Journal of Theoretical and Applied Information Technology
10th March 2014. Vol. 61 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FACE RECOGNITION SYSTEM ON ANDROID USING EIGENFACE METHOD

[1] OKA SUDANA, A. A. K., [2] DARMA PUTRA, I K. G., [3]ALAN ARISMANDIKA
[1,2,3] Department of Information Technology, Udayana University, Indonesia

E-mail: [1]agungokas@hotmail.com, [2] ikgdarmaputra@gmail.com,
[3]arismandika@gmail.com

ABSTRACT

Face recognition is a biometric technology that can be applied to the various fields in terms of human identity. The continued development of technology, face recognition is also adopted in the latest technologies such as android devices are applied to complete the job. In this paper, demonstrates the face recognition system in android device using eigenface. The system can be used as the base for the development of android applications such as android mobile security application and as an archive for the recognition of human identity. The experiment of face recognition system in this paper uses 50 facial images as test images. Test images and training images are taken directly with the camera in android device with a distance of face shooting is 60 cm. The test results showed that the system produces high accuracy with a success rate reaches 94,48% with FMR = 2,52% and FNMR = 3%.

Keywords: *Face Recognition, Eigenface, Threshold, Face Detection, Android Application*

1. INTRODUCTION

The new technologies continue to emerge with a variety of advanced features. It also affects the demand for self recognition system. One of them is face recognition. Face recognition is a complex and difficult problem that is important for surveillance and security, telecommunications, digital libraries, and human-computer intelligent interactions [1] [4] [12].

Eigenface face recognition approaches can be classified in an appearance-based method, because the eigenface face recognition use information from the raw pixel image which is used for training and classification of image identity [3] [6] [8]. The idea of this method is projecting an image of a face that can be seen as a vector. To produce the eigenface, the digital image of human faces take at the same lighting conditions, then normalized into a grayscale image [7] [13]. The image is processed in the same resolution and then used as the vector dimension where the components are derived from the pixel value of the image.

Basically human face recognition procedure consists of two stages. The first stage is where the face detection process takes place very rapidly in humans except in certain circumstances where the object is located at a far distance. The second stage is recognition stage which is recognizing the face as individuals face. The face recognition stages as it was then imitated and developed as a model for facial image recognition is one of the biometric technologies is widely studied and developed by the experts [2]. It is because in general, the face image can provide the specific information related to personal identification [11]. However the face image has high variation as the input. In general, these variations are caused by two factors. The first factor is variation on own face and second factor is the variation caused the object transformation of face into face image. The variations of the face image must be able to be resolved by face recognition system [5].

To solve a variety of problems, face recognition should be applied in the practical and flexible devices. The practical and flexible device is android device. In the display interface, android application is very easy to use by the user because it is the latest portable devices currently. In programming, Android also provides a variety of functions that support the programming of face recognition, as in image processing. [10]. In the case of a database as a repository, android uses the serverless database so the database is easy to manage [9].

From the problems, this paper is built the face recognition system that is solved by eigenface method. The system is run on android because it is the portable devices as described previously. The system is build with the several features for users and features that will require in the eigenface, for example, features face detection and preprocessing. The system through the recognition by the input

Journal of Theoretical and Applied Information Technology
10th March 2014. Vol. 61 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

image directly that take from the android device camera. In the future, the system can be developed for android application.

2. SYSTEM PROPOSED

2.1. System Overview

There are two main jobs in the face recognition system. The jobs are enrollment and identification. In the enrollment process there are the stages in the eigenface calculation, i.e. image acquisition, preprocessing and the threshold value for each image. While in the identification process stages are image acquisition, preprocessing and recognize the test image with eigenface process. The overview of the system is shown in Figure 1.

