Face Recognition between Two Person using Kernel Principal Component Analysis and Support Vector Machines

Ivanna K. Timotius[1], Iwan Setyawan[2], and Andreas A. Febrianto[3]

Department of Electronic Engineering
Satya Wacana Christian University, Salatiga - Indonesia.
[1]ivanna_timotius@yahoo.com, [2]iwan.setyawan@ieee.org, [3]andreas@visio-lab.org

Abstract: Face recognition is gaining enormous interest nowadays. However, the technical challenges to “teach” a computer to recognize faces have been very difficult. Many methods and approaches have been proposed in the literature. This paper presents a face recognition method based on the combined kernel principal component analysis (KPCA) and support vector machine (SVM) methods. First, the KPCA method is utilized to extract features from the input images. The SVM method is then applied to these extracted features to classify the input images. We compare the performance of this face recognition method to other commonly-used methods. Our experiments show that the combination of KPCA and SVM achieves a higher performance compared to the nearest neighbor classifier, support vector machine, and the combination of kernel principal component analysis and nearest neighbor classifier.

Keywords: Face recognition, Kernel Principal Component Analysis, Support Vector Machines

1. Introduction

Face recognition is a study of how machines can recognize face, a task that humans perform naturally and effortlessly throughout their lives. Currently, powerful and low-cost computing systems (in the forms of, for example, desktop or embedded systems) are widely available. These systems are powerful enough to perform face recognition tasks and are cheap enough to make such implementation economically viable. It is only natural, then, that enormous interest in automatic face recognition of digital images and videos has risen recently. There are numerous areas in which an automatic face recognition system can be implemented. These include biometric authentication, surveillance, human-computer interaction, and multimedia management (for example, automatic tagging of a particular individual within a collection of digital photographs). The interest in face recognition application for biometric authentication, in particular, has risen rapidly mainly due to several factors [1]. The first one is the rising public concern for security especially in public facilities such as airports. The second factor is the rising need for verifiable identity over the internet due to the rise in internet-based e-commerce. And finally, the use of face recognition in biometric authentication is viewed to be one of the least intrusive methods while retaining a high level of accuracy. The technical challenges to “teach” a computer to recognize faces have been very difficult. Many methods and approaches have been proposed in the literature. Two of these approaches, namely the kernel principal component analysis (KPCA) and support vector machines (SVM) are the subject of this paper.

KPCA is a development of the PCA (eigenfaces) method [2]. PCA method is used in the feature extraction step of a face recognition system. Face image data has a very high dimensionality. Trying to perform classification on such a data would be extremely difficult. Therefore, we need to reduce the dimensionality of the data. The reduced-dimensionality data would contain only important features that we need in order to perform classification. PCA Performs very well in reducing the dimensionality of the data. However, the performance of PCA method (or other linear methods) is not completely satisfactory for problems with high

Received: December 14, 2009. Accepted: February 23, 2010

nonlinearity, such as face recognition. Basically, the KPCA is used to tackle the nonlinearity of face recognition problem. By using a nonlinear kernel function, a dimensional reduction (i.e., nonlinear projection) is performed. The images are first transformed from image space into a feature space. In the feature space, the manifolds of the data become simple.

SVM is another method widely used in pattern recognition, including face recognition. SVM is a set of related supervised learning method used in the feature classification step of a face recognition system. It belongs to a family of generalized linear classifiers. The main idea of a SVM is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. The separating hyperplane is defined as a linear function drawn in the feature space [3]. Ideally, however, the hyperplane should not be linear in order to achieve better performance. By using kernel functions, the scalar product can be implicitly computed in a kernel feature space, without explicitly using or even knowing the mapping [4].

This paper presents a face recognition method based on the combined KPCA and SVM methods. The KPCA method is used to extract the important features of the data. The resulting features are then classified using the SVM method. By combining these two algorithms, we expect a significant performance gain. This is because the output of the KPCA algorithm is the principal components of the face images. These principal components are then used as the input of the SVM. Since these principal components are the most important part of the data from a classification point of view, we can expect an increase of the performance of the SVM. The block diagram of our system is presented in Figure1.

