ACCV2002: The 5th Asian Conference on Computer Vision, 23--25 January 2002, Melbourne, Australia

Face Recognition Based on Support Vector Method

Guoqin Cui, Wen Gao, Feng Jiao, and Shiguang Shan
Institute of Computing Technology, Chinese Academy of Sciences,
P.O.Box 2704, Beijing, 100080,China
e-mail:{cgq,wgao,fjiao,sgshan}@ict.ac.cn

Abstract

Support Vector Machines are a binary classification method and have demonstrated excellent results in pattern recognition. Face recognition is a multi-class problem, where the number of classes is of the known individuals. This paper we use face data extracted from Eigenfeatures and developed a method to extend SVM to using in multi-class. The training set consists of 5 images of each of the 50 persons equally distributed among frontal, approximately 15°rotated respectively, and the test set consists of 10 images each of the 50 persons. In the ICT-YCNC face gallery, the proposed system obtains competitive results highly: a correct recognition rate of 94.8% for all the 50 persons, to the less number of the persons and to the famous ORL face gallery we also get good face recognition rate.

Index Terms**-- Face recognition, Support vector machine, Principal components analysis, Multi-class problem.**

1 INTRODUCTION

HIGH-SECURITY verification systems based on biometric modalities such as gesture, signature dynamics, iris, speech and fingerprints have been commercially available for some time. However, one of the most attractive sources of biometric information is the human face recognition because highly discriminative measurements can be acquired without user interaction. Human face recognition is a well-established and very difficult research field, a great large number of algorithms have been proposed in the literature in the past 20 years but is still not used commercially as any other biometric modalities.

Human face recognition, also different from other classical pattern recognition problems such as character recognition, there are relatively few classes, and many samples for one class. Algorithms can classify samples not previously seen by interpolating among the training samples. On the other hand, in face recognition, there are many individuals (classes), and only a few images (samples) for a person, and algorithms must recognize faces by deducing from the training samples. Moreover since we don't know what are the relevant features of the problem, the data points usually belong to some high-dimensional space (for example a face image may be represented by its gray level values). Therefore there is a need for pattern recognition techniques that can degrade to fewer dimensions.

Support vector machines (SVMs) are formulated to solve a classical two-class pattern recognition problem [1,2]. We adapt SVM to face recognition by using eigenface technique [3] to get less dimension data to express the face image, modifying the interpretation of the output of SVM classifiers and devising standard way that is correspond to a multi-class problem. Thus the algorithm can return a confidence measure of the validity of the claim. We report the result on 750 images of 50 individuals that are extracted from the ICT-YCNC database of images, which is constructed by our research lab last two years. From our experience with the ICT-YCNC database, we selected 500 images with freedom on which to test the algorithms. The left 250 images are for training multi-class SVMs.

The plan of this paper is as follows: in section 2 we briefly introduce the SVM algorithm and the training method we used in our system, in section 3 we extend SVM to solve the multi-class problem, in section 4 we describe how to get the eigenface data from the face image in order to use SVM to solve the training problem, in section 5 we give the procedure of our face recognition system, in section 6 we introduced our results and in the section 7 we summarize the conclusion of the system.

2 SUPPORT VECTOR MACHINES (SVMS)

SVM is a binary classification method that finds the optimal linear decision surface based on the concept of *structural risk minimization (SRM)* principle [5]. In this section, we briefly review the algorithm of SVM and its motivation in classification problems. Interested readers may consult [1,2,7], for details.

Let the training set D be a set $\{(x_i, y_i)\}$, with each input $x_i \in R^N$ and $y_i = 1$ or –1 is the label of $x_i, i = 1,2,\cdots,d . d$ is the total number of the training data.