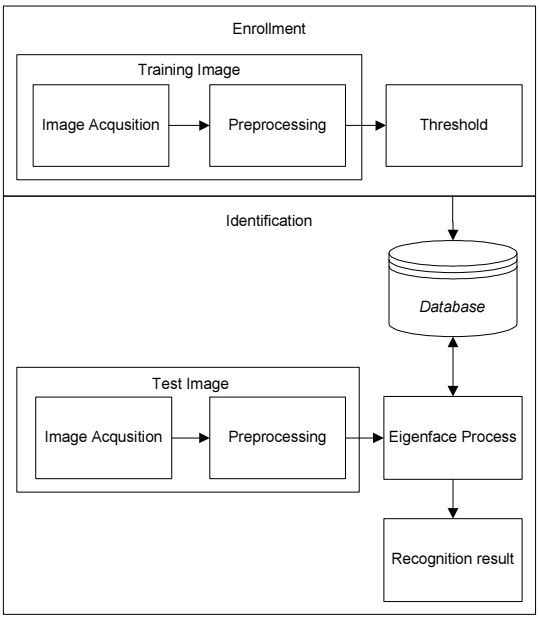


Figure 1: System Main Overview

The enrollment process is the process when the training image is stored and to be use as the reference in the database. Then the identification process is the process when the test face is identified in the recognition process using eigenface.

As the existing eigenface calculation, there are the calculation for the training face and the test face for face recognition [4] [6] [7]. Calculations of the training face and the test face done by calculates the value of each image pixel. The pixel value is changes shape for eigenface calculation. The mean of the pixel matrix of each training face is calculated for reference in the recognition process. That calculation process will be stored in the database. In the test face image is also changing the shape matrix for comparison process with the training face.

The system referenced by Matthew A. Turk and Alex P. Pentland formulas. In the system, the formulas used to complete the eigenface calculation. In accordance with the formulas, the works in the system early is the pixel value of training face in the form of N x H x W is converted into N x 1. From pixel value of N x 1 then searched the mean of all the training face that stored in the database. After getting the results of the mean, then calculated the difference between the pixel value of each training face with the mean. In the test face, the process is almost the same as in the training image step. But in test face only one image that performed the calculation process. The pixel value of training image is also converted into N x 1. Then the difference calculation between N x 1 pixel values with the mean of training face. For the recognition, that calculated the difference of the test face which has undergone a process previously with each training face that has gone through the process. The smallest value of the difference process is used as the recognition result.

2.2. Image Acquisition and Preprocessing

In the system, the process required to take the images. The image used as an object to be recognized. The image is taken directly from the android device. To take the image, the system activated the camera of android device.

The images captured and processed for the detection of the necessary areas, namely face detection process. The initial phase of the system is face detection process to obtain the face of the image capture. In android, the face detection process is done by activating the existing function of the android library [8]. We use the API shipped with Android SDK since level 1. The API is used, namely FaceDetector. It was written:

```
Import android.media.FaceDetector;.
```

FaceDetector used to detect face in the bitmap. To get the image width, image height and the number of faces. In the android face detection, it must get the face midpoint and the eyes distance to detect the faces. The function is:

After get the coordinates for the face detection, next phase is the cropping process. The cropping process is done by cutting the face detection area. The image converted by the bitmap image for cutting process on the face area. The system uses 80 x 80 pixel image for face image which is processed in the face recognition system.

Journal of Theoretical and Applied Information Technology
10th March 2014. Vol. 61 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the process of eigenface, the process before the calculation of eigen value is preprocessing. In the system, preprocessing of the image is converting the RGB image into grayscale image. In android the grayscale image is converted by filtering the image. The RGB image extracted and processed into grayscale image. In the system uses ColorMatrixColorFilter. It was written:

```
Import
android.graphics.ColorMatrixColorFilter;
```

