\emptyset=\begin{bmatrix} 4{,}125 & 11{,}75 & 13{,}625 & \cdots & -72{,}75\\ -0{,}875 & -10{,}25 & -32{,}375 & \cdots & 7{,}25\\ 45{,}125 & 49{,}75 & 41{,}625 & \cdots & -70{,}75\\ -5{,}875 & -8{,}25 & 1{,}625 & \cdots & 25{,}25\\ -6{,}875 & -5{,}25 & 16{,}625 & \cdots & 63{,}25\\ 9{,}125 & 15{,}75 & 15{,}625 & \cdots & 52{,}25\\ 12{,}125 & 11{,}75 & 16{,}625 & \cdots & 5{,}25\\ -56{,}875 & -65{,}25 & -73{,}375 & \cdots & -9{,}75 \end{bmatrix}$$

Zero mean value calculation results have a size of 8 x 80, or $\emptyset_{8\times 80}$, so the size of the transpose $\emptyset_{8\times 80}$ is $\emptyset_{80\times 8}$. To get the value of covariance, then do multiplication operation between zero mean to transpose zero mean and the result is divided by the amount of data and generated the eigen vector matrix.

$$E=\begin{bmatrix} 4{,}1638 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 2{,}6348 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0{,}6963 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0{,}6018 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0{,}2749 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0{,}1435 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}0000 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0{,}0617 \end{bmatrix}$$

Table 2 Changes Position of Eigenvalue

Initial Index	Eigen value of the initial value	Index Changes		Value of new Eigen value
		originally	become	
1	4,1638	1	1	4,1638
2	2,6348	2	2	2,6348
3	0,6963	3	3	0,6963
4	0,6018	4	4	0,6018
5	0,2749	5	5	0,2749
6	0,1435	6	6	0,1435
7	0,0000	7	8	0,0617
8	0,0617	8	7	0,0000

Based on Table 2 can be seen changes in the index. For example eigen value to 7 moved into 8th position, eigen value to 8 moved into 7th position, while for other eigen value position has not changed. Changes must also be followed by changes in the position of eigen vector.

$$Eigen\ vector=\begin{bmatrix} 0,1230 & 0,5147 & -0,0861 & 0,1772 & -0,3949 & -0,6112 & 0,1633 & 0,3536 \\ 0,1776 & 0,0063 & 0,4199 & 0,4301 & 0,6762 & -0,0914 & 0,1287 & 0,3536 \\ -0,3658 & 0,6004 & 0,0795 & -0,0510 & -0,0182 & 0,6067 & -0,0583 & 0,3536 \\ -0,1764 & -0,1889 & -0,8184 & 0,2369 & 0,2476 & -0,0048 & -0,1447 & 0,3536 \\ -0,3121 & -0,4545 & 0,1135 & -0,0224 & -0,2729 & 0,0968 & 0,6883 & 0,3536 \\ -0,2379 & -0,3408 & 0,3479 & 0,1347 & -0,3090 & -0,1205 & -0,6731 & 0,3536 \\ -0,0049 & -0,0103 & 0,0202 & -0,8377 & 0,3119 & -0,2682 & -0,0598 & 0,3536 \\ 0,7965 & -0,1269 & -0,0765 & -0,0678 & -0,2406 & 0,3926 & -0,0445 & 0,3536 \end{bmatrix}$$

It is seen that the dimension of the matrix V_1 is 80 x 8, and find matri V. The projection matrix can be obtained from the transposed matrix V. projection matrix is used to project the data set of trials to obtain the weight of test data sets. $W_{8\times 8}$ is a weighting matrix generated.

$$W_{8\times 8}=\begin{bmatrix} 0,0980 & 0,0091 & 0,0000 & -0,0006 & -0,0001 & -0,0011 & -0,0000 & 2,3926 \\ 0,0943 & -0,0066 & 0,0001 & -0,0011 & -0,0010 & -0,0005 & -0,0000 & 2,4508 \\ 0,1319 & 0,0117 & 0,0000 & -0,0003 & -0,0004 & 0,0002 & -0,0000 & 3,1768 \\ 0,1188 & -0,0126 & -0,0000 & -0,0007 & -0,0006 & -0,0004 & -0,0000 & 2,9441 \\ 0,1281 & -0,0208 & 0,0000 & -0,0003 & -0,0002 & -0,0003 & 0,0000 & 3,2447 \\ 0,1230 & -0,0173 & 0,0001 & -0,0006 & -0,0001 & -0,0006 & -0,0000 & 3,1363 \\ 0,1069 & -0,0071 & 0,0000 & 0,0011 & -0,0007 & -0,0007 & -0,0000 & 2,6733 \\ 0,0514 & -0,0107 & 0,0000 & -0,0002 & -0,0002 & -0,0000 & -0,0000 & 1,3745 \end{bmatrix}$$

Row 1 is the weighting matrix of images 1st, the weight is a characteristic of the image and have sorted starting from the right (most dominant) to the left (less dominant). So that the PCA process produces two important values those are the projection matrix (projections of experimental data to generate weights) and the weighting matrix (matrix characteristics).

