TWO-CLASS CLASSIFICATION WITH VARIOUS CHARACTERISTICS BASED ON KERNEL PRINCIPAL COMPONENT ANALYSIS AND SUPPORT VECTOR MACHINES

Ivanna Kristianti Timotius*), Iwan Setyawan, and Andreas Ardian Febrianto

Department of Electronic Engineering, Satya Wacana Christian University, Salatiga 50711, Indonesia

*)*E-mail: ivanna_timotius@yahoo.com*

Abstract

Two class pattern classification problems appeared in many applications. In some applications, the characteristic of the members in a class is dissimilar. This paper proposed a classification system for this problem. The proposed system was developed based on the combination of kernel principal component analysis (KPCA) and support vector machines (SVMs). This system has been implemented in a two class face recognition problem. The average of the classification rate in this face image classification is 82.5%.

Keywords: *characteristic, classification, face recognition, kernel principal component analysis, support vector machines*

1. Introduction

Machine pattern recognition is the study of how machine take raw data and makes a decision based on the category of the pattern [1]. A machine that can recognize pattern such as automated speech recognition, fingerprint identification, optical character recognition, face recognition and much more is very useful in our daily lives. One of the areas in pattern recognition is pattern classification, which aims to assign an object into a category.

Basically pattern classification can be divided according to the number of category: two-class pattern classification and multi-class pattern classification. Some works focused on two-class pattern classification [2-3], and some works focused on multi-class pattern classification [4-7]. This paper focuses on the two-class pattern classification, because there are many real-life problems that use two-class pattern classifications as a solution. For example, in biometric authentication application it is often more interesting to focus on deciding whether a person is allowed to enter a certain area or not. In object detection application, it is interesting to decide whether a particular object exists or not. In the medical application, it is often more interesting to develop a machine that can decide whether a given data is normal or abnormal.

In some two-class pattern classification applications, there are specific characteristics in each class. For instance, in gender recognition based on the information from facial images, the features of each gender are mostly obviously visible in an image. In face recognition between two persons [2] the features of each person are obviously different. However, in some two-class pattern classification applications, there are various characteristics in a class. For example, in a group classification based on face recognition, each member of the group has different face characteristics. Another example is in the abnormality detection from medical data, in which there are various possible causes for the abnormality, with different characteristics.

This paper proposes a classification system for group classification with various characteristics. The experiment is performed by using the face images from two different groups. To recognize the face images, we employ kernel principal component analysis (KPCA) and support vector machines (SVMs) methods. Each of the KPCA machines is used as a feature extractor. The extracted features are then classified using the SVM method.

Section 2 and 3 of this paper will discuss the theory and implementation of KPCA and SVMs methods, respectively. Section 4 will present the proposed system designed for a two-class classification with various characteristics. Section 5 will explain the details of our experiments. Finally, in section 6 the conclusions of this work are presented.

2. Experiments

Kernel Principal Component Analysis. Principal Component Analysis (PCA, eigenfaces) can be seen as an orthogonal transformation to a coordinate system that describe of data. The basis of the 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 is adequate to describe most of the structure in the data set.

KPCA is a development of the PCA method [8]. This method uses a nonlinear mapping Φ to map data into a higher dimensional feature space. For certain feature spaces there is a function for computing scalar products in feature spaces [9] that is known as kernel function. By using this kernel function, it is possible to construct a nonlinear version of a linear algorithm.

The method starts with 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 diagonalize the covariance matrix in the feature space F [9]:

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

In order to diagonalize C, we have to find the eigenvalues $\lambda \geq 0$ and eigenvectors satisfying:

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

The eigenvectors **v** with $\lambda \neq 0$ must lie in the span of $\Phi(\mathbf{x}_1)$, $\Phi(\mathbf{x}_2)$, … , $\Phi(\mathbf{x}_M)$. Hence,

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

Since **v** lie in the span of $\Phi(\mathbf{x}_1)$, $\Phi(\mathbf{x}_2)$, … , $\Phi(\mathbf{x}_M)$, there exist coefficients α_i (i = 1, 2, …, M) such that,

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

By combining the Eq. 4 and Eq. 5:

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

The kernel function is defined as

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

and the elements of $M \times M$ matrix K as

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

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

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

The projections onto the eigenvectors $\mathbf{v}^k$ in F are needed for the principal component extraction. Given **x** as a test point and $\Phi(\mathbf{x})$ 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 are:
Compute the matrix K,
Compute its eigenvectors and normalize them in F,
Compute projections of a test point onto the eigenvectors.

