Face Recognition with Kernel Principal Component Analysis and Support Vector Machine

by Dewi Yanti Liliana

Face Recognition with Kernel Principal Component Analysis and Support Vector Machine

1st Dewi Yanti Liliana
Department of Informatics and Computer
State Polytechnic of Jakarta
Depok, Indonesia
dewiyanti.liliana@tik.pnj.ac.id

2nd I Made Agus Setiawan
Udayana University, Denpasar, Indonesia
University of Pittsburgh, USA
ims13@pitt.edu

***Abstract*— Machine learning and pattern recognition recently become a hot topic in computing world. This is due to the fast-growing of resources as well as techniques that make it easier to solve machine learning and pattern recognition problems. Problems that require machine learning solutions may be very simple for humans but actually can be very complex for machines to solve them. Face recognition is amongst those problems. Almost all human can easily recognize others without require specific knowledge to do it, different from machines which require its. This paper discussed face recognition task using machine learning strategies which involved Kernel Principal Component Analysis (KPCA) and Support Vector Machine (SVM) to identify person. KPCA extracted features from 2D image input and produced the important features of an image input. The extracted face features are recognized by SVM by classifying human face according to their stored identity in a database. SVM, which was basically a binary classifier worked by using one-against-one strategy to compare the face feature vector of a single test image to the stored face image in a face image database. Experiment results on grayscale images with size 92x112 pixels gave 96.25% of accuracy rate. Hence, KPCA and SVM for face recognition is a robust machine learning method.**

Keywords— face recognition, kernel eigenface, kernel PCA, machine learning, pattern recognition, support vector machine

I. Introduction

Face recognition involves the activity of identifying face characteristics that differ one person from another person. The human face consists of part of the head area which is composed of chin, nose, mouth, cheeks, eyes and forehead. Human face images which is captured from camera devices can be used for many purposes. The traditional utilization of face recognition tools is for security and attendance systems, but today it has a greater implementation in retail, banking, hospital, restaurants, marketing, and health systems [1]. In machine learning, face recognition stages consist of a sequence of three activities: face detection; face feature extraction; face recognition. Each stage requires specific techniques. Face detection is done by segmentation process to separate face with other objects [2]. The significant issue in face feature extraction stage is in selecting the appropriate feature that describe human's face. In recognition stage, the system classifies an image face by finding the highest level of similarity to another image faces with certain labels or person identities through training and testing procedures. This is where the 'term' learning in machine learning comes from. By providing training face images to learn about the model, and then predict the testing images labels. A valid and satisfiable face features are very crucial to determine the success of face recognition.

Face recognition has been trained using several methods such as PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) [3]. PCA and LDA methods were both using linear proximity feature. But, in the case of face images with non-linear features, it reduced the recognition accuracy rate. The use of linear methods such as PCA and LDA usually extended the approach to search for eigen value and eigen vector to determine the principal component contained in human face images. The facial image feature was obtained by multiplying a number of principal components with the original face image. The recognition was based on the closest distance between two face image features. This method was less effective in overcoming non-linear image problems caused by the contents of the image, such as background changes, face position (orientation), lighting, and facial expressions [4]. Face recognition using PCA got a result with accuracy rate 78.3% [3].

Non-linear problems can be overcome by adopting a kernel trick [5]. Kernel trick aims to transform high-dimensional-low-features data space to a high-dimensional feature space, so that more features can be taken from an image of a human face. In the recognition process, those features can be applied to the high dimensional feature space. The accuracy rate is obtained from methods that adopt the kernel trick; and it will be higher than those that do not use the kernel method because kernel can overcome the non-linear properties of the human face input image.

SVM is a machine learning classifier which works by forming the optimum separating hyperplane that linearly separate between two classes. SVM can be treated for multiclass classification by applying some strategies: one-against-one or one-against-many classification. SVM is a robust classifier in machine learning that has been implemented in many prediction and recognition problems.