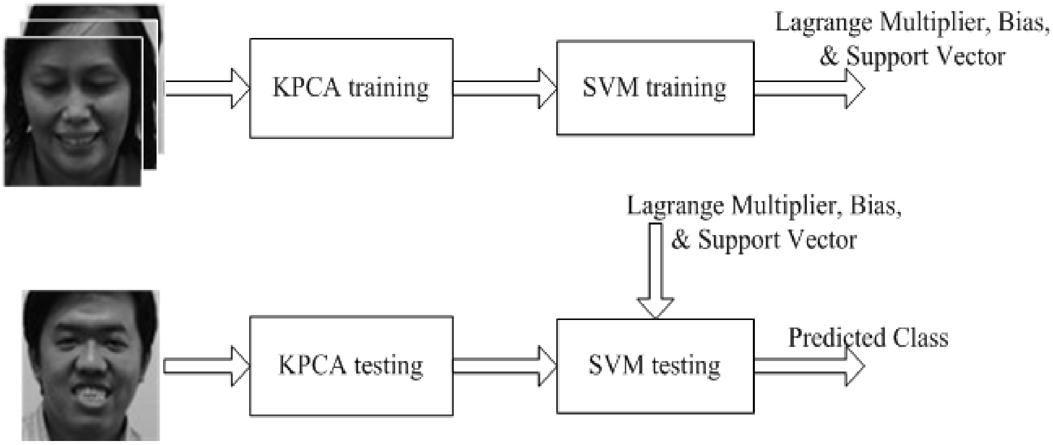

Figure 1. Block diagram

We will compare the performance of this combined method to other commonly used methods of face recognition. We also introduce the VISIO multi-view face database that we use to perform these tests.

In literatures, there are some works related to the work presented in this paper. For example, in Ref. [5], the authors used a multi-view face database to compare the performance of several face recognition algorithms, namely PCA (Principal Component Analysis) and ICA (Independent Component Analysis). They showed that the PCA and ICA algorithms gave similar performances, with ICA giving a slightly better result (around 95% success rate compared to around 93% success rate for PCA). Another related work is published in Ref. [6]. The authors of this paper compared the performance of several algorithms (including PCA and 2DLPGNN (2D log-polar Gabor transform and Neural Network)) in recognizing subjects with various facial accessories ("disguises"). Their result shows that for faces with multiple disguises the 2DLPGNN algorithm gives the best result (approximately 65% success rate). The authors of the paper also performed a test using face images with pose, expression, and illumination variations as a verification of the performance of the algorithms. It is interesting to note that in this test the performance of PCA they reported is very poor (around 35%). The 2DLPGNN algorithm again gave the best performance (around 85% success rate). Based on these two preceding examples, we can conclude that from algorithm point of view, our work presented in this paper presents a different combinations of algorithms not yet explored in the previous works.

The rest of this paper is organized as follows. In Sections 2 and 3 we discuss the theory and implementation of KPCA and SVM methods, respectively. In Section 4 we will discuss the details of our experiments and the details of the face database used in this paper. Finally, in Section 5 we will present some conclusions and pointers to our future works.

2. Kernel Principal Component Analysis

Component analysis aims to find the "right" features from data without supervision [7]. Principal component analysis (PCA) is used to lower the dimensional space of the feature. By doing this, we hope to reduce the time complexities. In the process of lowering the dimensional space, we hope to extract the principal structure of our data sets. PCA can be seen as an orthogonal transformation to a coordinate system that describes the data. The basis of our new coordinate is called principal components. In PCA, these principal components are estimated by solving an eigenvalue problem. In many cases a small number of principal components are adequate to describe most of the structure in the data set.

By using a nonlinear mapping, the data set can be mapped into a higher dimensional feature space *F*. The representation of features in this high dimensional feature space helps the classifier to perform better. Fortunately, for certain feature spaces *F* there is a function for computing scalar products in feature spaces [4]. This is known as a kernel function. By using a kernel function, every linear algorithm that uses scalar products can be implicitly executed in *F* without explicitly knowing the mapping Φ, constructing a nonlinear version of a linear algorithm. The nonlinear version of PCA that is constructed by using a kernel function is known as kernel principal component analysis (KPCA).