ColorMatrixColorFilter used for color transformation of grayscale process. The function of face detection, cropping and grayscale is:

```
public void FaceDetection(Bitmap Data)
{
    int imageWidth, imageHeight;
    int numberOfFace = 5;
    FaceDetector myFaceDetect;
    FaceDetector.Face[] myFace;
    int [] fpx = null;
    int [] fpy = null;
    PointF midpoint = new PointF();
    imageWidth = Data.getWidth();
    imageHeight = Data.getHeight();
    myFace = new
    FaceDetector.Face[numberOfFace];
    myFaceDetect = new
    FaceDetector(imageWidth,
    imageHeight,numberOfFace);
    count = myFaceDetect.findFaces(Data,
    myFace);
    Bitmap resizedBitmap;
    Bitmap bm;
    if (count > 0) //if image
    {
      fpx = new int[count];
      fpy = new int[count];
      TempBmp = new Bitmap[count];
      int eyesdist;
      int fx;
      int fy;
      for (int i = 0; i < count; i++) {
      try {//Crop, Resize dan Convert//
         myFace[i].getMidPoint(midpoint);
         fpx[i] = (int)midpoint.x;
         fpy[i] = (int)midpoint.y;
         eyesdist = (int)
         myFace[i].eyesDistance();
         fx=fpx[i]- (eyesdist +
         (eyesdist/2)); if(fx < 0){fx = 0;}
         fy=fpy[i]- (eyesdist +
         (eyesdist/2)); if(fy < 0){fy = 0;}
         Bitmap Temp =
         Bitmap.createBitmap(Data, fx, fy,
         ((eyesdist + (eyesdist/2)) * 2),
         ((eyesdist + (eyesdist/2))  * 2));
         int width = Temp.getWidth();
         int height = Temp.getHeight();
         int newWidth = Resolusi;
         int newHeight = Resolusi;
         float scaleWidth = ((float)
         newWidth) / width;
         float scaleHeight = ((float)
         newHeight) / height;
         Matrix matrix = new
         Matrix();matrix.postScale(scaleWidt
         h, scaleHeight);
         resizedBitmap =
         Bitmap.createBitmap(Temp, 0, 0,
         width, height, matrix, true);
         bm = Bitmap.createBitmap(Resolusi,
         Resolusi,Bitmap.Config.RGB_565);
         Canvas c = new Canvas(bm);
         Paint paint = new Paint();
         ColorMatrix cm = new ColorMatrix();
         cm.setSaturation(0);
         ColorMatrixColorFilter f = new
         ColorMatrixColorFilter(cm);
         paint.setColorFilter(f);
         c.drawBitmap(resizedBitmap, 0, 0,
         paint);
         TempBmp[i] = bm;
         resizedBitmap=null;
         bm = null;
         } catch (Exception e) {
         Log.e("setFace(): face " + i + ":
         ", e.toString()); }
        }
        BtnAddPic.setImageBitmap(TempBmp[0]
        );
        b = TempBmp[0];
    }else{
    Toast.makeText(AddData.this, "No Face
    Deteted On Your Selected Image..!!!",
    Toast.LENGTH_LONG).show();
    }
}
```

2.3. Threshold

Threshold is used as a comparison value when the smallest of the eigen value calculation result has been obtained. Threshold obtained by summing the distance between each image of the same face and dividing by the number of images being compared.

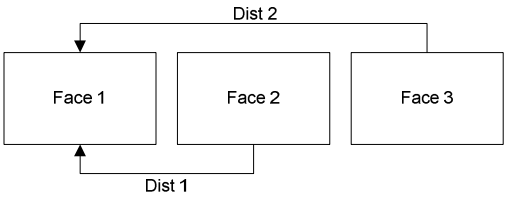


Figure 3: The Distance Of Faces

Figure 3 describes the image 1 is used as the reference to find the distance to image 2 and image 3. The formula of the calculation is:

$$T = \frac{Dist\,1 + Dist\,2}{2} \quad (1)$$

Equation (1) states the threshold that obtained from the sum of the distance between the pixel value of image 1 and pixel value of image 2 (Dist 1) and the distance between the pixel value of image 1 and image 3 (Dist 2) that divided by the number of

Journal of Theoretical and Applied Information Technology
10th March 2014. Vol. 61 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

image distances as much as 2. The result of the mean image is used as the threshold.