Image Data Testing

If known one image to be tested has a value of gray as the matrix Test.

Table 3 Sample of test image (*Testing*)

Test Image	Test image matrix
	$f(test)=\begin{bmatrix} 167 & 213 & 219 & 224 & 214 & 212 & 199 & 190 & 159 & 70 \\ 173 & 179 & 191 & 218 & 227 & 201 & 164 & 158 & 149 & 128 \\ 168 & 192 & 152 & 205 & 213 & 191 & 142 & 146 & 182 & 171 \\ 206 & 196 & 210 & 203 & 200 & 190 & 216 & 209 & 207 & 161 \\ 207 & 221 & 224 & 204 & 208 & 187 & 199 & 215 & 202 & 156 \\ 216 & 219 & 217 & 215 & 216 & 206 & 185 & 205 & 195 & 187 \\ 246 & 222 & 202 & 189 & 194 & 184 & 178 & 205 & 182 & 148 \\ 194 & 85 & 203 & 212 & 219 & 216 & 205 & 166 & 169 & 34 \end{bmatrix}$

To compare the similarity of this image with training image, need to determine the class of the test image, by transform the image Tests into a rows matrix, test weights calculate the image using the equation, test weights measure the similarity with weight training results. Test results matrix transformation into a row matrix. Test weights image can be calculated by multiplying the transformation matrix Test results with the projection matrix training. Furthermore, to determine the grade of the tested image, needs to be measured using the Manhattan method, this method can be calculated by absolute the result of reduction in weight training for weight data to be tested or otherwise. For first weight data training, the distance of weights first training data to the weighting of test data using Manhattan is as follows:

Table 4 Distance of weight data training to the weight data test using Manhattan

Weights Data Training	Weight Data Tested	Results Reduction
0.0980e+005	0.2088e+005	-0.1108e+005
0.0091e+005	-0.0188e+005	0.0279e+005
0.0000e+005	0.0001e+005	-0.0001e+005
-0.0006e+005	-0.0019e+005	0.0013e+005
-0.0001e+005	-0.0008e+005	0.0007e+005
-0.0011e+005	-0.0007e+005	-0.0004e+005
-0.0000e+005	-0.0000e+005	0.0000e+005
2.3926e+005	5.2763e+005	-2.8837e+005

The results of the calculations have been rounded off, and then summed up. So the distance of first weight training data to the weight test data is $Z_1 = 3.0177\text{x}\ 1.0\text{e} + 005$. The result calculation of the distance between the weight data training others to the test data that is: $Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8$ each of which can be seen in the following table:

Table 5 The distance of weight training data to the weights test

No.	Weights Training	Distance of Weight Test Data	Information
1	1	3.0177e+005	
2	2	2.9464e+005	
3	3	2.2030e+005	
4	4	2.4248e+005	
5	5	2.1240e+005	Minimum distance
6	6	2.2335e+005	
7	7	2.7127e+005	
8	8	4.0633e+005	

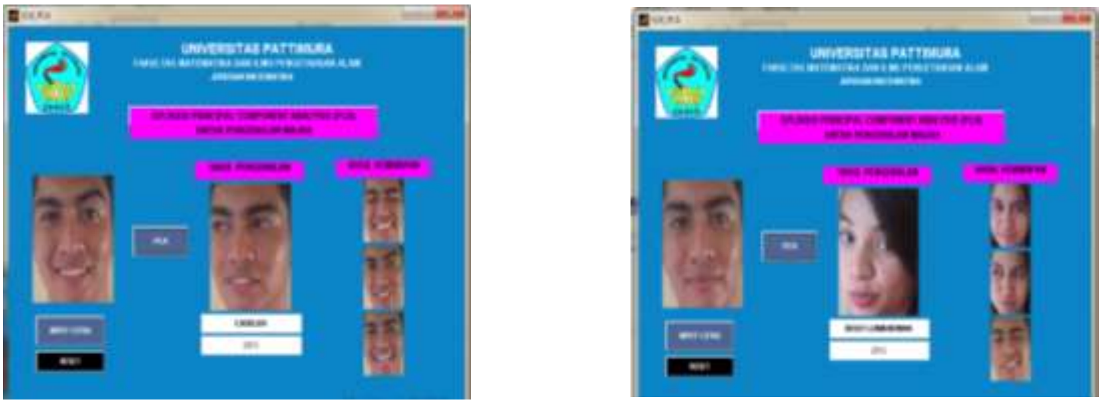
This means that the test image of the tested is similar to the data fifth. The Data fifth is first class, so that the image is being tested in the category of first class

