Face Recognition Based on SVM and 2DPCA

Thai Hoang Le, Len Bui

Faculty of Information Technology, HCMC University of Science
Faculty of Information Sciences and Engineering, University of Canberra
lhthai@fit.hcmus.edu.vn, len.bui@canberra.edu.au

Abstract

Face recognize:
- 2D-PCA (Tách feature)
- SVM (Mô hình phân biệt)

Datasets: FERET + AT&T
-> Tăng tỉ lệ phân loại

The paper will present a novel approach for solving face recognition problem. Our method combines 2D Principal Component Analysis (2DPCA), one of the prominent methods for extracting feature vectors, and Support Vector Machine (SVM), the most powerful discriminative method for classification. Experiments based on proposed method have been conducted on two public data sets FERET and AT&T; the results show that the proposed method could improve the classification rates.

Discriminative: Tập trung vào quyết định border để tách 2 labels -> Label phải có nhãn và dùng cho supervised data.
=> Đối ngược là Generative.

Keywords: *2DPCA, SVM.*

1. Introduction

Sự cần thiết của nhận diện khuôn mặt (Sinh trắc học-biometric) trong đời sống.
- Eg: Vùng an ninh, bảo mật, chấm công
-> Dùng ảnh chụp camera so sánh với ảnh có sẵn => Match or not

Sinh trắc học

Human faces contain a lot of important biometric information. The information can be used in a variety of civilian and law enforcement applications. For example, identity verification for physical access control in buildings or security areas is one of the most common face recognition applications. At the access point, an image of a claimed person's face is captured by a camera and is compared with stored images of the claimed persons. Then it will be accepted only if it is matched. For high security areas, a combination with card terminals is possible, so that a double check is performed.

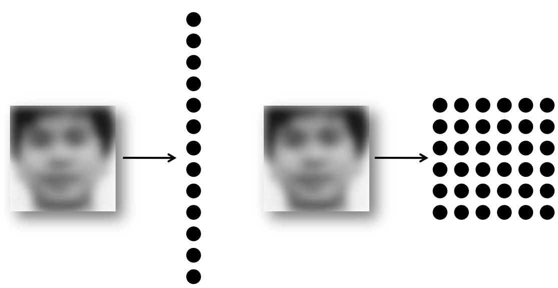


Figure 1. Image Representations in PCA and 2DPCA

PCA tốt + ổn định để tách feature vector

-> Không bắt được các biến thể local trừ khi nó được cung cấp từ trước ở training data.

Since Matthew Turk and Alex Pentland [1] used Principal Component Analysis (PCA) to deal with the face recognition problem, PCA has become the standard method to extract feature vectors in face recognition because it is stable and has good performance. Nevertheless, PCA could not capture all local variances of images unless this information is explicitly provided in the training data. To deal with this problem, some researchers proposed other approaches. For example, Wiskott et al. [2] suggested a technique known as elastic bunch graph matching to extract local features of face images. Penev and Atick [3] proposed using local features to represent faces; they used PCA to extract local feature vectors. They reported that there was a significant

3. ICA(Independent Component Analysis) để extract phần phụ thuộc cao của face mà không dựa vào phân phối Guassian
4. Kernel PCA (non-linear subspace) >>PCA(linear subspace) (Tốt nhưng cost tốn hơn)

History of solving local variance of img:

1. elastic bunch graph
2. local feature to represent faces (Tốt)

improvement in face recognition. Bartlett et al. [4] proposed using independent component analysis (ICA) for face representation to extract higher dependents of face images that cannot represented by Gaussian distributions, and reported that it performed better than PCA. Ming-HsuanYang [5] suggested Kernel PCA (or nonlinear subspace) for face feature extraction and recognition and described that his method outperformed PCA (linear subspace). However, the performance costs of them are higher than PCA.