3. METHODOLOGY

In general, the work process of the system is to perform the face recognition on image that the user input as a test images and identify the training face images that stored in database. In this section we explain the flow of face recognition system is created on the android platform. In accordance previously described, the system will perform all the processes automatically. Such processes include face detection process, conversion into a grayscale image, calculation of threshold values and eigenface calculation.

In the system, the provisions of the image that is used for face recognition is determined. The image captured is converted into 80 x 80 pixel image. 80 x 80 pixel image is converted to form the image of 6400 x 1.

In a face recognition system workflow, there are 2 main modules including enrollment module and identification module. Enrollment module in the system is the module that explains the enrollment process to obtain the training image as a reference which is will save to the database. There are three images of face input in the enrollment process. The images use to get the threshold value.

Identification is a module which describes the process of identification in the system. Identification module is describes the process flow of test image recognition that compared with the training image in database.

3.1. Enrollment Module

In the enrollment module, we explain the enrollment process in the system that created on android. The system flow in the enrollment module is described into several phases. The phases are the work overview in the system for the enrollment process. The enrollment process has five main phases, including image acquisition, preprocessing, threshold calculation, eigenvector calculation and storage. Enrollment module describes the overview of training face enrollment process that will be saved to the database as the reference for face recognition.

The process flow of the enrollment system that is the first phase i.e. the phase of image acquisition to retrieve data such as face images. The system will perform face detection process after captured the image and automatically be cropping on the detected face. In the enrollment process, there are 3 images used for threshold calculation process. The second phase is preprocessing, i.e. the phase to transform the RGB image into grayscale image and 80 x 80 pixel matrix for calculation of eigenvector. In the threshold phase, the distance of 3 images is calculating for threshold result. The distance that calculated is the distance of image 2 and image 3 to image 1. The distance of the images is divided by 2 and the result of the division is set as the threshold. Threshold will save into database. The 80 x 80 pixel matrix from preprocessing is also used for the eigenvector phase. The eigenvector calculation phase i.e. the phase for transform the N x (H x W) pixel matrix namely is 80 x 80 pixel to the N x 1 pixel matrix namely 6400 x 1. The image matrix calculation is to find the mean of flatvector which is summing the matrix of all image input and the result divided with the number of images that generate the mean flatvector. Then each flatvector reduced by a mean flatvector. All the results are stored into the database. The processes are showing in Figure 4.

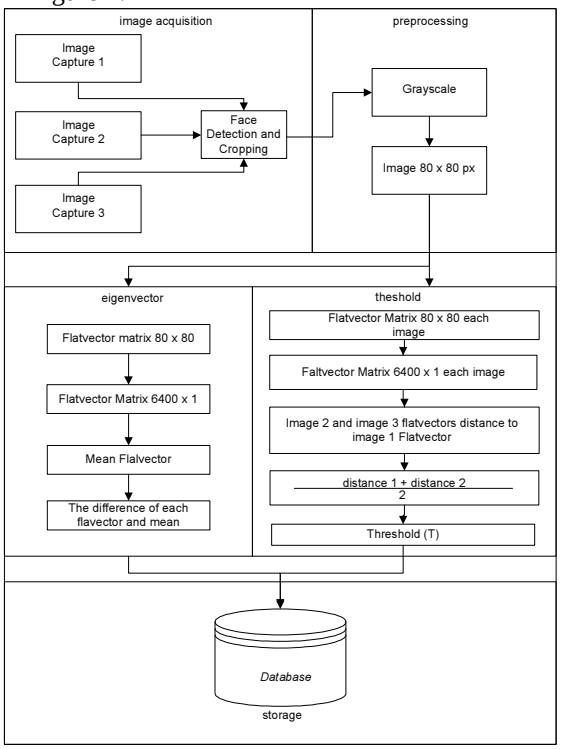


Figure 4: Enrollment Process In System

3.3 Identification Module

Identification module describes the recognition process in the system. In this section, the flow of the recognition processes are described through several stages before finally obtained the results of the test images were identified. The identification process has four main phases i.e. image acquisition, preprocessing, eigenvalue and identification.