Given a set of centered data:

$$\mathbf{x}_k \in \mathbf{R}^N \qquad \sum_{k=1}^{M} \Phi(\mathbf{x}_k) = \mathbf{0} \tag{1}$$

PCA diagonalized the covariance matrix in *F* [8]:

$$C = \frac{1}{M} \sum_{j=1}^{M} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^T \tag{2}$$

To do this, we have to find the eigenvalues $\lambda \geq 0$ and eigenvectors satisfying

$$\lambda \mathbf{v} = C\mathbf{v} \tag{3}$$

All solutions v with $\lambda \neq 0$ must lie in the span of Φ (x$_1$), Φ (x$_2$), … , Φ (x$_M$). Hence Eq. 2 is equivalent to

$$\lambda(\Phi(\mathbf{x}_k) \cdot \mathbf{v}) = (\Phi(\mathbf{x}_k) \cdot C\mathbf{v}) \text{ for } k = 1, 2, \dots, M \tag{4}$$

Since v lie in the span of Φ (x$_1$), Φ (x$_2$), … , Φ (x$_M$), there exist coefficients α_i (i = 1, 2, …, M) such that,

$$\mathbf{v} = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{x}_i) \tag{5}$$

Combining Eq. 4 and Eq. 5, we get

$$\lambda \sum_{i=1}^{M} \alpha_i \left(\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i) \right) = \frac{1}{M} \sum_{i=1}^{M} \alpha_i \left(\Phi(\mathbf{x}_k) \cdot \sum_{j=1}^{M} \Phi(\mathbf{x}_j) \right) \left(\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) \right) \qquad \forall k \tag{6}$$

By defining the kernel function as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{7}$$

and the elements of M x M matrix K by

$$K_{ij} := \left(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \right) \tag{8}$$

we get

$$M \lambda K \boldsymbol{\alpha} = K^2 \boldsymbol{\alpha} \tag{9}$$

where α denotes the column vector with entries α_1, α_2, … , α_M. To find the solution, we solve the eigenvalue problem

$$M \lambda \boldsymbol{\alpha} = K \boldsymbol{\alpha} \tag{10}$$

For the principal component extraction, the projections onto the eigenvectors v^k in F are needed. Given x as a test point and Φ(x) is its image in F, then its nonlinear principal components is

$$\left(\mathbf{v}^k \cdot \Phi(\mathbf{x}) \right) = \sum_{i=1}^{M} \alpha_i^k \left(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) \right) \tag{11}$$

The steps to compute the principal components can be summarized as:

1. Compute the matrix K,
2. Compute its eigenvectors and normalize them in F,
3. Compute projections of a test point onto the eigenvectors.

This paper uses the radial-basis function network as a kernel function:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\tfrac{1}{2s^2} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \right) \tag{12}$$

where s is the width which is specified empirically.

In this paper, $x_k \in R^N$ are taken from the face images by rearranging the pixel value order as shown in Figure2.

$$\Rightarrow \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \Rightarrow \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \\ x_{21} \\ \vdots \\ x_{mn} \end{pmatrix}$$

Figure 2. Pre-processing step

3. Support Vector Machines

SVM is a supervised classification method which utilizes optimalization approach to construct a hyperplane. For pattern recognition tasks, the construction of an optimal hyperplane in SVM is done in a high-dimensional feature space obtained from a nonlinear mapping. Given a training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where x$_i$ is a training vector and y_i is its class label being either +1 or -1, SVM wants to find the weight vector w and the bias b of the separating hyperplane such that [3][9]:

$$\begin{aligned} y_i(\mathbf{w}^T\varphi(\mathbf{x}_i)+b) &\geq 1-\xi_i, \quad \forall i \\ \xi_i &\geq 0, \quad \forall i \end{aligned} \tag{13}$$

with w and the slack variables ξ_i minimizing the cost function:

$$\Phi(\mathbf{w},\xi_i) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{14}$$

where the slack variables ξ_i represent the error measures of data, C is a user-specified positive parameter (which is the penalty assigned to the errors), and ϕ (.) is a nonlinear mapping which maps the data into a higher dimensional feature space from original input space.

The dual problem of SVM is given as follows. Find the Lagrange multipliers $\{\alpha_i\}_{i=1}^{N}$ that maximize the objective function:

$$Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)$$

subject to

$$\begin{aligned} &\sum_{i=1}^{N}\alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C \qquad \forall i \end{aligned} \tag{15}$$

where C is a user-specified positive parameter. If $0 < \alpha_i \leq C$, the corresponding data points are called support vectors.