Based on the above background, we propose a new approach capable of recognizing face by using Kernel PCA as feature extraction and SVM as recognition methods. First, the input face image is converted into grayscale mode. Second, the KPCA is used to extract the face features. Finally, face recognition is performed by SVM which has been trained using database of face and resulting a face model. The main contribution is in combining Kernel PCA and SVM for identifying biometric of human from face which has many implementation benefits. The next section discusses KPCA and SVM for face recognition, followed by the design of the proposed system. We also present the experiment scenario and discuss the result in the last part of the paper.

978-1-7281-2930-3/19/$31.00 ©2019 IEEE

II. Machine Learning for Face Recognition

In this section we discuss the machine learning techniques which are used in this proposed study: Kernel PCA and Support Vector Machine.

A. Kernel Trick

In machine learning, kernel trick is a method that uses a linear classification algorithm to solve nonlinear problems by attaching the input to a higher dimension space, thus creating a linear classifier in the new dimension space using nonlinear classifier with the original dimension space [6]. Kernel trick is used to transform the input that implements an algorithm which depends on the dot product operation between two vectors [6]. The dot product function is replaced by the kernel function, so that linear algorithms can be easily applied in non-linear algorithms. With kernel trick, the mapping function is never explicitly calculated, because the high dimensional space is possibly used on infinite dimensions.

Theoretically, every state that is continuous, symmetrical, semi-positive is definitely positive. The kernel function K (x, y) can be expressed as a dot product in high-dimensional space. The mercer theory itself is a generalization of each positive semi-definite matrix in the Gramian matrix set of vectors [3], [7]. Gram matrix is commonly used in machine learning applications as a representation of kernel functions [5]. Kernel PCA

KPCA (Kernel Principal Component Analysis) is the improvement of the traditional PCA method, which is also called as eigenface [7]. KPCA can handle the shortcomings of the PCA method, which is to overcome problems that are nonlinear. By applying a nonlinear kernel function, it can reduce dimensions to projecting nonlinear dimensions.

In PCA, the obtained principal component is derived from the estimated results of solving the eigenvalue problem. In many cases, the small number of main components results in a lack of describing most of the structure in the data set. With PCA it will reduce the dimensions of the face until leaving only a few important features for classification.

By using nonlinear mapping, the dataset can be mapped into a higher dimension, namely F feature space. Representation of features in the high-dimensional feature space helps better classification. But for clarity of feature space, F is a function for calculating scalar products in feature space [5]. This is called as a kernel function. By using the kernel function, each linear algorithm used in a scalar product can be calculated in F without knowing the mapping of KPCA Φ by constructing a nonlinear version of the linear algorithm. The nonlinear version of PCA is built with a kernel function called kernel principal component analysis (KPCA).

Given a set of M centralized data $x_k \in R^N$ that corresponds to the following Equation (1).

$$\sum_{k=1}^{M} \Phi(x_k) = 0 \quad (1)$$

The computation of kernel Φ with size MxM to become matrix K is by applying Equation (2):

$$K'_{ij} := \left(\Phi(x_i) \cdot \Phi(x_j)\right) \quad (2)$$

where i and j is the dimension of feature in matrix M. The next step is performing kernel matrix centering using Equation (3).

$$K_{ij} = K'_{ij} - \frac{1}{M}\sum_{m=1}^{M} O_{im}K'_{mj} - \frac{1}{M}\sum_{m=1}^{M} K'_{in}O_{nj} + \frac{1}{M^2}\sum_{m=1}^{M}\sum_{m=1}^{M} O_{im}K'_{mn}O_{nj} \quad (3)$$

Where O is a ones matrix with size MxM and all elements are 1. To get the KPCA feature, we must find the eigen value and eigen vectors of matrix K using the decomposition in Equation (4).

$$M\lambda\alpha = K\alpha \quad (4)$$

where λ is eigen value and α is eigen vector with $\alpha_1, \alpha_2, \ldots, \alpha_M$ are the corresponding eigen values. Eigen vector is being normalized using Equation (5).