Simulation by Matlab

Testing was conducted using 240 training face images taken with multiple expressions as shown in the following figure. In this study, used 400 face images of 10 people. Everyone has taken a sample of 10 face images consisting of six images for training and 4 images for testing.

Figure 1 Image Data *Training*

This test is performed to determine the success of face recognition system that has been made by the method Eigenface PCA. With the facial image data as discussed previously with the method of face recognition stage Eigenface then this can be done by testing the facial image by finding the nearest training images into the PCA subspace.



(a) *(b)*

Figure 2 Testing Results show the similarity of facial recognition
(a) The success of test image; (b) Failed of test image

It appears from the results that the program can recognize objects that were tested with several kinds of facial expressions and positions. After testing the facial image 400 turns true testing as many as 352 and 48 failed to obtain the percentage of success of the test is: 88 %. While controlling the failed about 12%. Errors in testing due to several factors, such as:

Firstly, about position of Pose Image. Pose frontal or facial position perpendicular to the location of the cameras tend to have the similar pattern or texture between the current face image with trainer face pose that tested. There is no rotation (shift angle) in the frontal position that causes the facial image recognition rate one becomes small. While the position of non-frontal angle tend to differences corner result in any recognition.

Secondly, about the distance of image to camera. Student distances from camera have effect by resizing process of the face image, this errors caused by the light intensity. The bright light intensity will produce the different quality of images compare to the dark light intensity. The bright light and dark light intensity will produce the different features of pattern of the same image. Features of different textures of image will produce different composition of the matrix. The pattern of matrix could be differences between the two images caused by different intensities and it will generate the different of recognition results. The more similar composition of the matrix to image then the true percentage will be even greater recognition.

CONCLUSION

From the results it can be concluded, that: PCA method is used for face recognition by converting images into numbers in a matrix, subsequently transformed into a vector. Comparison of two similar images is by comparing the shortest distance between two vectors that corresponding to the two images are compared, so the closer of two vectors, the more similar of two images

REFERENCES

[1] Atalay, I. (1996). *Face recognition using eigenfaces*. Istambul Technical University, Turki.

[2] A.M. Martinez and A.C. Kak. (2001). *PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell*., Vol. 23, no. 2, pp. 228-233.

[3] Arif Muntasa. (2013). Line detection model and adaptive threshold based image segmentation for handwriting word regocnition. *International Conference on Electrical, Informatics and Its Education*, Malang State University.

[4] Arif Muntasa, Hariadi. M, Purnomo. H. M. (2009). A new formulation of face sketch multiple features detection using pyramid parameter model and simultaneous landmark movement. *International Journal of Computer Science and Network Security*, Vol. 9 Nomor 9.

[5] Arif Muntasa, Hariadi. M, Purnomo. H. M. (2008). *Penyeleksian eigenface secara otomatis untuk pengenalan citra wajah.* SITIA, ITS.

[6] Calder, A.J., Burton, A.M, Miller, P., Young, A.W., and Akamatsu, S. (2001). A principle componen analysis of facial expressions. *Vision Research* 41: 1179-1208.

[7] Jianke, Z., Mang, V. and Peng, U.M. (2002). *Face Recognition Using 2D DCT with PCA*. University of Macau, Macau.

[8] Kresimir Delac, Mislav Grgic, Panos Liatsis. (2005). Appearance Based Statistical Methods for Face Regocnition. *47th International Symposium ELMAR*, Zadar, Croatia Sidik jari.

[9] Sax, J.D., and Dillon, O.W. (1963). The simulation of radiation. *Radiation Journal*, No. 2, Addison-Wisley, San Fransisco, 173-179.

[10] Turk, M., and Pentland. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* 3: 71-86.