ISBN 978-602-53531-8-5

Alumni Face Recognition Using Fisherface Method

Asep Nana Hermana[1,*], Irma Amelia Dewi[1], Gelar Aulia Prima Putra[2]

[1] Department of informatics, Institut Teknologi Nasional (Itenas), Bandung - INDONESIA

[2] Student Department of informatics, Institut Teknologi Nasional (Itenas), Bandung - INDONESIA

* Corresponding author e-mail: asepnana1966@gmail.com

Abstract

Face is part of the human body that became one of the uniqueness and characteristics of each individual. From time to time changes in the face will of course happen by many factors, especially age, in this case the face of alumni. Alumni is a group of people who have attended or graduated from a school or college. One of the technologies used to identify faces is facial recognition. Fisherface is one of the methods used to recognize a person's face. This study has produced a system that can help search alumni information quickly only by using face variable. Testing has been done by taking a face image using a face detector that has been inserted in the face recognition application on Android. Taken image data will be matched with the training data that has been stored in the database using the Euclidean Distance approach. The Euclidean distance between the training data and the test image data will then be calculated and searched for the lowest distance between those two. The lowest Euclidean distance means that the test image data tested has a high degree of compatibility with the training data. From the tests results can be concluded that the Fisherface method can recognize well test image data using 150 training data and 75 test image data with 100% accuracy percentage on neutral face, 80% on smiling face, 60% on face with glasses attribute, 86.66% with make-up, and 100% with sideways face.

Keywords: Face Recognition, Fisherface, Android, Euclidean Distance, Alumni

1. INTRODUCTION

1.1. Background

Difficulty to find information about alumni is still one of the problems experienced by some parties who still have business with alumni or parties who just want to find out about alumni information. Currently the way to obtain alumni information is by using the registration number of students to look for that most likely the data is not the latest data from the alumni. On the other hand, many photographs of alumni are displayed on the campus building, this could be an opportunity that can be used to search for alumni data. Therefore a system is needed that could help to obtain alumni information quickly only by using faces as the information. With the face recognition system, the search process of alumni data would be faster. Therefore, to assist in searching the alumni data, the development of facial recognition system for alumni uses Fisherface method.

1.2. Problem Identification

Based on the background of this final project proposal, it can be formulated problems that exist in this research, namely:

1. How attributes or changes in expression affect facial recognition processes.
2. How the facial recognition process with Fisherface method applied to Android-based devices.
3. How the implementation of Google Mobile Vision is used as a system to detect faces.

1.3. Aim

The purpose of this research is to apply Fisherface method to make facial recognition in order to recognize alumni on Android based mobile device.

1.4. Problem Scope

Problem scope on the research that will be implemented are:

1. Face that to be recognized is the faces of alumni of 2012 Itenas Informatics Engineering Majors.
2. The part used in face recognition is to take the whole of each pixel on a frame that has a face.
3. Frame used for facial processing only contains one face.
4. The size of the image to be used for the face recognition process is 100 x 150 pixels.

2. RESEARCH METHODOLOGY

2.1. Literature Study

Literature study in the form of reading the source of books, journals and articles about the work of facial recognition done with Fisherface method and also about the workings of applications running on Android

2.2 Data Collection Techniques

Data collection techniques by taking a photo or image that has a face, then tested with the image that has been through the process of data training so that the system can recognize the face.

2.3 Research Subjects

Research subjects in this study is the image or photo that has a face in it to examine how much the accuracy of Fisherface as face recognition method.

3. DISCUSSION

3.1. System Work Process

The development of face recognition system has two basic processes. the first one is the registration process, as the stage for entering the face image into the database as data for the training process, and the second one is the identification process, as the stage where the face recognition system is used to recognize the test face image. Figure 1 illustrates how the process performed by the system to recognize faces.

3.1.1. Workflow System

Here is an overview of how the process of facial recognition system is built. Workflow system has two basic processes, that is the registration process and identification process. The picture can be seen in figure 1..