2D-PCA ra đời
- PCA biểu diễn Img = Vector (1D)
- 2D-PCA biểu diễn Img = ma trận(2D)
 -> Simple + LocalInfo đầy đủ
 -> Tạo nhiều feature quan trọng để đại diện img

Eg:Frontal face(mặt trực tiếp) << profile face img (ảnh đại diện)->nhiễu

=> Dùng Weighted-2DPCA

To solve these problems, Jian Yang [6] proposed a new method called 2D Principal Component Analysis (2DPCA). In conventional PCA, face images have been represented in vectors by some technique like concatenation. As opposed to PCA, 2DPCA represents face images by using matrices or 2D images instead of vectors (Fig. 1). Clearly, using 2D images directly is quite simple and local information of the original images is preserved sufficiently, which may bring more important features for facial representation. In face identification, some face images are easy to recognize, but others are hard to identify; for example, frontal face images are easier than to be recognized than profile face images. Therefore, we proposed a weighted-2DPCA model to deal with the difficulty.

Giới thiệu về SVM và idea

In 1995, Vapnik and Cortes [7] presented the foundations for Support Vector Machine (SVM). Since then, it has become the prominent (nổi bật) method to solve problems in pattern classification and regression. The basic idea behind SVM is finding the optimal linear hyperplane such that the expected classification error for future test samples is minimized, i.e., good generalization performance. Obviously, the goal of all classifiers is not to get the lowest training error. For example, a k-NN classifier can achieve the accuracy rate 100% with k=1. However, in practice, it is the worst classifier because it has high structural risk.

Dù trainning là 100% ->Overfitted, nhiều risk và phức tạp
-> Vào testing thì error lớn(Vì nó quá điểm trũng)
-> Chỉ nên vừa đủ

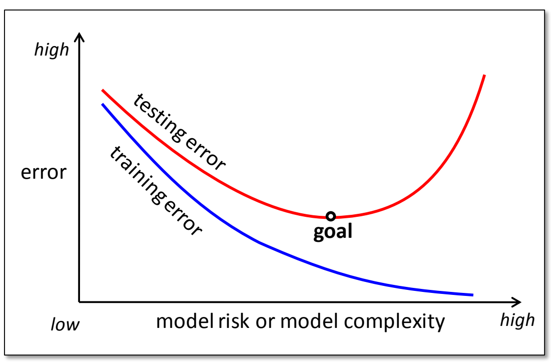
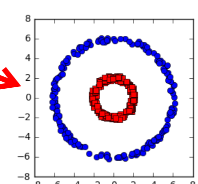
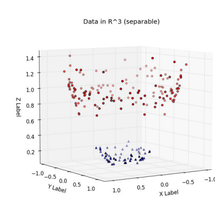


Figure 2. Curves of Testing Error and Training Error

They suggested the formula testing error = training error + risk of model (Fig. 2). To achieve the goal to get the lowest testing error, they proposed the structural risk minimization inductive principle (quy nạp). It means that a discriminative function that classifies the training data accurately and belongs to a set of functions with the lowest VC dimension will generalize best results regardless of the dimensionality of the input space. Based on this principle, an optimal linear discriminative function has been found. For linearly non-separable data, SVM maps the input to a higher dimensional feature space where a linear hyperplane can be found. Although there is no warranty that a linear solution will always exist in the higher dimensional space, it is able to find effective solutions in practice. To deal with the face gender classification, many researchers [8-11] have applied SVM in their studies and stated that the experiment results are very positive. In our research, we have combined the power of each method, weighted-2DPCA and SVM, to solve the problem.

Vapnik–Chervonenkis dimension





Mục tiêu:

Testing error: Trainning error + Risk model -> Cần minimum
-> Giảm risk bằng quy nạp

SVM-> Tìm discriminative function: phân loại data + lowest VC dimension bất kể Dim của input

SVM: Luôn có 1 function tuyến tính để phân tách 2 label.
- Ở linearly non-separable data, maybe nhìn vào hình 2D thì không có line nào tách điểm Blue và Red nhưng phải chuyển sang 3D (higher Dim) thì tồn tại.
- Ko sure là luôn có linear sol trong higher Dim nhưng sure là có effective sol.

EG: Apply phân biệt giới tính qua khuôn mặt

Algorithm

Implementation

The remaining sections of our paper will discuss the implementation of our face recognition system, related theory, and experiments. Section 2 gives details of 2DPCA. Section 3 discusses how to use SVM in face classification. In Section 4, we will describe the implementation and experiments. Finally, Section 5 is our conclusion.