$$\alpha^{(n)}.\alpha^{(n)} = \frac{1}{\lambda_n} \quad (5)$$

The next step is extracting principle component by projecting eigen vector to a feature space. Given x as a test point of a testing image $\Phi(x)$ in an F dimensional space. The non-linear principal analysis extraction is computed using Equation (6).

$$q^{(n)} = \sum_{i=1}^{M} \alpha_i^{(n)} \left(\Phi(x_i) \cdot \Phi(x)\right) \quad (6)$$

where $\alpha(n)$ is n eigen vector components, and $q(n)$ is the projection components, or the KPCA components as a result of feature extraction process.

B. Support Vector Machine

Support Vector Machine (SVM) is one of the supervised classification methods in Machine Learning [8]. The main purpose of this method is to build an Optimum Separated Hyperplane (OSH) which makes an optimum dividing function (linear function) that can be used for classification. Fig. 1 depicts an SVM.

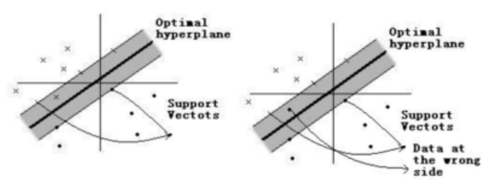

Fig. 1 SVM with optimal hyperplane

Fig. 1 shows the description of SVM with optimum hyperplane and maximum margin which separated two classes. In the case of pattern recognition, to form an optimum hyperplane in SVM, it must be done in a high dimensional feature space obtained from nonlinear mapping. In the following training example $\{(X_i, Y_i)\}_{i=1}^{N}$ where xi is the training vector and yi is the label class that is +1 or -1, in this case SVM is used to find the weight vector values (w) and bias (b) of the dividing hyperplane [6].

The SVM method can be used to classify data that can be separated linearly, semi-linearly, and data that cannot be linearly separated. For semi-linear or nonlinear data, slack variables are added to the optimization, as well as for nonlinear data using the kernel method. Face recognition is considered to be a nonlinear case. The following formula is used to look for hyperplane in general as given in Eq. (7).

$$y_i(W^T \varphi(x_i) + b) \geq 1 - \xi_i \quad (7)$$

where ξ_i is a slack variable and b is bias. Another derivation form of Equation (7) is called a primal Lagrange formula is shown in Equation (8).

$$L_P = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i\{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i \qquad (8)$$

The lagrange equation can only be solved if it satisfies certain conditions. Karush-Kuhn-Tucker condition is a way to optimize in nonlinear programming by fulfilling several conditions. With the KKT approach (Karush-Kuhn-Tucker) nonlinear inequality can be used in generalizing Lagrange multipliers that can use ordinary equations. The following equation shows $\xi_i = 0$ if $\alpha_i < C$ where C (Complexity) is gamma parameters, so each training data that is at a range $0 < \alpha_i < C$ that is also used counts b [11].

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \qquad (9)$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad 0 \le \alpha_i \le C, \forall i$$

Where C is a positive value user parameter, and $0 \le \alpha_i \le C, \forall i$ is a data point (α) or called as *support vector*. Therefore, we can get the optimum Lagrange multiplier by using weight vector which is computed using Equation (10).

$$W_0 = \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i) \qquad (10)$$

Where W_0 is the weight vector. Based on Lagrange KKT, by giving a sample $0 < \alpha_i < C$ bias can be computed using Equation (11).

$$b = \frac{1}{\#SV}\sum_{x_i \in SV}\left(\frac{1}{y_i} - \sum_{x_j \in SV} \alpha_j y_j k(x_j, x_i)\right) \qquad (11)$$

Where #SV is the number of support vectors $0 < \alpha_i < C$. For the unknown class data, we can predict the corresponding class by using Equation (12).

$$D(z) = sign\left(\sum_{i=1}^{N} \alpha_i y_i k(x_i, z) + b\right) = sign\left(\sum_{i=1}^{Ns} \alpha_i y_i k(x_i, z) + b\right) \qquad (12)$$

Where Ns is the support vectors. Kernel in a SVM in this study is using radial basis function which is applying Equation (13).

$$k(x_i, x_j) = \exp\left(-\frac{1}{2g^2}\|x_i - x_j\|^2\right) \qquad (13)$$

In KPCA and SVM can use the same kernel where the sigma value (g) can be different.