The radial-basis function network is used as a kernel function in this paper:

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

where s is the width which is specified empirically.

Support Vector Machines. SVM is a method widely used in pattern recognition, including face recognition. The main idea of a support vector machine 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 by utilizing optimalization approach. The separating hyperplane is defined as a linear function drawn in the feature space [10]. 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 [9].

For pattern recognition task, the method starts with a training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ is a training vector and y_i is its class label being either +1 or -1 (member or non-member), SVM aims to find the optimum weight vector *w* and the bias *b* of the separating hyperplane such that [10-11]:

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

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 the penalty assigned to the errors, and $\varphi(\cdot)$ is a nonlinear mapping which maps the data into a higher dimensional feature space.

By finding the Lagrange multipliers $\{\alpha_i\}_{i=1}^{N}$ that maximize the objective function, the dual problem is given as follows:

$$\begin{aligned} &Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)\\ &\sum_{i=1}^{N}\alpha_i y_i=0\\ &0\leq\alpha_i\leq C \qquad \forall i \end{aligned} \tag{15}$$

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 $\mathbf{w}_o$ could be computed by:

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

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:

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

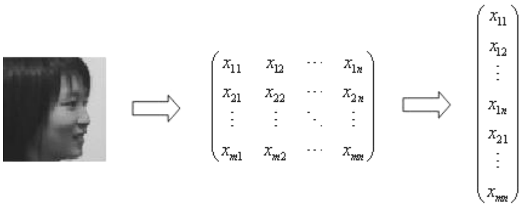
where N_s is the number of support vectors. The kernel function used in SVM in this paper is radial-basis function network similar to Eq. 12:

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

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

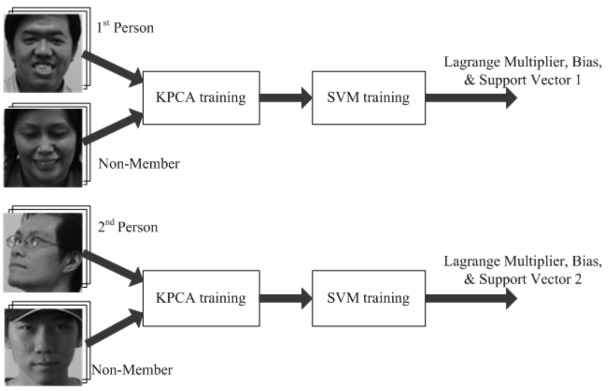
Proposed System. The proposed system starts with a preprocessing step which is constructing a column vector from all the input images. This process is performed by concatenating the columns of the input image. This process is shown in Fig. 1.

After the images are preprocessed, the images go through a cascade system as shown in Fig. 2 and Fig. 3. Fig. 2 shows the block diagram of the training phase. Each machine in the training phase produces Lagrange



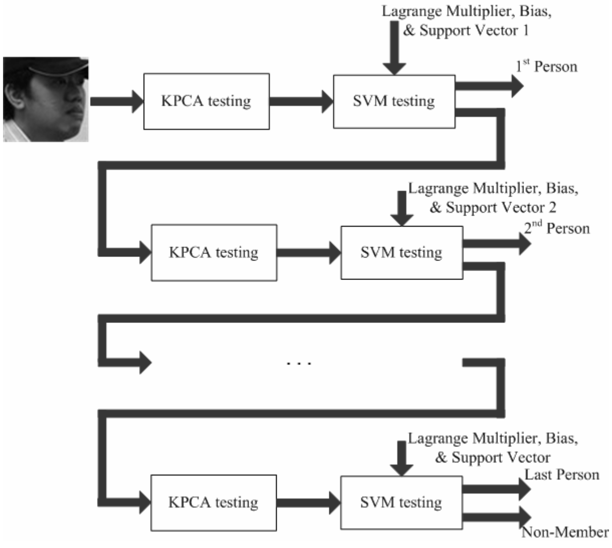
x_{11} x_{12} ⋯ x_{1n}
x_{21} x_{22} ⋯ x_{2n}
x_{m1} x_{m2} ⋯ x_{mn}
x_{11} x_{12} ⋮ x_{1n} x_{21} ⋮ x_{mn}