In basic form, SVMs learn linear decision rules

$$f(x) = sign(\omega \cdot x + b)$$

described by a weight vector ω and a threshold b .The idea of *SRM* is to find a hypothesis f for which one can guarantee the lowest probability of error. For SVMs, C. J. C. Burges [1] shows that this goal can be translated into finding the hyper-plane with maximum soft-margin between the two classes, where the margin is defined as the sum of the distances of the hyper-plane from the closest point of the two classes. *Figure 1* gives a geometric interpretation of the margin and the positions of the support vectors. Computing this hyper-plane is equivalent to solving the following optimization problem.

$$\begin{aligned} &\underset{\omega,b}{Min}\ \frac{1}{2}(\omega\cdot\omega) \\ &s.t.\quad y_i(\omega\cdot x_i + b) \geq 1,\quad i = 1,2,\cdots,d \qquad (*) \end{aligned}$$

Using the optimization theory and Method [11], the above problem can be changed to the following Wolfe dual Lagrangian:

$$\begin{aligned} &Max\,W(\alpha) = \sum\nolimits_{i=1}^{d}\alpha_i - \frac{1}{2}\sum\nolimits_{i,j=1}^{d}\alpha_i\alpha_j y_i y_j (x_i x_j) \\ &s.t.\quad \alpha_i \geq 0, i = 1,2,\cdots,d \qquad (**) \\ &\qquad \sum\nolimits_{i=1}^{d}\alpha_i y_i = 0 \end{aligned}$$

The optimal hyper-plane is mainly defined by the weight vector $\omega = \sum\nolimits_i \alpha_i y_i \cdot x_i$, which consists of all the data elements with non-zero Lagrange multipliers α_i, those elements lay on the margins of the hyper-plane (note: if the α_i). They define both the hyper-plane and the boundaries of the two classes. The decision function of the optimal hyper-plane is thus:

$$f(x) = \mathrm{sign}(\sum\nolimits_{i=1}^{d} y_i\alpha_i(x\cdot x_i) + b)$$

For noisy data sets where there are some training examples lie on the "wrong" side of the hyper-plane, positive slack variables $\xi_i \geq 0, i = 1,2,\cdots,d$ is introduced in the constrains, which then become:

$$y_i(\omega\cdot x_i + b) \geq 1 - \xi_i,\quad i = 1,2,\cdots,d$$

Thus for an error to occur, the corresponding ξ_i must exceed unity, so $\sum\nolimits_i \xi_i$ is an upper bound on the number of training errors. Hence a natural way to assign an extra cost for errors is to change the objective function to be minimized from (*) to:

$$\begin{aligned} &\underset{\omega,b}{Min}\ \frac{1}{2}(\omega\cdot\omega) + C\sum\nolimits_i \xi_i \\ &s.t.\quad y_i(\omega\cdot x_i + b) \geq 1 - \xi_i, \\ &\qquad \xi_i \geq 0 \qquad\qquad i = 1,2,\cdots,d \end{aligned}$$

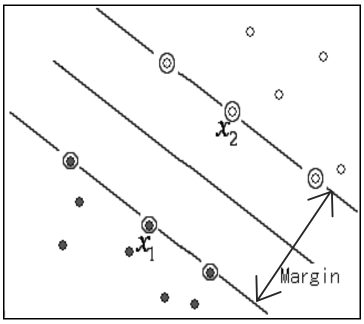


Figure 1. The margin is the perpendicular distance between the separating hyper-plane and a hyper-plane through the closest points the support vectors are circled such as x_1, x_2

where C is a parameter to be chosen by the user, a large C corresponding to assigning a higher penalty to errors.

According to the same way of using Wolfe dual method, (**) constrained condition of $\alpha_i \geq 0$ must be changed to $0 \leq \alpha_i \leq C$.

A hyper-plane classification function attempts to fit an optimal hyper-plane between two classes in a training data set, which will inevitably fail in cases where the two classes are not linearly separable in the input space R^N . Therefore, a high dimensional mapping

$$\Phi:\quad R^N \mapsto F$$

is used, and we can search the optimal linear planes in the new space F ,to cater for nonlinear cases in R^N . As both the objective function and the decision function is expressed in terms of dot products of data vectors x, the potentially computational intensive mapping $\Phi(\cdot)$ does not need to be explicitly evaluated. A kernel function $K(x,z)$, satisfying Mercẻr's condition [6] can be used as substitute for $(\Phi(x)\cdot\Phi(z))$ which replaces $(x\cdot z)$.