Journal of Theoretical and Applied Information Technology
10th March 2014. Vol. 61 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

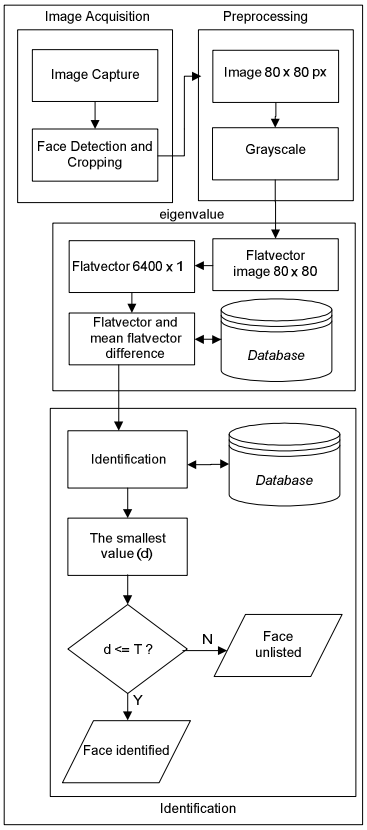


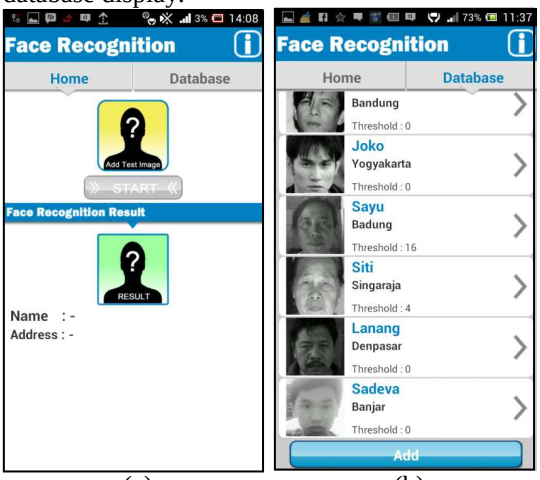
Figure 5: Identification Process In System

Figure 5 shows an overview of the identification process.The process of identification of the system in this paper through the several phases. The first phase is image acquisition phase as the phase for take the test image for recognition. The processes of image acquisition are captured the image, face detection and cropping on the detected face. The second is preprocessing phase. The phase is transforming the image into grayscale image and 80 x 80 image. The 80 x 80 image is transform into 6400 x 1 matrix image in the eigenvalue process. The 6400 x 1 matrix image is dividing with the mean of training face that existing in the database. The latter is identification process. The identification process is calculation the smallest value of the difference between eigenvalue in the test face process and eigenvector in training face that stored in the database. The smallest value (*d*) compared with Threshold (T) of the image is called the identified image. If the value of *d* is smaller than and equal to T then the image identification result is valid. Otherwise, the image is not valid and is not considered registered.

4. IMPLEMENTATION

4.1 System Interface Result

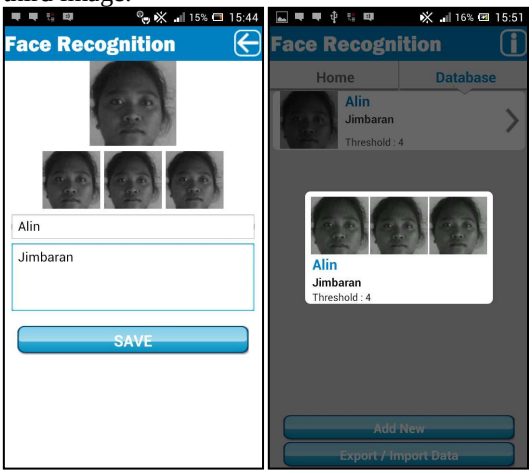
The system interface becomes the main view for the user to use the face recognition system. Eigenface and threshold calculation processes is done automatically by the system. There are two main view of the system that is home display and database display.