Figure 1. Pre-processing Step



1st Person
Non-Member
KPCA training
SVM training
Lagrange Multiplier, Bias, & Support Vector 1
2nd Person
Non-Member
KPCA training
SVM training
Lagrange Multiplier, Bias, & Support Vector 2

Figure 2. Proposed Training Phase



Lagrange Multiplier, Bias, & Support Vector 1
KPCA testing
SVM testing
1st Person
Lagrange Multiplier, Bias, & Support Vector 2
KPCA testing
SVM testing
2nd Person
. . .
Lagrange Multiplier, Bias, & Support Vector
KPCA testing
SVM testing
Last Person
Non-Member

Figure 3. Proposed Testing Phase

multipliers, bias and support vectors for each member. Fig. 3 shows the block diagram of the testing phase.

Experiments. The experiment was performed using the Video Image and Signal Processing (VISIO) laboratory of the Satya Wacana Christian University (SWCU) multiview face database. The subjects in this database are evenly distributed in gender. 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. We use the automatic white balance setting of the camera in our experiments. For each subject, the VISIO face database contains 105 face images with variations of viewpoint, facial expression, and facial accessories. Each image in the VISIO multi-view face database is manually cropped around the facial area. Then the images are resampled into a 64 × 64 pixel 8-bit grayscale images.

The experiment is designed to classify 30 different subjects into two groups (6 subjects are assigned as members and 24 subjects are assigned as non-member). The 30 subjects used in this experiment are shown in Fig. 4. The experiment is done by using 2-fold cross validation. For each experiment, the parameters for KPCA and SVM are determined empirically. In our experiments, we used identical parameters for KPCA and SVM for each stage of the cascade system.

3. Results and Discussion

Several parameter settings of this combined algorithm were tried by using 1 run of 2-fold cross validation. The results of the experiments using these parameter settings are given in Table 1. From this table, we can see that the best classification rate average is achieved by setting the KPCA parameters into s^2 = 100000 and number of eigenvalue = 40, and the SVM parameters into s_1^2 = 0.05 and C = 30. This parameter combination yields a classification rate of 82.48%, with a standard deviation of 0.13%.

In order to further verify the validity of our results, we also performed an experiment using 10 runs of 2-fold cross validation. In each 2-fold cross validation run the training and testing sets are determined randomly. In other words, each run use different training and testing sets. Therefore, by performing more runs, we will get a more representative result. In this second experiment, we use the parameter set that gives the best result as described above. The 10 runs of 2-fold cross validation gives a classification rate average of 82.51% with a standard deviation of 0.37%. This result is consistent with the one presented in Table 1. The increase of the standard deviation is understandable since we are introducing more variation to the training and testing data sets.