Therefore, the nonlinear objective function is

$$Max\,W(\alpha)=\sum_{i=1}^{d}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{d}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$
$$s.t.\quad 0\le\alpha_i\le C, i=1,2,\cdots,d$$
$$\sum_{i=1}^{d}\alpha_i y_i=0$$

Here the weight vector ω must be modified as $\omega=\sum_i \alpha_i y_i\cdot\Phi(s_i)$, with non-zero Lagrange multipliers α_i , s_i is the elements lay on the margins of the hyper-plane. Since we have use $K(x,z)$ replace $(\Phi(x)\cdot\Phi(z))$, thus the extended nonlinear decision surfaces is:

$$f(x)=sign(\sum_{i=1}^{d}y_i\alpha_i K(x,x_i)+b)$$

Since the bias, b , does not feature in the above dual formulation it is found from the primal constraints:

$$b=-\frac{1}{2}\left[\max_{\{i|y_i=-1\}}(\sum_{j\in\{SV\}}^{d}y_j\alpha_j K(x_i,x_j))+\min_{\{i|y_i=+1\}}(\sum_{j\in\{SV\}}^{d}y_j\alpha_j K(x_i,x_j))\right]$$

There are a number of kernel functions [2,5,10], which have been found to provide good generalization capabilities, e.g. polynomials $K(x,x_i)=(x^T x_i+1)^p$, multi layer perception function $K(x,x_i)=\tanh(\kappa\cdot x^T x_i-\theta)$ (with gain κ and offset θ) etc. In our system we explore the use of a radial basis function (also called Gaussian kernel function), the correspondent nonlinear decision surface is

$$f(x)=sign(\sum_{i=1}^{d}y_i\alpha_i\exp(\frac{|x-x_i|^2}{2\sigma^2})+b)$$

The confidence of a classification is directly related to the magnitude of $f(x)$. When the maximal margin hyper-plane is found in feature space, only those points, which lie closest to the hyper-plane, have $\alpha_i>0$ and these points are the *support vectors*. All other points have $\alpha_i=0$. This means that those points, which are closest to the hyperplane, solely give the representation of hypothesis and they are the most informative patterns in the data. During testing, for a test vector $x\in R^N$, we compute $f(x)$, and then get the class label of x .

From the algorithm we have narrated above, we can know the usual method to get the hyper-plane is to optimized the value of α_i step by step, in the same time $f(x)$ is changing until the hyper plane is got. In our system, John Platt's Sequential Minimal Optimization method [8,9] is used for computing the solution of this optimization problem.

3 SVM FOR MULTI-CLASS CLASSIFICATION

Multi-class pattern recognition systems can be obtained by combining two-class SVMs. The standard method for multi-class SVMs is to construct k SVMs where k is of the total number of the classes. The *i*th SVM will be trained with all of the examples in the *i*th class with positive labels, and all other examples with negative labels. We refer to SVMs trained in this way as *one vs. total* SVMs. In our face recognition system we use the *one vs. total* SVMs and get the output value of the k SVMs with the support vectors and the correlated multiplier with its' label '+' or '-'. Note: the multiplier is nonnegative number forever.

The disadvantage of this scheme (*one vs. total*) is that some test data might not be classified in a single class. In order to solve this problem the system give the 5 choices to each of the test samples.