III. Human Face Recognition System

The methodology of the human face recognition system can be seen in Fig. 2. The system is divided into two stages: the training stage and the recognition stage. At the training stage, the system used training data on a number of human faces. The face used will only be sampled as many as 5 face images for each human subject. All datasets will be performed a feature extraction process to obtain image features that will be used at the classification stage. In previous face recognition studies, several methods have been used, such as PCA, LDA, Wavelet, and Gabor. In this system, KPCA feature extraction will be used.

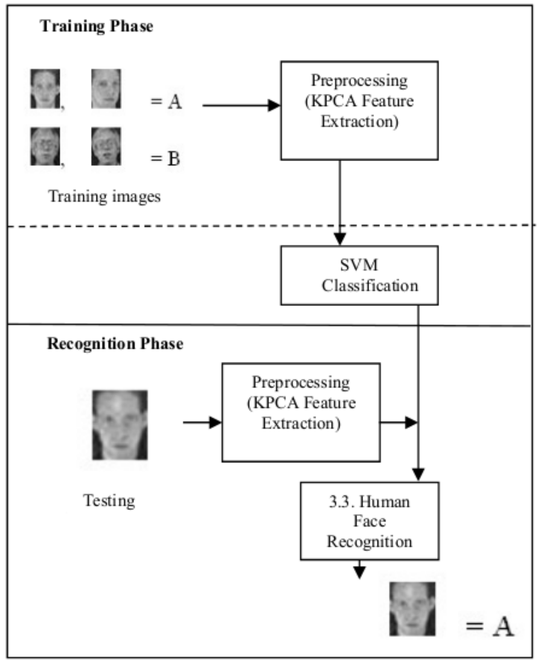

Fig. 2. Design of human face recognition system

The feature extraction process using KPCA is used at the next training stage to distinguish one human face from another human. In another studies, classification has been used with Neural Network and Bayesian methods [9]. This system used the SVM classifier to be able to classify faces from multiple classes (several different people).

In the recognition phase, the system has been trained by using human face training data, so the system is able to recognize faces of the same human with different expressions. This recognition process is the result of the voting process of a set of binary SVM classifiers, so the voted class is the largest voting value of the whole SVM classifier. The training and testing stages, including pre-processing (feature extraction with KPCA), classification with SVM, and human face recognition will be explained further in the next subsection.

A. (KPCA Feature Extraction)

In the facial feature extraction, the kernel eigenface method is used. The kernel eigenface / KPCA process is a modification of the PCA method with the kernel method. This is because the vector results in the kernel process have very high dimensions, sometimes even higher than the original image vector. KPCA can also be used to reduce the dimensions of the kernel to make it easier for the computing process.

At the stage of forming the kernel or mapping input vector image to high dimensional feature space using Equation (2), there is no need to explicitly map all image data to high dimensional. In the learning process, only dot product values are needed, so the kernel trick can be used to calculate the dot product value of image vectors, using the RBF kernel.

The next step is to form a covariance matrix that contains the kernel matrix centering value. The dimensions of this covariance matrix are the same as the K kernel matrix. The next step is to find the eigen value and eigen vector of the covariance matrix. The process of finding eigen value and eigen vector can be used by the decomposition method to solve the decomposition using Equation (4). Eigenvector that has been obtained must be normalized by equation (5).