Having the Lagrange multipliers, the optimum weight vector w$_o$ could be computed by

$$\mathbf{w}_o = \sum_{i=1}^{N}\alpha_i y_i \varphi(\mathbf{x}_i) \tag{16}$$

From the Lagrangian function, according to Kuhn-Tucker [10], by taking the samples with $0 < \alpha_i < C$, the bias could be calculated by

$$b = \frac{1}{\#SV}\sum_{x_i \in SV}\left(\frac{1}{y_i} - \sum_{\mathbf{x}_j \in SV}\alpha_j y_j k(\mathbf{x}_j,\mathbf{x}_i)\right) \tag{17}$$

where #SV is the number of support vectors with $0 < \alpha_i < C$.

For an unseen data z, its predicted class can be obtained by the decision function:

$$\begin{aligned} D(\mathbf{z}) &= \operatorname{sign}\left(\sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{z}) + b\right) \\ &= \operatorname{sign}\left(\sum_{i=1}^{N_S} \alpha_i y_i k(\mathbf{x}_i, \mathbf{z}) + b\right) \end{aligned} \tag{18}$$

where N_s is the number of support vectors.

The requirement on the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)$ is to satisfy Mercer's theorem. The kernel used in SVM in this paper is radial-basis function network similar to Eq. 12:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\tfrac{1}{2s_1^2}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right) \tag{19}$$

where s_1 is the width which is specified by the user. However, the width used in KPCA and SVM does not have to be the same.

4. Experiments and Results

The experiment was done by using the Video, Image and Signal Processing (VISIO) laboratory Satya Wacana Christian University (SWCU) multiview face database. This database currently contains face images of 100 subjects. The subjects are evenly distributed in gender (i.e., 50 subjects are female and 50 subjects are male). The age varies between 19 and 69 years. The images are taken under controlled condition in our laboratory. Each subject is photographed against a uniform white background using a single camera and identical settings. We used a single fluorescent light source that is placed in front of the subject. The camera's automatic white balance is used for all subjects. For each subject, we take 105 photographs. Each photograph has a different combination of viewpoint, facial expression, and accessories. These variations are summarized in Table 1. In total, the VISIO multiview face database contains 10500 different images. Each image in the VISIO multiview face database is manually cropped around the facial area. Then the images are resampled into a 64 × 64 pixel 8-bit grayscale image. Figure3 shows the image variations for one subject.

We shall now give a brief comparison of the VISIO multiview face database to other face databases found in the literature. One of the well known face databases is FERET (Facial Recognition Technology) database [11]. In FERET database, the face images of every subject was taken in multiple head pose angles and different face expressions, similar to the VISIO database. However, most of the subjects in FERET database are Caucasians. This database is therefore not very suitable to be used as training and testing database for a system to be implemented in Indonesia. To have a more reliable database, we develop VISIO face database whose subjects come from typical ethnic groups in Indonesia. Another face database is the one used in Refs. [5] and [6]. The face database in [5] consists of only 10 subjects. Furthermore, the variations used in the database consist of only angle (5 different angles, namely frontal, right and left profiles, and right and left quarters) and expression. The VISIO multiview face database has more subjects (100 compared to 10) and variations compared to the database used in the first example (a total of 10500 different face images in contrast to 3600 in [5]). Furthermore the database used in [5] does not include facial accessories, while the one we use does. The database used in [6] consists of 125 different subjects. Each subject is photographed only from one angle (frontal). However, the subjects have extensive variations of disguises applied (both real and synthetic). Compared to this database, our database has much less subjects and "disguises". However, our database does have more viewpoint and expression variations.

Table 1. Variations in the VISIO Multiview Face Database

Variations	
Viewpoint	Level, low, high, frontal (0°), right 45°, right 90°, left 45°, left 90°
Accessories	Glasses, Cap
Facial expression	Smiling, laughing, neutral