4 FACES REPRESENTATION AND FEATURE SELECTION

4.1.Face Database

We have used the ICT-YCNC face database, which contains a set of faces taken between August 1999 and December 2000 at institute of computing technology of Chinese Academy of Sciences. There are about 200 different images of 50 distinct subjects. For some of the subjects the images were taken at different poses. There are variations in facial expression and facial details .All the images were taken against different homogeneous background with the subjects in an up right, frontal position, with tolerance for some tilting and rotation of up to about 15 degrees. There is some variation in scale of up to about 20% thumbnails of part of the images in the gallery are shown in *figure 2.*

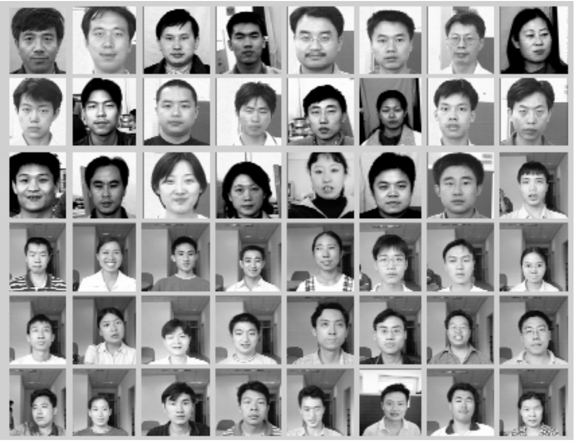
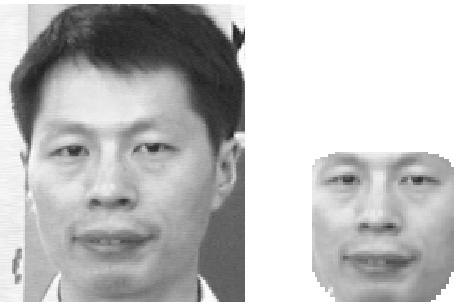


Figure 2.The ICT-YCNC face database.

4.2. Face Representations

It is natural to pursue dimensionality reduction schemes because great amounts of storage and are very difficult to process for its large dimension. A technique now commonly used for dimensionality reduction in computer vision, particularly in face recognition is principal components analysis (PCA). PCA techniques, also known as Karhunen-Loève methods, choose a dimensionality reducing linear projection that maximizes the scatter of all projected samples. Let's give the simple preview of this method [3,4,13].



(a) (b)

Figure 3. (a) Original image (b) Image after preprocessing.

Certainly at first we must pre-process the image to normalize geometry and illumination, and to remove background and hair (*figure 3*). The processing procedure consisted of manually locating the centers of the eyes; translating, scaling, and rotating the faces to place the center of the eyes on specific pixels; masking the faces to remove background and hair; histogram equalizing the non-masked facial pixels; and scaling the non-masked facial pixels to have zero mean and unit variance.

All the following algorithm and process are based on the image after the preprocessing.

Now we set a face image as a matrix or an array $[b_{ij}]$ that is the pixel value of ith line and jth row. And a $M \times M$ matrix in our system we set M =64 image can be constructed by a M^2-dimensional vector:

$$x = (b_{11}b_{21}\cdots b_{M1}b_{12}b_{22}\cdots b_{M2}\cdots b_{1M}b_{2M}\cdots b_{MM})$$

.

Let us consider a set of N sample images $\{x_1, x_2, \cdots, x_N\}$ taking values in a $m = M \times M$ dimensional feature space, and assume that each image belongs to one of c classes $\{\chi_1, \chi_2, \cdots, \chi_c\}$. Let us also consider a linear transformation mapping the original m dimensional feature space into a n dimensional feature space, where $n < m$. Denoting by $Q \in R^{m\times n}$ a matrix with orthonormal columns, the new feature vectors $y_k \in R^n$ are defined by the following transformation:

$$y_k = Q^T x_k, \quad k = 1,2,\cdots,N.$$

Now we narrate how to get the matrix *Q*.
The total scatter matrix Σ can be defined as

$$\Sigma = \sum_{k=1}^{N}(x_k - \mu)(x_k - \mu)^T \cdot$$

where x_k is the vector of *k-th* face image and $\mu = \frac{1}{N}\sum_{l=1}^{N} x_l \in R^n$ is the mean image of all samples. Certainly, the matrix Σ is *N-by-N*, real and symmetric; the diagonal elements are the variances of the individual random variables, while the off-diagonal elements are their co variances.