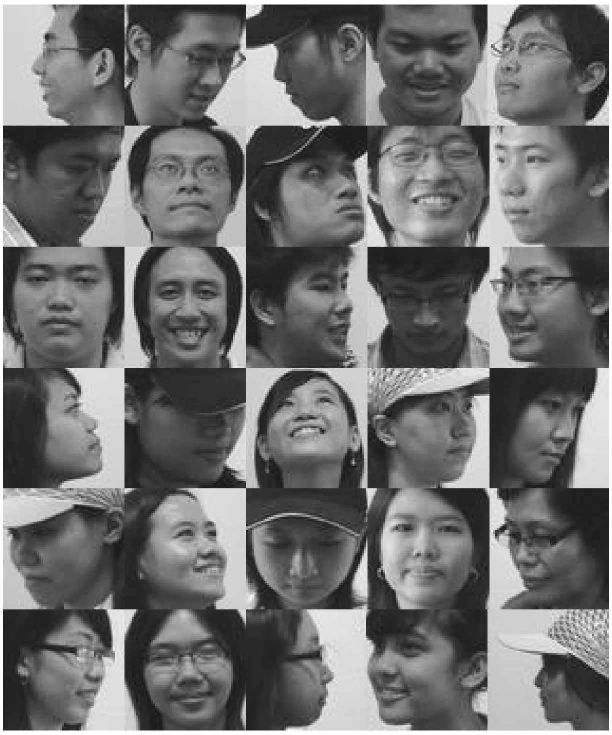


Figure 4. 30 Subjects Used in the Experiment

Table 1. Experimental Results

KPCA parameters		SVM parameters		Results	
s^2 ($\times 10^5$)	#ev	s_1^2	C	avg (%)	std (%)
1	11	0.01	10	80.95	0.05
1	11	0.05	50	81.21	0.13
1	100	0.05	50	82.03	0.40
1	200	0.05	50	82.03	0.13
1	100	0.1	100	81.65	0.58
1	50	0.05	50	82.32	0.09
1	25	0.05	50	82.22	0.94
1	40	0.05	50	82.38	0.01
0.5	40	0.05	50	81.94	0.18
0.5	40	0.05	50	80.00	0.22
1	30	0.05	50	82.29	0.40
1	40	0.05	50	82.38	0.01
1	40	0.05	30	82.48	0.13
0.8	40	0.05	30	82.38	0.18
1.2	40	0.05	30	82.32	0.09

#ev = number of eigenvalue
avg = average of classification rate
std = standard deviation

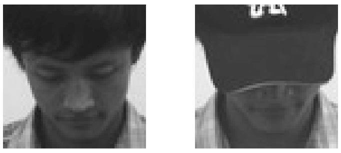


Figure 5. Effect of Facial Accessories

The results also show that there are still some misclassifications that occurred during the experiments. There are two reasons for this. The first reason is that some of the data are projected into points that lie very close to each other in the eigenvector space. Therefore, the SVM has difficulties in classifying these data points. The second reason is the presence of facial accessories, in particular the hat. This accessory obscures much of the facial area. This in turn hides some facial features necessary for correct data classification. This is particularly evident when the subject is not frontally facing the camera. This is shown in Figure 5.

4. Conclusion

Our experimental results shows that the classification rate average of the proposed system can reach 82.48%. We can therefore conclude that the cascade system is suitable for application in group classification problem with various characteristics.

In the future, we will try to develop this method further and investigate other possible methods to have a higher classification rate. In particular, we would like to investigate the use of different KPCA and SVM parameters for each stage of the cascade system.

Acknowledgment

This work has been funded by the Higher Education Directorate of the Ministry of Education under grant number: 213/SP2H/PP/DP2M/III/2010.

References

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd ed., Wiley, New York, 2001, p.635.

[2] Timotius, I. K., Setyawan, I. & Febrianto, A. A., Int. J. Electr. Eng. Inf. 2/1 (2010) 53.

[3] E. Makinen, R. Raisamo, IEEE Trans. Pattern Anal. Mach. Intell. 30/3 (2008) 541.

[4] Z. Li, X. Tang, IEEE Trans. Inf. Forensics Secur. 2/2 (2007) 174.

[5] I. Kotsia, I. Pitas, IEEE Trans. Image Process. 16/1 (2007) 172.

[6] M. Gonen, A.G. Tanugur, E. Alpaydın, IEEE Trans. Neural Networks, 19/1 (2008) 130.

[7] Z. Rustam, B. Widjaja, B. Kusumoputro, J. Makara Seri Sains 7/3 (2003) 15.

[8] A.K. Jain, R.P. W. Duin, J. Mao, IEEE Trans. Pattern Anal. Mach. Intell. 22/1 (2000) 4.

[9] K.R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, IEEE Trans. Neural Networks. 12/2 (2001) 181.

[10] S. Haykin, Neural Network: A Comprehensive Foundation, Prentice-Hall, New Jersey, USA, 2008, p.936.

[11] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998, p.736.