The KPCA feature is obtained by projecting the input image vector from the eigenvector calculation results with Equation (6). It is used to synchronize the input dimensions and feature dimensions. This projection includes all datasets used in the system.

B. SVM Classification

This process will explain the stages in the training process with the SVM method. Each learning process is the result of the binary classification of a group of objects. The best SVM function is a function with the maximum margin value. Margin is the distance between the hyperplane and the support vector of each object in the training stage.

The SVM classification process starts with entering the results of the preprocessing in the form of KPCA features into the SVM. Next, determining two SVM parameters: gamma SVM for the SVM dot-product mapping process and complexity (*C*) for margin values tolerance in the hyperplane function. The *C* parameter is a constant for slack variable to measure the distance of misclassification, and epsilon and tolerance.

The process of forming a hyperplane function involves a number of k (k-1) / 2 classifier, where k is the number of class or human subjects in one-against-one strategy. Because at each time the SVM function running is a binary classifier that includes features from two different classes. The label class setting is positive class for the first class and negative class in the second class. Iteration is carried out as many as k (k-1) / 2 of the number of training classes, according to the one-against-one method, where k is the number of classes included in the training process. The one-against-one method was chosen because it has a simple decision-making class and the accuracy rate is higher than one-against-al [7].

In the training process, the Lagrange value is sought to find a global optimization of the value and number of support vectors used in learning the SVM function, which is the α value that meets the KKT conditions. After that, it calculates the bias value and weight to form the SVM hyperplane function. Finally, it forms the hyperplane function of support vector and SVM bias. This SVM training process can use the SMO (Sequential Minimal Optimization) algorithm which prioritizes global optimization in each process [7]. The results of this SVM training are a number of binary SVM classifiers. This classifier function can be used for data testing recognition processes.

C. Human Face Classification

Human face recognition is a classification task which classifies the human subject based on their class using SVM. it can be searched by looking for the highest voting value in the classification. Image testing classes can be obtained by weighting the largest sign function on a particular object.
The testing process is done on test data, then do the RBF kernel process on test data. Furthermore, it will be checked by the hyperplane (sign) function which has been previously extracted by the KPCA feature. If the result of the sign function is positive then the class's decision refers to the first class, otherwise if it is negative it will refer to the second class.
After all test data has been entered into the sign function will result in a class decision. In the class decision finally used a voting mechanism. The test data class is the largest voting class of the hyperplane function. This process is based on the One against One (One vs One) mechanism, due to recognize multiclass faces. If there is the same voting value then it will be chosen, the first largest voting index, so that if found the same voting value for the next class index will be ignored.

IV. Experiments and Results

Experiments using a number of training and testing data taken from the face image database. The result is the accuracy of the recognition performance. The database used is taken from the ORL database which is developed by AT&T Laboratory, Cambridge University [10]. ORL face image is a collection of face images consisting of 40 human faces. ORL image is the standard face of one head. This face has different variations in lighting, different expressions (eyes open / closed, smile / no smile) and accessories in the form of glasses. The background in each image is dark. The image is in jpg format with size of 92x112 pixels and in the form of grayscale images. ORL database consisted of 40 different faces of different persons and each person had 10 different face images, thus the total were 400 face images. For testing stage, 3 face images for each subject were taken, so the number of testing data were 120 face images (30% of the total dataset). The example of images from ORL database is shown on Fig. 3.

Fig. 3. Face image examples from ORL database [10]

Testing environment in face recognition application using kernel eigenface/KPCA and SVM (Support Vector Machine) methods was carried out with several proportion of test data (30% of data sets) and training data used 70% of data sets. The test scenario was based on the use of parameters in the training process: i.e. the KPCA gamma (γ) parameter, the gamma (γ) SVM, and complexity, for the number of principal components used were 25 components. Every parameter change was being tested on the test data and we observed the accuracy obtained. The test results can be summarized as in Table 1.