Let matrix $U = (\xi_1, \xi_2, \cdots, \xi_N)$ where ξ_i is the eigenvectors of Σ. For convenience, we arrange the rows in order of decreasing magnitude of the corresponding eigenvalues.

Thus

$$U\Sigma U^T = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix}$$

And the λ_i are the eigenvalues of Σ (corresponds to ξ_i), and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \cdot$

Now we discard the lower m-n rows of U, and get the matrix *Q*.

5 Recognition System

In our system we project the facial image on the eigenvectors generated according to the above method and seat $N = 64$. Thus a facial image can be represented as a feature vector $p \in R^N$, where R^N is called as *face spac*e in the system.

A recognition algorithm is presented with an image p and the number of the class is to be certified.

To the train data we labeled the different person in the different label such as $0,1,\cdots,m-1$, and the face data are from the eigenface technique that we have narrated in *section 4.2.*According to the method we have depicted in section 3, an *m-class* SVM algorithm will generate m different decision surfaces. For this m - $class$ (one for each algorithm a_k) , we can get the binary SVM classifier $u_k(x)$ and can separate a_k to other classes.

Thus we get the classifier of multi-class problem $a_{L(x)}$, for an input sample x:

$$L(x) = \arg\max_l \{u_l(x)\}$$

the $u_l(x)$ express the classifier function from SVM：

$$u_l(x) = \sum_{i=1}^{l_k} a_{ki} y_{ki} K(x_{ki}, x) - b_k$$

When we test a sample z to decide the class it belong to we calculate the valued of $u_l(z)$, l is from 1 to m, after that we can get $L(z)$ and the correspondent class label l_0, since we can't get all the first one as the right choice, we also get the four other classes that are the nearest to the value of l_0 within the set $\{u_l(x)\}$, otherwise the claim is not the class label we want

This classifier is designed to minimize *the structural risk*--an overall measure of classifier performance.

The recognition problem can be simply stated: Given a set of face images labeled with the person's identity (the training set) and an unlabeled set of face images from the same group of people (the test set), label the class number of each face image in the test set.

6 Experimental Results

We performed various experiments and present the results here. Except when stated otherwise, all experiments were performed with 5 training images and 10 test images per person for a total of 250 training images and 500 test images. There was no overlap between the training and test sets. We vary only one parameter in each case.

Table 1. Test results rate of the face recognition system with varying number of classes.

Classes Num.	T^*	Recog. Rate (%)
2	6	99.4
6	4	96.7
10	4	98.0
11	2	96.4
15	2	97.0
20	2	96.0
30	2	95.4
32	2	94.0
35	2	94.6
40	2	96.0
45	1	95.6
50	1	94.8

T^* **express the experiment times we have done. Recog. Rate is the average of the recognition rate (**T^* **times)**.

Beside that we choose not only all the 50 persons to test the algorithm, also choose less number of classes such as 45,35 randomly. The experiments are as Table 1.Variation of the number of output classes –*table 1* and *figure 4* show the recognition rate of the system as the number of classes is varied from 2 to 50. We made no attempt to optimize the system for the other numbers. As we expect, performance improves with fewer classes to discriminate between.

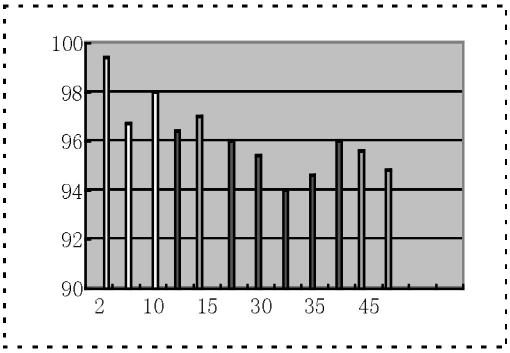


Figure 4. Recognition rate of the system with varying number of classes

After the above experiment we also use a famous ORL face database [12] to examine our system, we choose forty persons and ten images one person to train and ten images one person to test, all the same number class are do one time experiment *Table-2* shows the result:

Table 2. Test results rate of the face recognition system on ORL face database.