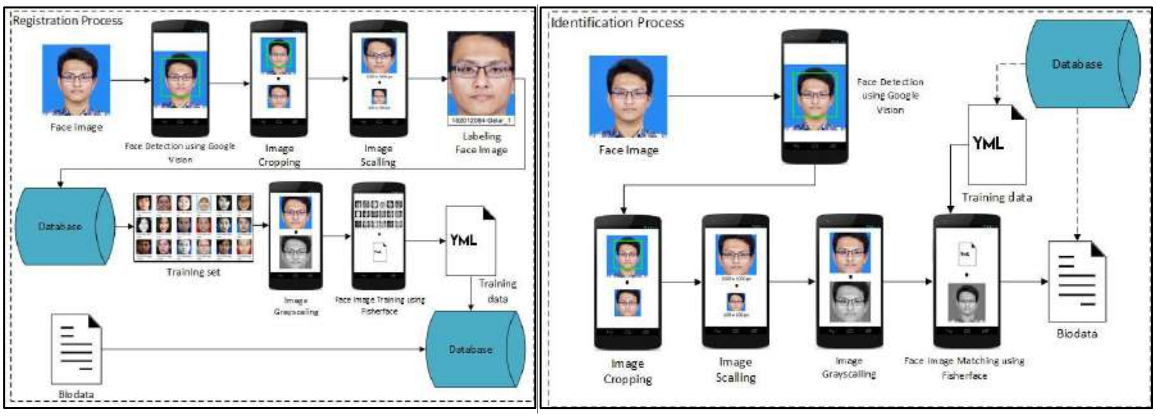

Fig. 1: Workflow system

3.1.2. Flowchart System

With the workflow used as a reference for the built system, here is an overview of each process that is done by the system as a flowchart. The system description can be seen in Figure 2 as the registration flowchart, and Figure 3 as the flowchart identification.

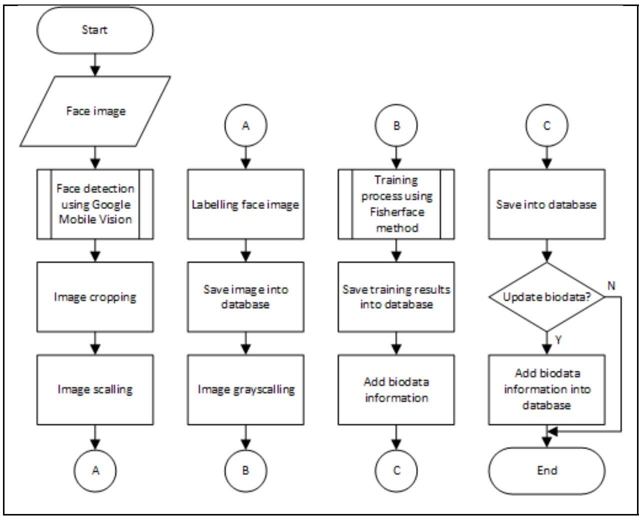

Fig. 2: Registration flowchart

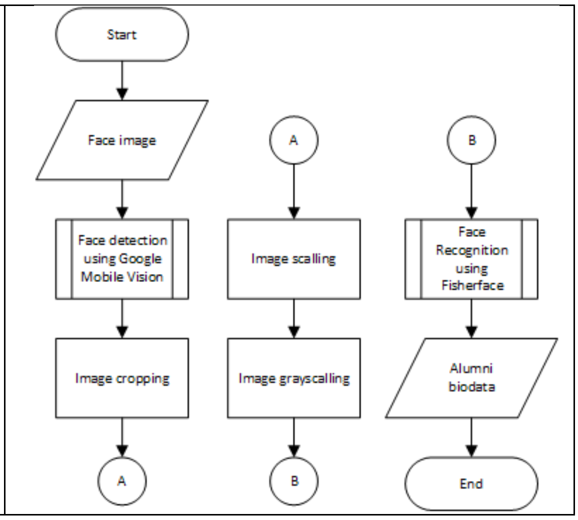

Fig. 3: Identification flowchart

3.2. Face Detection Stage

The face detection system used on systems built using face detection is owned by Google Mobile Services, that is the Vision API. This Google Mobile Vision system can detect faces that exist in an image and generate values or variables that can be used for other processes. Here is the results of the researcher how the face detection process conducted by Google Mobile Vision using flowchart in Figure 4.