Step1: Có matrix NxXxY

Step2: Trừ vs Mean N ảnh size (X,Y)

Step3: Tìm Covariance matrix

Step4: Eigen Vec for Covan matrix

Step5: Hình thành tạo độ mới (nhỏ hơn ban đầu)

PCA: https://www.youtube.com/watch?v=ZwiDOse1wQU

2D-PCA: https://www.youtube.com/watch?v=Ack2sDP4gxo

2. 2D Principal Component Analysis

2D-PCA, lấy ví dụ

2.1. Face Model Construction

Có trọng số

As mentioned above, we propose a weighed-2DPCA to deal with some practical situations in which some face images in database are difficult to identify due to their poses (front or profile) or their qualities (noise, blur).

Training data $D=\left\{\left(\mathbf{A}^{(i)}, w_i\right), i=1,\ldots,N\right\}$

Tự setup

- Tập data có N phần tử. 1 tấm ảnh có size là XxY
- A.shape = (X,Y) ở PCA thì làm phẳng (1,XY)
- w = number tự setup, Eg: male=1, female=2,..

Default là bằng nhau mà đây là Weighed-2DPCA

EVD(Eigen Value Decomposition)

Algorithm 1: Construct proposed face model

Step 1: Compute the mean image

Ma trận XxY

$$\overline{\mathbf{A}}=\frac{\sum_{i=1}^{N} w_i \mathbf{A}^{(i)}}{\sum_{i=1}^{N} w_i} \tag{1}$$

Step 2: Compute matrix

Covanriance

G = E[(A-E[A])^T * (A-E[A])]

YxX XxY

$$\mathbf{G}=\frac{\boxed{\sum_{i=1}^{N} w_i}\left(\mathbf{A}^{(i)}-\overline{\mathbf{A}}\right)^T\left(\mathbf{A}^{(i)}-\overline{\mathbf{A}}\right)}{\boxed{\sum_{i=1}^{N} w_i}} \tag{2}$$

YxY

Each size: MNx1-vector cột

Con số

Step 3: Compute eigenvectors $\{\mathbf{\Omega}_1, \mathbf{\Omega}_2, \ldots, \mathbf{\Omega}_n\}$ and eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ of G.

có n nghiệm = Y

Vector riêng, Giá trị riêng => Linear Algebra

2.2. Feature Extraction

Lấy ví dụ

Tính toán thử trên 1 ảnh

First, a projection point of image A on 2DPCA space is matrix $\left(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\right)$

có n nghiệm

có d nghiệm

$$\mathbf{X}_k=\left(\mathbf{A}-\overline{\mathbf{A}}\right)\mathbf{\Omega}_k, k=1,\ldots,d \tag{3}$$

Xét trong 1 tấm ảnh

Từ 2D thành 1D

Second, the matrix is projected on PCA space to convert matrix to vector and reduce the dimension.

3. Support Vector Machine

The goal of SVM classifiers is to find a hyperplane that separates the largest fraction of a labeled data set $\boxed{\{(\mathbf{x}^{(i)}, y^{(i)}); \mathbf{x}^{(i)} \in \mathbb{R}^n; y^{(i)} \in \{-1,+1\}; i=1,\ldots,N\}}$. The most important requirement, which the classifiers must have, is that it has to maximize the distance or the margin between each class and the hyperplane (Fig 3.).

Dùng cho 2 label

Binary Classification

In most of real applications, the data could not be linearly classified. To deal with this problem, we transform data into a higher dimensional feature space and assume that our data in this space can be linearly classified (See Fig 4.).

$$\begin{aligned}\Phi:\ \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ \mathbf{x} &\mapsto \Phi(\mathbf{x})\end{aligned} \qquad (4)$$

Giống với Linear Programming của MMH

In fact, determining the optimal hyperplane is a constrained optimization problem and can be solved using quadratic programming techniques. The discriminant hyperplane is defined as the following

Hyperplane cần tìm

$$y(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y^{(i)} K\left(\mathbf{x}^{(i)}, \mathbf{x}\right) + b \qquad (5)$$