(a) (b)

Figure 6:Main Display Of System: (A) Home (B) Database

Figure 6 (a) shows the home display that serves as the recognition result display for the user. While figure 6 (b) is shows the face training that stored in the database.In the add process for training face, take 3 images that will automatically calculate the threshold. The first image input is used as the main image for distance calculation with second and third image.



(a) (b)

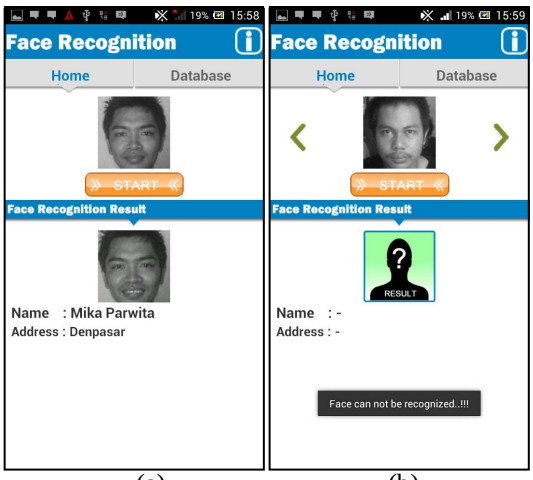
Figure 7: Enrollment Display : (A) Training Face Input (B) Threshold Calculation

Journal of Theoretical and Applied Information Technology
10th March 2014. Vol. 61 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The user facilitated by simply pressing start button to obtain identification results. Figure 9 shows the enrollment of the system. Figure 9 (a) shows training face input in the system and figure 9 (b) shows the training face that has been stored and threshold of the training face.

The identification result will display the existing face training on the database and display the name and address. If the identification result displays the result were not identified, it will displays an alert.



(a) (b)

Figure 8: Identification Result: (A) Identified Test Face (B) Unidentified Test Face

Figure 8 shows the identification result by the system. Figure 8 (a) shows the display of identified face by the system. While figure 10 (b) shows the display of unidentified face.

4.2 The Experiment Results

The successfully experiment rate of the face recognition system using eigenface method obtained by FMR (False Match Rate) and FNMR (False Non Match Rate) calculations. FMR is the error calculation when the system rejects the face that stored into the database. FNMR is the error calculation when the system is receiving or identifying the face who not registered.

FMR and FNMR is the function of the threshold value. If the threshold value is large, it makes the system more tolerant to variations in the input then the FNMR go up. Otherwise, if the threshold value is small, it makes the system more secure but could have rejected the face that stored in the database.

$$FMR = \frac{Number\ of\ Original\ Facial\ Features\ Rejected\ by\ The\ System}{Number\ of\ All\ Tests} \times 100\% \quad (2)$$

$$FNMR = \frac{Number\ of\ False\ Facial\ Features\ Accepted\ by\ The\ System}{Number\ of\ All\ Tests} \times 100\% \quad (3)$$

Equation (2) and (3) shows the rate calculation of experiment result about the error of the system to receive the test face that does not exist in the database and reject the test face that existed in the database. From the calculation, it will give the results of the test accuracy.

In the calculation process of experiment result, the experiment performed using 50 test faces. The shoot of image of all face image is within 60 cm between the camera of android device and owner face. The face image shoot takes at the same place and same lighting. The experiment results are shown in table 1:

Table 1:Table Of Experiment Results.