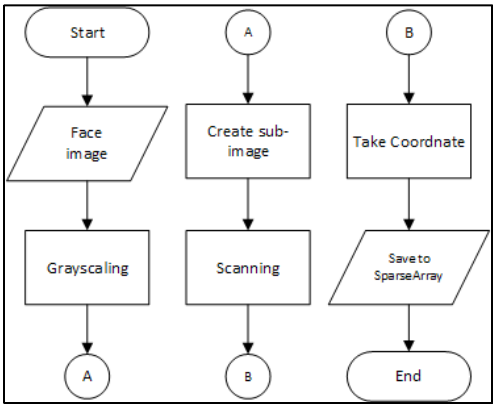

Fig. 4: Google Mobile Vision Face Detection Flowchart

3.3. Pre-Processing Stage

3.3.1. Cropping Process

This process are done so that the face image to be trained or tested will be simplified so that it can be processed by the system. As the name implies, this process of cropping or cutting the image image in accordance with the position of the detected face on the image.

This process uses 4 variables to crop the image. These variables are horizontal axis coordinate values, vertical axis coordinate values, face length values, face width values.

The value of this variable is obtained from the face detection process using Google Mobile Vision, because at the time of face detection, Google Mobile Vision will provide these values.

Suppose there are images like illustrations in figure 5, measuring 4 x 3 pixels, meaning that there is a total of 12 pixels in the image.

	0		1		2		3	
0	R	31	R	42	R	42	R	44
	G	33	G	44	G	44	G	46
	B	22	B	33	B	33	B	35
1	R	46	R	41	R	64	R	82
	G	48	G	43	G	66	G	84
	B	37	B	32	B	55	B	73
2	R	188	R	186	R	193	R	157
	G	189	G	187	G	194	G	158
	B	181	B	179	B	186	B	150

Fig. 5: Input image illustration

With the face detection system gives for example the value 0 for the horizontal axis coordinates and the value 0 for the vertical axis coordinates (coordinates (x, y) = (0,0)). From the coordinates that have been obtained, it will continue with the next process with the face detection system gives for example the value 2 for the width of the face, and the value of 2 for the face value is detected.

	0		1		2		3	
0	R	31	R	42	R	42	R	44
	G	33	G	44	G	44	G	46
	B	22	B	33	B	33	B	35
1	R	46	R	41	R	64	R	82
	G	48	G	43	G	66	G	84
	B	37	B	32	B	55	B	73
2	R	188	R	186	R	193	R	157
	G	189	G	187	G	194	G	158
	B	181	B	179	B	186	B	150

Fig. 6: Illustration of cropping area from image

Once obtained where the image coordinates and the image size, now it can be created a new image with the same size to contain the value of the old image.

	0		1	
0	R	31	R	42
	G	33	G	44
	B	22	B	33
1	R	46	R	41
	G	48	G	43
	B	37	B	32

Fig. 7: Image cropped part illustration

3.3.2. Grayscalling Process

Grayscaling process is a process whereby an image that originally has three color values, namely red, green, and blue (RGB) is converted into a single color value that can be called a gray color. Suppose there is an image input that has a 2 x 2 pixel dimension with its RGB values as follows:

Table 1: RGB values on a 2 x 2 pixel image

		y			
		0		1	
x	0	R	31	R	42
		G	33	G	44
		B	22	B	33
	1	R	46	R	41
		G	48	G	43
		B	37	B	32

Equation 1,

$$Grayscale(x,y) = \frac{R(x,y) + G(x,y) + B(x,y)}{3}$$

Explanation :
$Grayscale(x,y)$ = Grayscale value at x and y coordinate
$R(x,y)$ = Red value at x and y coordinate
$G(x,y)$ = Green value at x and y coordinate
$B(x,y)$ = Blue value at x and y coordinate

The calculation is performed on each coordinate and RGB value in the calculation is taken from the RGB value in table 1. From the calculation we have obtained grayscale value that has been rounded to be processed at the next stage. The results of the calculation of grayscale value can be seen in table 2 below.