Classes Num.	Recog. Rate (%)
≤15	100
16	98.75
20	98.0
24	98.4
28	98.57
30	98
35	98.3
40	97

From *table-1* and *Table-2,* we can see the result of experiments we have done on ORL database is better than on ICT-YCNC database (*figure 5 shows the difference*.) The reason is that The ICT-YCNC is constructed by different people and has not so high quality that ORL have.

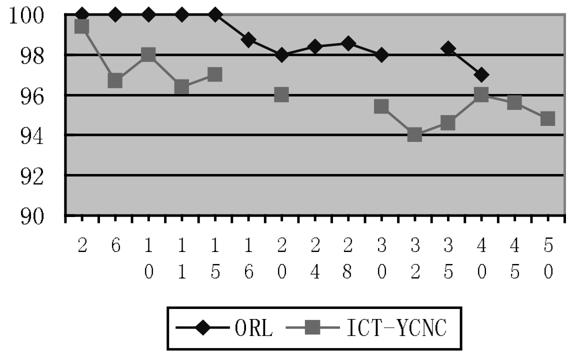


Figure 5. Difference of the recognition rate between the ORL and ICT-YCNC

7 CONCLUSIONS AND FUTURE WORK

This paper studied support vector machines in the application of face recognition by using *PCA* technique as the way to extract feature data. The face image database comes from our research lab and the open face image ORL database, we talk about the feature data are the same dimension not as P. J. Phillips do[14].Since from the experiment the recognition rate are almost same when the dimension are above 32. Beside that we use Radial Basis Functions as kernel approximation functions to training and test SVM. We presented an evaluation on a large face database showing competition recognition rates for recognition scenarios.

Since SVM appear to provide robust classification, we are beginning the test on about 1000 persons database, after that we are going to use SVM in surveillance system.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their detailed comments and questions, which improved the quality of the presentation of this paper. This paper supported by Yinchen Networks & Communication Corporation (YCNC), Chengdu, China.

REFERENCES

[1] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, Vol.2, pp.121-167. 1998

[2] V.Vapnik. The nature of statistical learning theory. Springer, New York, 1995.

[3] M. Turk and A. Pentland. Eigenfaces for recognition. J. Cognitive Neuroscience, 3(1): 71–86, 1991.

[4] M.Kirby and L.Sirovich,(1990). Application of the Karhunen-Loève Procedure for the characterization of human faces. IEEE Trans. on Pattern Analysis and Machine Intelligence, 12(1): 103-108.

[5] V.Vapnik, Statistical learning theory. Chichester, GB:Wiley, 1998

[6] F.Riesz and B.Sz.-Nagy. Functional Analyses. (242-246).Ungar, New York, 1955

[7] A.Snika abd V,Schölkopf. Atutorial on support vector regression.NeuroColt 2 TR 1998-03,1998

[8] J.C.Platt. Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods, MIT press, 271-284,1999

[9] S.K.Shevade, S.S.Keerthi, C.Bhattacharyya & K.R.K. Murthy, Improvements to SMO Algorithm for SVM Regression, Technical Report CD-99-16.

[10] V. Vapnik, S. Golowich and A. Smola. 1997. *Support Vector Method for Function Approximation*, Regression Estimation, and Signal Processing. In: M. Mozer, M. Jordan, and T. Petsche (eds.): Neural Information Processing Systems, Vol. 9. MIT Press, Cambridge, MA, 1997

[11] Yuan Yaxiang and Sun Wenyu, Optimization Theory and Method (In Chinese), 422-454, Science Press1999,

[12] ftp://ftp.uk.research.att.com/pub/data/att_faces.tar.Z

[13] Kenneth R. Castleman Digital Image Processing, (pp.294-297), Tsinghua University press 1998.

[14] P. J. Phillips. Support vector machines applied to face recognition. In M. I. Jordan,M. J. Kearns, and S. A. Solla, editors, Advances in Neural Information Processing Systems 11, 1998.