Threshold	FMR (%)	FNMR (%)	Accuracy (%)
30	0,28	9,88	89,84
31	0,36	9,72	89,92
32	0,48	9,16	90,36
33	0,96	8,44	90,6
34	1,08	8,16	90,76
35	1,16	7,44	91,4
36	1,24	6,24	92,52
37	1,44	6,2	92,36
38	1,56	5,68	92,76
39	1,72	5,04	93,24
40	1,88	4,68	93,44
41	2,04	3,72	94,24
42	2,12	3,44	94,44
43	2,32	3,24	94,32
44	2,52	3	94,48
45	3,16	2,92	93,92
46	3,2	2,76	94,04
47	3,68	2,56	93,76
48	4,6	2,36	93,04
49	5,08	2,16	92,76
50	5,4	2,04	92,56

Based on the experiments performed with 50 test images with 2500 times of experiments, the best accuracy rate is obtained when the threshold is at a value of 44. The accuracy rate is 94,48%. FMR in system is equal to 2,52% and FNMR is equal to 3%. The system has an rejected accuracy about 97,48% and accepted accuracy rate about 97%.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

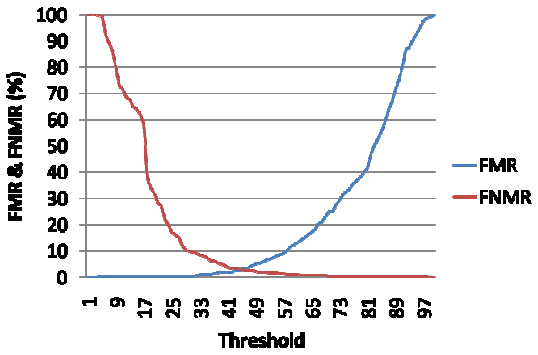


Figure 9:Graph Of Experiment Results

5. CONCLUSION AND FUTURE WORKS

In this paper, we have conducted the research of the eigenface method for face recognition system. The system is built in Java programming for android. The system run and applied on android platform. The experiment of the face recognition system obtained the high accuracy result, with rejected accuracy about 97,48% and accepted accuracy rate about 97%. The accuracy rate show that the system we have created with the eigenface method was applied to the android platform produces the high success rate.

The system that created on the android platform is the system which is run in the mobile android applications and other application in other android devices. For the future, the system that we created can be use as a base to create the android application that uses the face for the application. For example, the key lock application for mobile using face, face recognition application for police fugitive, offenders face archive application and Another application that uses face recognition.

REFERENCES:

[1] Harry Wechsler, "Face recognition from theory to applications", Springer, 1998.
[2] Hadif al Fatta, "Rekayasa Sistem Pengenalan Wajah", Andi. 2009.
[3] Darma Putra, "Konsep Dasar, Teknik Analisis Citra dan Tahapan Membangun Aplikasi Biometrika", Andi, 2009.
[4] Matthew A. Turk and Alex P. Pentland, "Eigenfaces for Face Detection/Recognition", Journal of Cognitive Neuroscience. 1991.
[5] Stan Z. Li, Anil K. Jain, "Handbook of Face Recognition" Springer, 2005.
[6] Vinay Hiremath and Ashwini Mayakar, "Face Recognition Using Eigenface Approach", 2002
[7] Nick Pears Thomas Heseltine and Jim Austin. "Evaluation of image preprocessing techniques for eigenface based face recognition". ACA Group, Dept. of Computer Science, University of York, 2002.
[8] Ive Billiauws, Kristiaan Bonjean and dr.id T. Goedeme, "Image recognition on an Adroid mobile phone", 2009.
[9] Jason Wei, "Android Database Programming", Packt Publishing, 2012.
[10] Alex Pentland, Baback Moghadam and Thad Starner, "View-Based and Modular Eigenspaces for Face Recognition", IEEE Conference on Computer Vision & Technical Report, No.245, 1994.
[11] Brunelli, R., and Poggio, T., "Face Recognition: Features versus Templates", IEEE Trans. Pattern Anal. Machine Intell, vol. 15, pp.1042-1052, 1993.
[12] Arif Muntasa, Indah Agustien Sirajudin and Mauridhi Hery Purnomo, "Appearance Global and Local Structure Fusion fo Face Image Recognition", Telkomnika, Vol.9, No.1, pp. 125-132, April 2011.
[13] Lin, Shang Hung., "An Introduction to Face Recognition Technology", Informing Science Special Issue on Multimedia Informing Technologies-Part 2, Vol. 3, No. 1, 2000.