Table 2: Grayscale value on 2 x 2 pixel image

		Y	
		0	1
x	0	29	40
	1	44	39

3.4. Facial Recognition Phase Using Fisherface Method

Phase owned by Fisherface method has two basic processes used to simplify the image form that later results from both of these processes will enter the process of finding conclusions using Euclidean Distance calculation. The process is PCA (Principal Component Analysis) and FLD (Fisher Linear Discriminant) calculations as can be seen in Figure 8, Fisherface process flowchart.

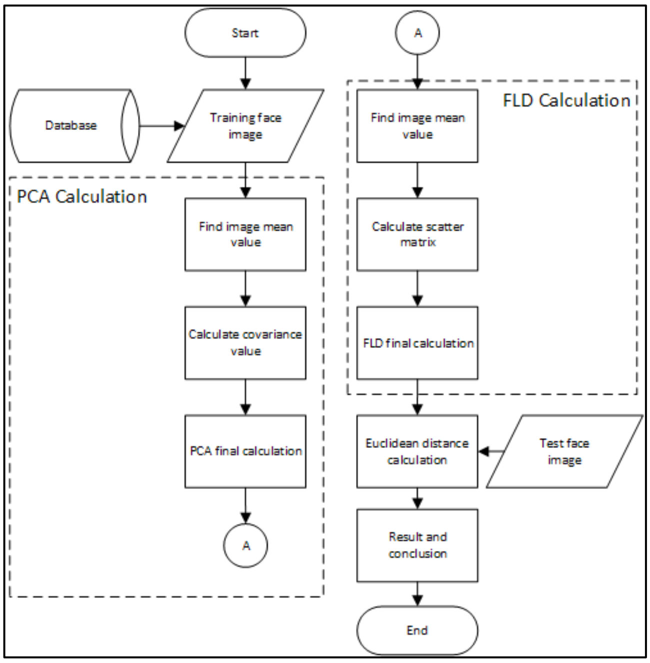

Fig. 8: Fisherface method flowchart

Here's an explanation of the picture 8:

The face image of the training included for the training process is taken from a database previously filled with facial images entered during the registration process, along with the name / label of the file.

29	44
40	39

Figure 9: First training image illustration

52	14
40	29

Figure 10: First training image illustration

After getting the facial images to be trained, then the images will be converted into flatvector, that is a one dimensional matrix. With illustration of image image 9 and picture 10 we got flatvector of each image as illustration of flatvector arrangement in figure 11.

29	44										
40	39		29	40	44	39					
Citra 1								29	40	44	39
								52	40	14	29
52	14		52	40	14	29					
40	29										
Citra 2			*flatvector*					*flatvector* gabungan			

Fig. 11: flatvector arrangement

3.4.1. PCA Calculation

PCA is a technique where the processed data will undergo dimensional reduction process. But the weakness of PCA is that this calculation technique does not include labels as information from the processed data, making this PCA didn't store information from the class where the data came from. Therefore, in the Fisherface method, PCA calculation techniques will be combined with FLD calculation techniques.

After the vector of the face image is obtained, add the line to the flatvector and then divide the matrix by the number of images to obtain the average value of the image vector or its average value on each face image (average face) with equation 2.

Equation 2	Explanation:
$avg = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n}$	Avg = Average value $(x_1 + x_2 + x_3 + \cdots + x_n)$ = matrix value n = amount of data

After obtaining the average value of the image, we will find the value of variance and its covariance using equation 3 and equation 4.

Equation 3	Explanation:
$var(x_i) = \sum_{i=1}^{n} \frac{(x_i - avg_{x_i})^2}{(n-1)}$	$var(x_i)$= variance value x_i= *i*-th matrix value avg_{x_i}= Average value *n* = amount of data
Equation 4 $cov(x_1, x_2) = \sum_{i=1}^{n} \frac{(x_{i1} - avg_{x_{i1}}) * (x_{i2} - avg_{x_{i2}})}{(n-1)}$	Explanation: $cov(x_1, x_2)$= Covariance value

After we get the value of variance and covariance of each image, then the value of variance and covariance is entered into equation 5 that is calculation of covariance matrix.