Alpha_i là hệ số góc
Beta là hệ số
--> Cần tìm

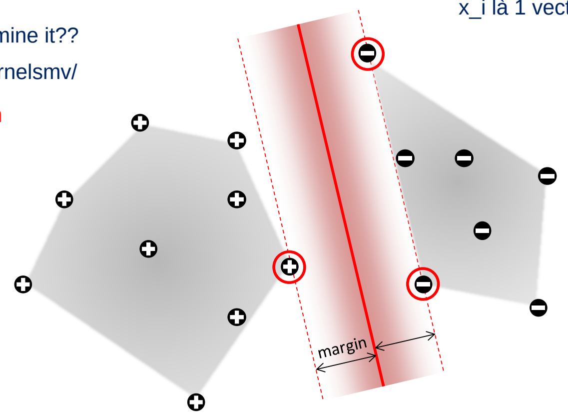
where $K(\mathbf{x}', \mathbf{x}'')$ is the kernel function.

y_i là 0, 1 tùy vào class label
x_i là 1 vector input (1D or 2D)

Ko hiểu
How to determine it??

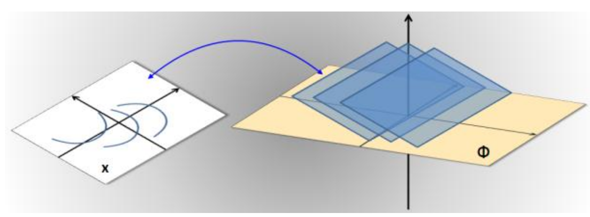
Link: https://machinelearningcoban.com/2017/04/22/kernelsmv/

5-6 hàm có sẵn: Chú ý hàm Radial Basic Function



Làm max phần margin

Figure 3. An SVM Classifier



2D không thấy thì lên 3D, 4D

Figure 4. Input Space and Feature Space

3.1. Classifier Construction Phase

Algorithm 2: Construct classifier

Step 1: Compute matrix H

$$H_{ij} = y^{(i)} y^{(j)} K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) \qquad (6)$$

i,j từ 1->N(N là số img)
Matrix H size (N,N)

Step 2: Use quadratic solver to solve the optimization problem with objective function:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left(\frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} - \sum_{i=1}^{N} \alpha_i \right)$$

là 1 con số để so sánh

alpha là ma trận cột của các alpha_i

Tìm min biểu thức ở trong và dùng nghiệm alpha của nó

$$\begin{cases} 0 \le \alpha_i \le C \\ \sum_{i=1}^{N} \alpha_i y^{(i)} = 0 \end{cases} \qquad (7)$$

Constraint

Link: https://www.youtube.com/watch?v=GZb9647X8sg

Step 3: Compute b

Có alpha và tính tiếp Beta

$$
\begin{aligned}
& idx = \{ i \mid \alpha_i > 0 \} \\
& N_{idx} = |idx| \\
& b = \frac{1}{N_{idx}} \sum_{i \in idx} \left(y^{(i)} - \sum_{j \in idx} \alpha_j y^{(j)} K\left(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}\right) \right)
\end{aligned} \tag{8}
$$

3.2. Classification Phase

Algorithm 3: Classify

Dùng để test trên data và ra độ chính xác:
x là 1 điểm đang xét => Tính toán các biểu thức=> y=+1 or -1
=> Có label của x

Step 1: Compute the value y

$$
y = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y^{(i)} K\left(\mathbf{x}^{(i)}, \mathbf{x}\right) + b \right) \tag{9}
$$

Signum function

Bằng 1 khi ở trong >0, bằng -1 khi ở trong <0, bằng 0 khi ở trong = 0

Step 2: Classify for x

$$
\boxed{\begin{cases} \text{if } y = +1 \text{ then } \mathbf{x} \text{ belong class } \{+1\} \\ \text{if } y = \text{-}1 \text{ then } \mathbf{x} \text{ belong class } \{\text{-}1\} \end{cases}} \tag{10}
$$

3.3. SVM for Face Identification

https://www.youtube.com/watch?v=kb4apnc2imA&t=264s Approach 2: (8:18) Video

To apply SVM in face recognition, we use One-Against-All decomposition to transform multi-class problem to a set of two-class problems.