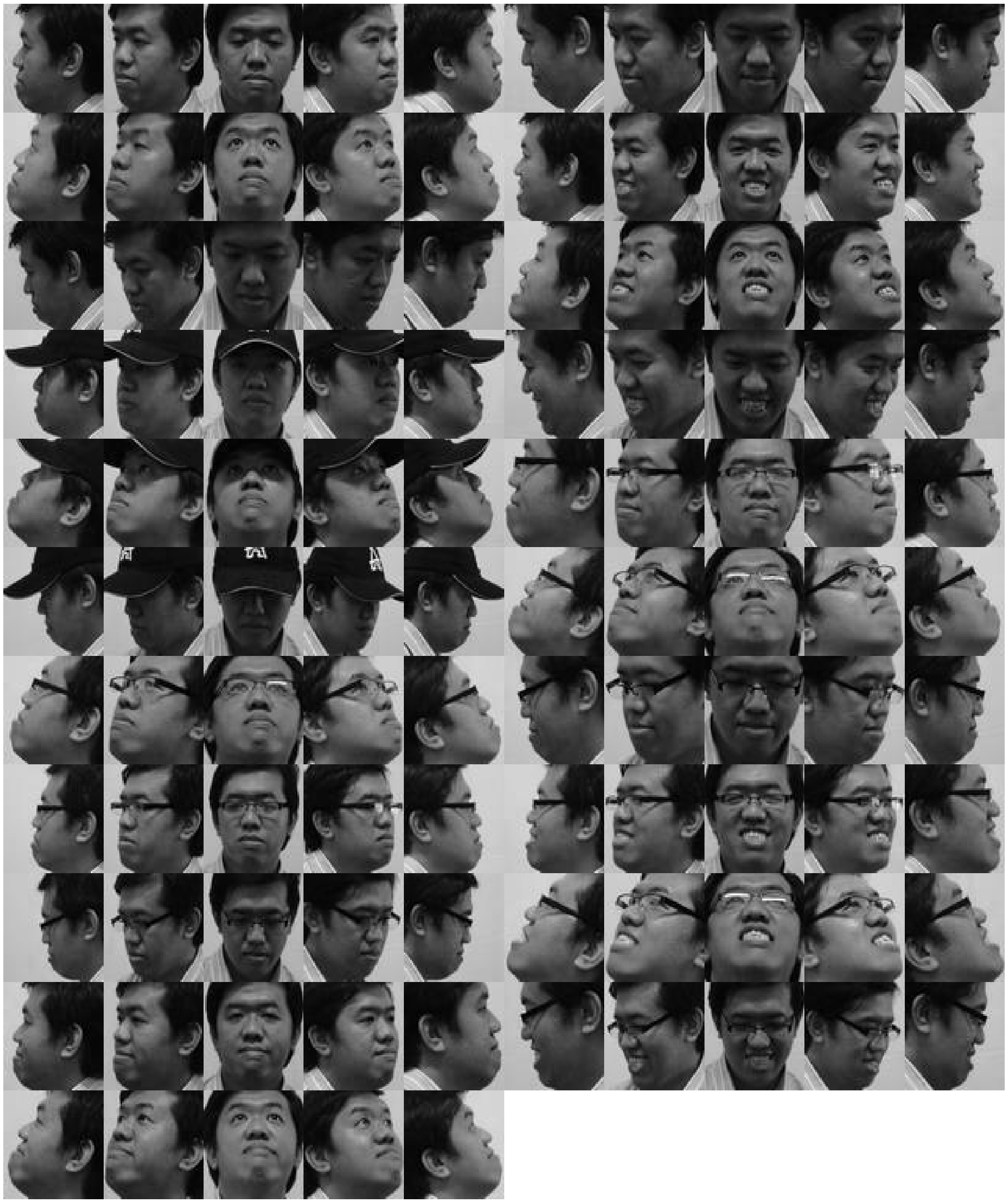

Figure 3. Image variation of one subject in VISIO multiview face database.

The experiment is designed to classify two different subjects and is done by using 10 runs of 2-fold cross validation. For each experiment, the parameters for KPCA and SVM are determined empirically. The classification rate average by using the combination of KPCA and SVM is compared with other commonly used methods in face recognition, namely the nearest neighbor (NN), the SVM (without using KPCA as feature extractor), and the combination of KPCA (as feature extractor) and NN as classifier. The parameters, classification rate averages, training time (in second) and testing time (in millisecond) are summarized in Table 2. Every algorithm used 105 images in the training phase. The testing time is calculated for each image. Since in the training phase NN only store all training images, the training time is approximately zero.