Equation 5	Explanation:
$Mcov = \begin{bmatrix} var(x_1) & cov(x_2, x_1) \\ cov(x_1, x_2) & var(x_2) \end{bmatrix}$	*Mcov* = Covariance Matrix value

After obtaining the covariance matrix value as in equation 5, the value of the calculation result of this covariance matrix will be recalculated to find the eigenvalues and the eigenvector value. Both of these, eigenvalues and eigenvectors, are used to obtain the component score value (PCA) using equation 6,

Equation 6	Explanation:
$\lvert Mcov - \lambda I \rvert = 0$	λI= Eigenvalue

After the eigenvector value has been obtained, the last step of the PCA calculation process is to find the value score of the component of PCA using equation 7.

Equation 7	Explanation:
$Wpca = \sum (eigenvector1 * a) + (eigenvector2 * b)$	*Wpca* = component score value *a, b* = subtraction of image vector value with the mean

After the PCA calculation is complete and the value of the component score is obtained, then the system will send the score value to the next process is the calculation of FLD.

3.4.2. FLD Calculation

FLD is one of the class specific methods, where it seeks to establish inter-class spacing between in-class scatter and intra-class scatter in order to produce good data classification. The purpose of this FLD is to reduce the dimensions of a data, but also keep the label information that shows the class or uniqueness of the data.

Find what is the average value of each image class with the following 8 equations.

Equation 8	Explanation:
$mean_{vector} = \frac{1}{n} \sum \begin{bmatrix} x1 \\ x2 \end{bmatrix}$	$mean_{vector}$= average value of each data class *n* = amount of data x1, x2 = value of each data

After obtaining the average value of each class of data, then the results of the equation 8 will be used to find the value of the difference matrix with the average of each class using equation 9.

Equation 9	Explanation:
$Si = \sum (x_{class\ i} - mean_{class\ i})^2$	*Si* = subtraction value of matrix with the mean $x_{class\ i}$ = matrix value of each class $mean_{class\ i}$= mean value of each matrix

After obtaining the value of the matrix difference with the average of each class, then the values will be processed to find the value between-class scatter and within-class scatter using equations 10 and 11.

Equation 10	Explanation:
$S_w = \sum Si$	S_w = within-class scatter value $\sum Si$ = amount of subtraction value of matrix with the mean
Equation 11 $S_b = (\mu 1 - \mu 2) * (\mu 1 - \mu 2)^T$	Explanation: S_b = between-class scatter value $\mu 1$ = $mean_{class\ 1}$ $\mu 2$ = $mean_{class\ 2}$

If the within-class scatter and between-class scatter matrix values have been found, the scatter values as well as previously obtained PCA values will be used to find the FLD value using equation 12.

Equation 12	Explanation:
$J(w) = \frac{W_{pca} S_b w}{W_{pca} S_w w}$	*J(w)* = FLD value W_{pca} = PCA component value S_w = within-class scatter value S_b = between-class scatter value

3.4.3. Euclidean Distance

Euclidean Distance is a calculation that becomes the distance between two points. Illustration of Euclidean distance calculation will be explained as follows. Suppose that there are four face image data already stored in the database and the value is described in table 3.

Table 3: traning image data illustration

x-th person	*x*-th Pixel			
	1	2	3	4
1	5	5	4	12
2	10	4	4	3
3	4	12	7	10
4	15	5	13	12

Also note the value of the image to be tested in table 4. The value of this image that will be compared with the image of training data,

Table 4: test image data illustration

x-th person	*x*-th Pixel			
	1	2	3	4
1	6	3	7	11

With equation 13, Euclidean Distance value can be searched between test data with each training data,

Equation 13

$$d_{i,j} = \sqrt{\sum_{k=1}^{m} (x_{i,k} - j_{i,k})^2}$$

Explanation:

$d_{i,j}$ = Euclidean distance *i*-th training image, *j*-th test image

$x_{i,k}$ = *i*-th person and *k*-th pixel training image value

$j_{i,k}$ = *i*-th person and *k*-th pixel test image value

With the equation 13, we can calculate the Euclidean distance value of each training data with the test data. After each training data is calculated the distance of Euclidean it with test data and got the result, then the system will compare each result Euclidean distance and sought the lowest value which can be seen in table 5.