Training set $D = \{(\mathbf{x}^{(i)}, y^{(i)}); \mathbf{x}^{(i)} \in \mathbb{R}^n; y^{(i)} \in \{-1, +1\}; i = 1, \ldots, N\}$ is transformed to series of $D_k = \{(\mathbf{x}^{(i)}, y_k^{(i)}); y_k^{(i)} \in \{-1, +1\}\}$

Tách ra thành các

where

$$
\boxed{y_k^{(i)} = \begin{cases} +1 & y^{(i)} = k \\ -1 & y^{(i)} \neq k \end{cases}} \tag{11}
$$

Algorithm 2 is used to compute the discriminant functions corresponding to D_k.

$$
f_k(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_k^{(i)} K\left(\mathbf{x}^{(i)}, \mathbf{x}\right) + b \tag{12}
$$

In classification phase, we use the following rule to identify the class for input x.

$$
k = \arg\max_k \left(f_k(\mathbf{x}) \right) \tag{13}
$$

Chọn ra Class phù hợp trong k class

4. Implementation and Experiments

Viết code

We select FERET and AT&T databases to evaluate our approach. The FERET database [12] was collected at George Mason University between August 1993 and July 1996. It contains 1564 sets of images for 14,126 images that include 1199 individuals and 365 duplicate sets of images. In our experiments, face regions of FERET images were identified and extracted from the background of the input images using the ground truth information of images but some images do not contain information on face locations. In this case, we used the well-known algorithm developed by Viola and Jones [13, 14] to find face positions. Then, they were scaled to 50-by-50 resolution. In dataset

tách vùng mặt bằng technique

How to know Face region
- Dùng face truth cho sẵn
- Ko cho thì dùng Viola and Jones để tìm Face position. Scale 50-50

Cách build riêng dataset D: Chia nhỏ ra A,B,C

Dataset M

building task, we constructed a dataset D containing 1000 individuals which are chosen from sets fa, fb, fc, dup1 and dup2 of 1996 FERET database. All images of the dataset D are frontal face images. Next, we randomly divided the dataset into 3 separate subsets A, B and C. The reported results were obtained with Cross-Validation analysis on these subsets. We also use training set M of database provided by FERET for PCA feature extraction and 2DPCA extraction.

Pick AT&T database

The AT&T database was taken at AT&T Laboratories. It contains 400 images (92-by-112) of 40 individuals; each person has ten images. We performed the same tasks to build datasets for experiments.

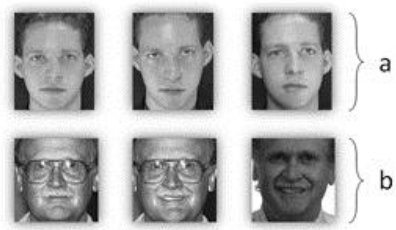


Figure 5. a) Three faces from AT&T b) Three processed faces from FERET

4.1. Experiments on AT&T database

Chọn cái này vì cụ thể và ez hơn

Có Library sẵn

We implemented five methods to conduct experiments on the AT&T database:

Cái này dùng tensorflow ổn

- MLP (PCA): This method uses PCA to extract feature vectors and Multi Layer Perceptron (MLP) for classification. The MLP has three layers: input layer has 163 nodes, hidden layer has 100 nodes, and output layer has 40 nodes. This MLP uses Gradient Back-Propagation algorithm for training. The active function of MLP is sigmoid function $f(x)$ and the range of learning rate η is between 0.3 and 0.7.

$$f(x)=\frac{1}{1+e^{-x}},\ f\in[0,1] \tag{14}$$

- k-NN (PCA): We use PCA to obtain feature vectors and employ k-Nearest Neighbor (k-NN) with distance metric L2 for classification.
- SVM (PCA): It uses PCA to get feature vectors and applies SVM with two kernel functions (Polynomial, Radial Basis Functions-RBF) for classification. The value of d of Polynomial is 3; for RBF kernel we used some values $C=\{2^{-5},...,2^{14}\}$ and $\sigma=\{2^{-15},...,2^{3}\}$ for classification.

$$K(\mathbf{x},\mathbf{x}')=(\mathbf{x}^T\mathbf{x}'+1)^d$$
$$K(\mathbf{x},\mathbf{x}')=e^{-\left(\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)} \tag{15}$$

- k-NN (2DPCA): The method uses our proposed 2DPCA to get feature vectors and employs k-NN for classification.