Table 2. Classification Rate Averages

Method	**Parameters**	**Classification Rate Average**	**Training Time (s)**	**Testing Time (ms)**
NN	-	94.95%	0	10
SVM	*SVM*: $s_1 = 400$, $C = 0.006$	95.04%	1.28	10
KPCA – NN	*KPCA*: $s = 50000$ number of eigen vectors = 11	97.14%	1.56	10
KPCA – SVM	*KPCA*: $s = 50000$ number of eigen vectors = 11 *SVM*: $s_1 = 0.07$, $C = 500000$	99.05%	2.00	10

Table 2 shows that the KPCA-NN method performs significantly better than NN. The combination of KPCA and NN methods even shows better result than the SVM method (without using KPCA as feature extractor), although the NN method is a linear classifier. The SVM with kernel function performs significantly better than NN. By using the combination of KPCA and SVM the system can achieve a much higher performance compared to the use of the other methods.

The training and testing time of these algorithms have no significant difference. This is due to the fact that we only use a few images (105 images) in the training phase and one image to calculate the testing time.

The most frequently misclassified images are shown in Fig 4. Misclassification happens mostly because of the similarity with the other images or the accessories worn by the subject, especially the hat.

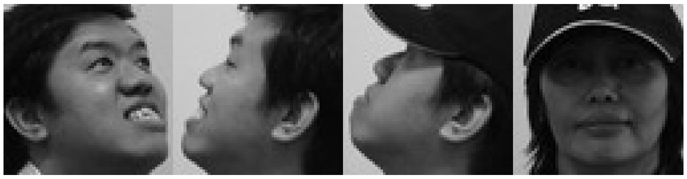

Figure 4. Most misclassified images

5. Conclusions and Future Works

Our experimental results can be summarized as follows. The use of KPCA as a feature extractor prior to the classification process significantly increases the performance of the system. The combination of KPCA and SVM methods shows a very satisfactory result.

In the future, we plan to continue our research by expanding the number of subjects used in the experiments, investigating other possible methods (both feature extractors and classifiers) to achieve a better system performance, and combining our face recognition method with a face detection method to form a complete face recognition system.

Acknowledgement

This work is supported by DP2M DIKTI under the research grant 383/SP2H/PP/DP2M/VI/2009.

Reference

[1] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, Springer Science+Business Media, Inc., 2005.

A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical Pattern Recognition: A Review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, 2000.

[2] S. Haykin, *Neural Network: A Comprehensive Foundation*, Prentice-Hall, New Jersey, 1999.

[3] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, ”An Introduction to Kernel-Based Learning Algorithms,” *IEEE Transactions on Neural Networks*, vol. 12, no. 2, 2001.

[4] D. Ginhac, F. Yang, X. Liu, and J. Dang, “Robust Face Recognition System Based on a Multi-Views Face Database”, in *Recent Advances in Face Recognition (K. Delac, et.al., Eds)*, In-Tech, Croatia, 2008.

[5] R. Singh, M. Vatsa, and A. Noore, “Recognizing Face Images with Disguise Variations”, in *Recent Advances in Face Recognition (K. Delac, et.al., Eds)* , In-Tech, Croatia, 2008.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley, New York, 2001.

[7] B. B. Scholkopf and A. Smola, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” *Neural Computation*, vol. 10, pp. 1299-1319, 1998.

[8] V. Vapnik, *Statistical learning theory*, Springer, Berlin Heidelberg, New York, 1998.

[9] R. Fletcher, *Practical Methods of Optimization*, New York, Wiley, 1987.

[10] http://www.itl.nist.gov/iad/humanid/feret/feret_master.html, The Facial Recognition Technology (FERET) Database.

Ivanna K. Timotius received her bachelors degree in Electronic Engineering from Satya Wacana Christian University, Indonesia in 2003. She received her master degree from Chung Yuan Christian University, Taiwan in 2009. She is currently a lecturer in department of Electronic Engineering, Satya Wacana Christian University, Indonesia. Her research interest includes pattern recognition and digital image processing.

Iwan Setyawan received his Bachelor's and Master's degrees in 1996 and 1999, respectively, from the Department of Electrical Engineering, Bandung Institute of Technology, Indonesia. He received his Ph.D. degree from the Department of Electrical Engineering, Delft University of Technology, The Netherlands. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering, Satya Wacana Christian University, Indonesia. His current research interest includes digital image and video compression and watermarking.

Andreas Ardian Febrianto received his bachelors degree in Electronic Engineering from Satya Wacana Christian University, Indonesia in 1999. He received his master degree from Gadjah Mada University, Indonesia in 2003. He is currently a lecturer in department of Electronic Engineering, Satya Wacana Christian University, Indonesia since 2003. His research interest is in digital signal processing.