Table 5: each training data Euclidean distance value comparision

Number.	Euclidean Distance value	Result
$d_{1,1}$	3.8729	Has lowest value
$d_{2,1}$	9.4868	
$d_{3,1}$	9.2736	
$d_{4,1}$	11.0453	

4. TESTING

4.1. Testing Level Accuracy

Below is the result of facial recognition test using Fisherface. How to determine the success of face recognition is seen from the value of its Eucledian Distance, the smaller the value of its distance, the greater the suitability of the face. This test was conducted using 150 training data and 75 test data.

To calculate what percentage success rate of each test can be calculated by equation 15:

Equation 15

$$\frac{Sucessful\ tests}{amount\ of\ testing} * 100\%$$

Table 6: Neutral face test

Num.	Test Data	Lowest Euclidean Distance	Nearest training data	Information
1.	1st person	1537.5744	1st person	Matched
2.	2nd person	1600.8019	2nd person	Matched
3.	3rd person	926.6310	3rd person	Matched
4.	4th person	894.5393	4th person	Matched
5.	5th person	921.0817	5th person	Matched
6.	6th person	1343.0229	6th person	Matched
7.	7th person	849.1632	7th person	Matched
8.	8th person	634.4488	8th person	Matched
9.	9th person	774.6221	9th person	Matched
10.	10th person	1049.9138	10th person	Matched
11.	11th person	999.5180	11th person	Matched
12.	12th person	981.5346	12th person	Matched
13.	13th person	1283.1518	13th person	Matched
14.	14th person	846.0421	14th person	Matched
15.	15th person	1727.7218	15th person	Matched

The success rate for neutral face recognition testing is:

$$\frac{15}{15} * 100\% = 100\%$$

Table 7: Smile expression face test

Num.	Test Data	Lowest Euclidean Distance	Nearest training data	Information
1.	1st person	1628.8189	14th person	Not Matched
2.	2nd person	1366.5821	2nd person	Matched
3.	3rd person	978.8660	3rd person	Matched
4.	4th person	878.5860	4th person	Matched
5.	5th person	955.8139	5th person	Matched
6.	6th person	1430.6518	6th person	Matched
7.	7th person	972.5187	11th person	Not Matched
8.	8th person	939.7925	8th person	Matched
9.	9th person	1060.0308	9th person	Matched
10.	10th person	1191.7007	10th person	Matched
11.	11th person	1040.6404	11th person	Matched
12.	12th person	1249.6730	12th person	Matched
13.	13th person	1300.4484	11th person	Not Matched
14.	14th person	1136.6180	14th person	Matched
15.	15th person	1203.1110	15th person	Matched

The success rate for testing facial recognition of a smile expression is:

$$\frac{12}{15} * 100\% = 80\%$$

Table 8: Glasses attribute face test

Num.	Test Data	Lowest Euclidean Distance	Nearest training data	Information
1.	1st person	1675.9737	1st person	Matched
2.	2nd person	1992.5491	1st person	Not Matched
3.	3rd person	1580.6384	10th person	Not Matched

4.	4th person	1453.5284	10th person	Not Matched
5.	5th person	1390.7373	5th person	Matched
6.	6th person	1445.4361	6th person	Matched
7.	7th person	1255.4550	11th person	Not Matched
8.	8th person	1622.6678	8th person	Matched
9.	9th person	1306.5664	10th person	Not Matched
10.	10th person	1323.5438	10th person	Matched

Table 9: Glasses attribute face test (Continuation)

Num.	Test Data	Lowest Euclidean Distance	Nearest training data	Information
11.	11th person	1783.5147	11th person	Matched
12.	12th person	930.1917	12th person	Matched
13.	13th person	1621.4787	1st person	Not Matched
14.	14th person	1357.0009	14th person	Matched
15.	15th person	1032.5162	15th person	Matched