Implement 2 cái để so sánh

- SVM (2DPCA): It uses the proposed 2DPCA to get feature vectors and SVM for classification.

We used the subset M to create PCA feature extractor. The default dimension of feature vector is k=163. With this k, we can get to a reasonable PCA reconstruction error of MSE = ~~0.0015~~. We also used the same subset M to create 2DPCA feature extractor. A weight for each training image is its rotate angle. The dimension of feature vector is k = 20.

Weight

For each method, we conducted three experimental trials on subsets A, B and C. It means that we trained classifiers on two subsets and evaluated on the remaining subset. The results are reported on their average performance scores in Table I.

Làm trên A,B
Đánh giá lại trên C

The cumulative match score vs. rank curve for each method has been show in Fig. 6. The values of curve are the percentage of correct matches in the top n matches (rank-5).

Chưa hiểu về chart

The experimental results point that our proposed method for feature extraction is better than PCA and 2DPCA methods. As mentioned above, PCA is a method to reduce the dimension. There is not any mathematical evidence that it will increase the recognition rate. Our method has more advantages than traditional 2DPCA because it can create a subspace that reserves some importance discriminative information of face images such as pose.

The experimental results also show that MLP is the worst classification method and SVM is the best one. Obviously, MLP is easy to be overfitting because they usually focus on finding the lowest error rate although we use some techniques such as cross validation to limit the weak point. In other hand, SVM method always gives a suitable solution.

TABLE I. **EXPERIMENT RESULTS ON AT&T DATABASE**

Feature extraction	Classification	Accuracy (%)
PCA	MLP	75.2
PCA	k-NN	95.2
PCA	SVM	95.7
2DPCA	k-NN	96.2
2DPCA	SVM	97.3

SO SÁNH

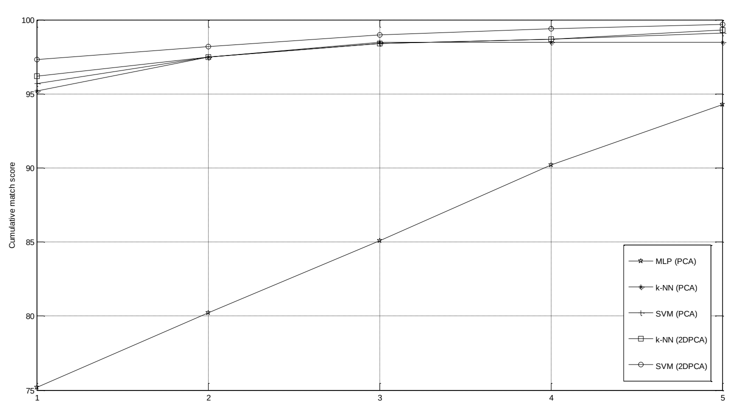


Figure 6. Identification Performance on AT&T Database

4.2. ~~Experiments on FERET Database~~

We implemented four methods to conduct experiments on FERET database, which are k-NN (PCA), SVM (PCA), k-NN (2DPCA) and SVM (2DPCA). We did the same task to build feature extractors. First, we used the subset M to create PCA feature extractor. The default dimension of feature vector is k=100. Then, we continued to use the same subset M to create 2DPCA feature extractor. In our experiments, we set weight for female is 3, for male is 2 and for individual with glass is 1. It means that an image be easy to recognize has higher weight. The dimension of feature vector is k = 10. Weight

We conducted three experimental trials on subsets A, B and C for each method. The results are reported on their average performance scores in Table II; and the cumulative match score vs. rank curve (rank-50) for each method has been shown in Fig. 7. The method 2DPCA with SVM for classification still gets the best performance on the FERET dataset.