TABLE I. RECOGNITION ACCURACY RATE

No.	Parameters			Accuracy (%)
	γ KPCA	γ SVM	*C*	
1	1.5	5000	1200	96.67
2	1	5000	1200	96.67
3	1.5	5000	1500	96.67
4	1.5	6000	1500	95
5	1	6000	1200	96.67
6	1	6000	1500	95.83
			Average	**96.25%**

From the experiment we obtain the results on Table. 1 which shows the level percentage of recognition accuracy rate generated from the face recognition system using KPCA and SVM. The average accuracy rate is 96.25%. KPCA and SVM face recognition systems depend on the parameter setting: the γ KPCA, γ SVM, and Complexity.

We also compare the recognition accuracy with other methods using AT&T dataset. We present the comparison results in Table 2.

TABLE II. RECOGNITION ACCURACY RATE

No.	Methods	Accuracy (%)
1	K Nearest Neighbor (KNN)	95.00
2	Principle Component Analysis (PCA)	94.25
3	Linear Discriminant Analysis (LDA)	96.00
4	Kernel PCA (proposed method)	96.25

Results on Table 2 implies that our proposed method gained the highest accuracy rate compared to other feature extraction methods. Moreover, when it is compared with traditional PCA, the proposed KPCA with SVM classifier increased 2% in terms of accuracy rate. Hence, the proposed KPCA and SVM has proven its performance is the best amongst other methods (KNN, PCA, and LDA).

V. CONCLUSION

This paper proposed a face recognition problem using Kernel Principal Component (KPCA) method for feature extraction and Support Vector Machine (SVM) as its classifier. The accuracy rate of facial recognition tested in the ORL dataset using the kernel eigenface/KPCA as feature extraction method and SVM yielded an average accuracy rate of 96.25%. The parameter values C, γ SVM, and γ KPCA influenced the accuracy rate achieved. With the C parameter value increasing, the accuracy was also increasing, while the smaller the gamma SVM value increased accuracy, and the greater the KPCA gamma value also increased accuracy. Face recognition is applicable in many areas, including health, security, business and marketing, etc. In the future the challenge is to detect and recognize face in a real-time situation.

REFERENCES

[1] S. Arya, N. Pratap, and K. Bhatia, 'Future of Face Recognition: A Review', Procedia Comput. Sci., vol. 58, pp. 578–585, 2015.

[2] A. Cheddad and D. Mohamad, 'SEGMENTATION METHODS IN FACE RECOGNITION', vol. 2, p. 38.

[3] A. Bouzalmat, J. Kharroubi, and A. Zarghili, 'Comparative Study of PCA, ICA, LDA using SVM Classifier', J. Emerg. Technol. Web Intell., vol. 6, no. 1, pp. 64–68, Feb. 2014.

[4] I. K Timotius and A. A Febrianto, 'Face Recognition between Two Person using Kernel Principal Component Analysis and Support Vector Machines', Int. J. Electr. Eng. Inform., vol. 2, no. 1, pp. 53–61, Mar. 2010.

[5] J.-B. Li, S.-C. Chu, and J.-S. Pan, Kernel Learning Algorithms for Face Recognition. 2011.

[6] J. C. Platt, 'Advances in Kernel Methods', B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.

[7] E. Gumus, N. Kilic, A. Sertbaş, and O. Ucan, 'Eigenfaces and Support Vector Machine Approaches for Hybrid Face Recognition', vol. 2, 2010.

[8] S. N. Sujay, H. S. M. Reddy, and J. Ravi, 'Face recognition using extended LBP features and multilevel SVM classifier', in 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, 2017, pp. 1–4.

[9] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[10] F. Samaria and A. Harter, 'Parameterisation of a stochastic model for human face identification', Proc. 1994 IEEE Workshop Appl. Comput. Vis., pp. 138–142, 1994.

Face Recognition with Kernel Principal Component Analysis and Support Vector Machine

ORIGINALITY REPORT

6%	6%	5%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

2%

★ airccse.org

Internet Source

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		