The success rate for facial recognition testing with eyeglass attributes is:

$$\frac{9}{15} * 100\% = 60\%$$

Table 10: Make-up face test

Num.	Test Data	Lowest Euclidean Distance	Nearest training data	Information
1.	1st person	1759.4565	14th person	Not Matched
2.	2nd person	1399.9398	8th person	Not Matched
3.	3rd person	1053.6622	3rd person	Matched
4.	4th person	1070.2269	4th person	Matched
5.	5th person	905.5183	5th person	Matched
6.	6th person	1240.4045	6th person	Matched
7.	7th person	908.1701	7th person	Matched
8.	8th person	1322.8184	8th person	Matched
9.	9th person	1096.1935	9th person	Matched
10.	10th person	1254.6527	10th person	Matched
11.	11th person	1042.1934	11th person	Matched
12.	12th person	1150.3180	12th person	Matched
13.	13th person	1422.5011	13th person	Matched
14.	14th person	845.5093	14th person	Matched
15.	15th person	1670.3373	15th person	Matched

The success rate for face recognition testing with make-up is:

$$\frac{13}{15} * 100\% = 86.67\%$$

Table 11: Sideway face test

Num.	Test Data	Lowest Euclidean Distance	Nearest training data	Information
1.	1st person	1168.974	1st person	Matched
2.	2nd person	321.9707	2nd person	Matched
3.	3rd person	303.772	3rd person	Matched
4.	4th person	678.7715	4th person	Matched
5.	5th person	1194.736	5th person	Matched
6.	6th person	634.7459	6th person	Matched
7.	7th person	228.9673	7th person	Matched

8.	8th person	241.6128	8th person	Matched
9.	9th person	249.9228	9th person	Matched
10.	10th person	1079.774	10th person	Matched
11.	11th person	978.3082	11th person	Matched
12.	12th person	1669.471	12th person	Matched
13.	13th person	1172.337	13th person	Matched
14.	14th person	1050.551	14th person	Matched
15.	15th person	637.6841	15th person	Matched

The success rate for sideway facial recognition testing is:

$$\frac{15}{15} * 100\% = 100\%$$

5. Conclusion

Based on the results of face recognition testing that has been done using 150 test data and 75 data of this training can be concluded the use of Fisherface method for face recognition system has 100% accuracy with neutral face, 80% with smile expression, 60% with attribute glasses, 86.66% With make-up, and 100% with a sideways face. The differences that occur in this face recognition test occur because of inadequate training data. For example if you want to recognize faces that use face attributes, training data must have image data that uses face attributes.

It can be concluded that the facial recognition system using Fisherface method is good enough to recognize faces as can be seen on the percentage of success in testing that has been done, with a record of training data that must be adequate. The more face variations that are stored in the database, the more likely it is that the face is recognizable.

6. Bibliography

Ade Bambang Kurnia. 2016. "Implementasi Pendekatan *K-Means* Dalam Menentukan Ekspresi Wajah Terhadap Suasana Hati". Bandung. Teknik Informatika. Institut Teknologi Nasional.

Deni Ramdani. 2011. "Aplikasi Perbandingan Algoritma Metode *Fisherface* Dengan Metode *Eigenface* Pada Sistem Pengenalan Pola Wajah". Bandung. Teknik Informatika. Universitas Komputer Indonesia.

Febrian Ardiyanto. 2007. "Sistem Pengenalan Wajah Berbasis Metoda *Fisherface*". Bandung. Teknik Elektro. Institut Teknologi Bandung.

Google Developers, "Detect Facial Features In Photos", https://developers.google.com/vision/android/detect-faces-tutorial, diakses (October 14, 2016).

Muhammad Boby Nicholas Argadiraksa. 2016. "Deteksi *Gender* Pada *Single Board Computer* Sebagai Kamera CCTV". Bandung. Teknik Informatika. Institut Teknologi Nasional.

Riko Arlando Saragih. 2007. "Pengenalan Wajah Menggunakan Metode *Fisherface*". Bandung. Teknik Elektro. Universitas Kristen Maranata.

Robil Widiyanto. 2013. "Analisis dan Implementasi Algoritma *Fisherface* Pada Sistem Pengenalan Wajah Untuk Keamanan *Handphone* Berbasis Android". Yogyakarta. Teknik Informatika. Universitas AMIKOM Yogyakarta.