TABLE II. **EXPERIMENT RESULTS ON FERET DATABASE**

Feature extraction	Classification	Accuracy (%)
PCA	L2	80.1
PCA	SVM	85.2
2DPCA	L2	90.1
2DPCA	SVM	95.1

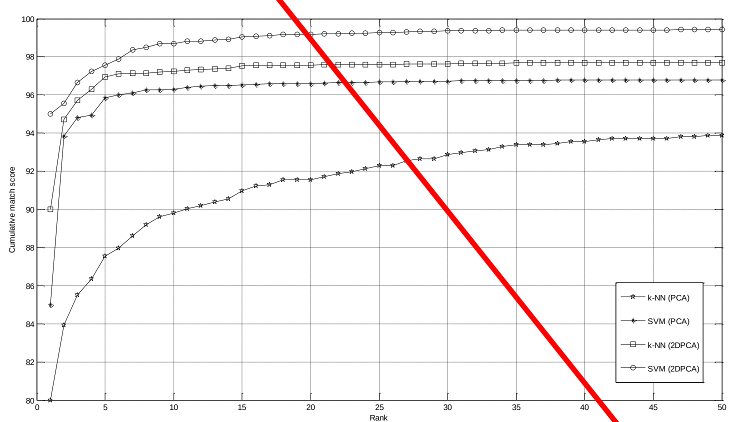


Figure 7. Identification Performance on FERET Database

5. Conclusions

In summary, we have proposed a new approach for face recognition. The first contribution of this paper is to propose a novel face model based on conventional 2DPCA for extracting feature vectors. The second contribution of this paper is to combine our proposed face model with SVM. We have compared our method with traditional methods. The results from our methods outperformed significantly.

References

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, 1991, pp. 586-591.

[2] L. Wiskott, *et al.*, "Face recognition by elastic bunch graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, pp. 775-779, 1997.

[3] P. Penev and J. Atick, "Local feature analysis: a general statistical theory for object representation," *Network: computation in neural systems*, vol. 7, pp. 477-500, 1996.

[4] M. S. Bartlett, *et al.*, "Face recognition by independent component analysis," *Neural Networks, IEEE Transactions on,* vol. 13, pp. 1450-1464, 2002.

[5] Y. Ming-Hsuan, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face recognition using kernel methods," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on,* 2002, pp. 215-220.

[6] Y. Jian, *et al.*, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, pp. 131-137, 2004.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, pp. 273-297, 1995.

[8] C. Huajie and W. Wei, "Pseudo-Example Based Iterative SVM Learning Approach for Gender Classification," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, 2006, pp. 9528-9532.

[9] B. Moghaddam and Y. Ming-Hsuan, "Gender classification with support vector machines," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 306-311.

[10] H. Xia, *et al.*, "Gender Classification Based on 3D Face Geometry Features Using SVM," in *CyberWorlds, 2009. CW '09. International Conference on*, 2009, pp. 114-118.

[11] L. Xue-Ming and W. Yi-Ding, "Gender classification based on fuzzy SVM," in *Machine Learning and Cybernetics, 2008 International Conference on*, 2008, pp. 1260-1264.

[12] P. Phillips, *et al.*, "The FERET evaluation methodology for face-recognition algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, pp. 1090-1104, 2002.

[13] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision,* vol. 57, pp. 137-154, 2004.

[14] L. Thai Hoang and B. Len Tien, "A hybrid approach of AdaBoost and Artificial Neural Network for detecting human faces," in *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on*, 2008, pp. 79-85.

Authors

Thai Hoang Le received B.S degree and M.S degree in Computer Science from Hanoi University of Technology, Vietnam, in 1995 and 1997. He received Ph.D. degree in Computer Science from Ho Chi Minh University of Sciences, Vietnam, in 2004. Since 1999, he has been a lecturer at Faculty of Information Technology, Ho Chi Minh University of Science, Vietnam. His research interests include soft computing pattern recognition, image processing, biometric and computer vision. Dr. Thai Hoang Le is co-author over twenty five papers in international journals and international conferences.

Len Bui received B.S degree and M.S degree in Computer Science from Ho Chi Minh University of Science, Vietnam, in 1996 and 2001. Since 2008, he has been a Ph.D. student in Computer Science at Faculty of Information Sciences and Engineering, University of Canberra, Australia.