Computer Sciences Department

An Evaluation of Bayes and Large Margin Classifiers for Face Expression Recognition

Guodong Guo
Charles R. Dyer

Technical Report #1447

October 2002

UNIVERSITY OF
WISCONSIN
MADISON

An Evaluation of Bayes and Large Margin Classifiers for Face Expression Recognition

Guodong Guo and Charles R. Dyer

Computer Sciences Department

University of Wisconsin-Madison

E-mail: gdguo@cs.wisc.edu, dyer@cs.wisc.edu

Version: Feb. 6, 2002

In this paper, we investigate three representative methods for face expression recognition. The first one is the Bayes decision approach, which is the most classical algorithm for general pattern recognition. The second is support vector machine (SVM) classification, and the third is the AdaBoost method. Both SVM and AdaBoost are considered Large Margin Classifiers. We evaluate these three methods for face expression recognition on a common database. To solve the multi-class (7 expressions) recognition problem, we use a voting scheme and a binary tree scheme. For the Bayes and AdaBoost methods, we use a pairwise framework for both feature selection and discrimination in order to simplify the problem, and get good results. In contrast, with SVMs, we use all the features without selection. We compare linear and non-linear SVMs to see if there is any improvement using non-linear mapping. We also find that normalization makes recognition performance worse for SVMs but has no influence for Bayes and AdaBoost methods.

Key Words: Face expression recognition (FER), Gabor wavelets, Bayes decision, large margin classifiers, AdaBoost, support vector machine (SVM), pairwise recognition, feature selection, statistical learning.

1. INTRODUCTION

Face expression recognition (FER) by computer is very useful for many applications such as human behavior understanding, perceptual user interfaces, and interactive computer games. In an automatic FER system, the first step is face detection or localization in a cluttered scene. Next, relevant features from the face must be extracted, and finally the expression can be classified based on the extracted features. Unlike face recognition, FER focuses on how to discern the same expressions from different individuals. Because different people may show the same expression in different ways, the FER problem is very challenging.

There are two versions of the face expression recognition problem depending on whether an image sequence is the input and the dynamic characteristics of expressions are analyzed, or a single image is the input and expressions are distinguished based on static differences. Previous work on dynamic expression recognition includes the following. Suwa *et al.* [28] analyzed dynamic facial expressions by tracking the motion of twenty markers. Mase [22] computed first- and second-order statistics of optical flow at evenly divided small blocks. Yacoob and Davis [37] used the inter-frame motion of edges extracted in the areas of the mouth, nose, eyes, and eyebrows. Bartlett *et al.* [1] combined optical flow and principal components obtained from image differences. Essa and Pentland [8] built a dynamic parametric model by tracking facial motion over time. Donato *et al.* [6] compared several methods for feature extraction, and found that Gabor wavelet coefficients and independent component analysis (ICA) gave the best representation. Tian *et al.* [30] tracked upper and/or lower face action units over sequences to construct their parametric models.

There has also been considerable previous work on face expression recognition

from a single image. Padgett and Cottrell [23] used seven pixel blocks from feature regions to represent expressions. Cottrell and Metcalfe [4] used principal component analysis and feed-forward neural networks. Rahardja *et al.* [26] used a pyramid structure with neural networks. Lanitis *et al.* [17] used parameterized deformable templates to represent face expressions. Lyons *et al.* [19] [20] and Zhang *et al.* [39] [38] demonstrated the advantages of using Gabor wavelet coefficients to code face expressions. See [24] for a good review of different approaches for face expression recognition.

In this paper we investigate face expression recognition from static images and use Gabor filters for facial feature extraction. Our major focus is on the evaluation of some new methods for face expression recognition. Recently, large margin classifiers such as support vector machines (SVMs) and AdaBoost were proposed from machine learning society, and have been used for solving some vision problems. Here we are interested to see if they are useful for face expression recognition. To our knowledge, it is the first time to evaluate the large margin classifiers for face expression recognition. On the other hand, the Bayes classifier is a classical method for pattern recognition. We describe a simplified Bayes classifier for face expression recognition, and use its results as a baseline for comparison.

Our approach to face expression recognition is summarized in Fig. 1. Each input face image is convolved with 18 Gabor filters and results in 18 filtered images. The amplitude of each filtered image at selected fiducial points are used as feature values. The organization of the paper is as follows. In Section 2, the image database, Gabor filter bank design, and feature extraction methods are described. In Sections 3, 4, and 5, we describe the three classifiers to be compared. In Sections 6 and 7 we present two strategies for multi-class expression recognition, and a pairwise

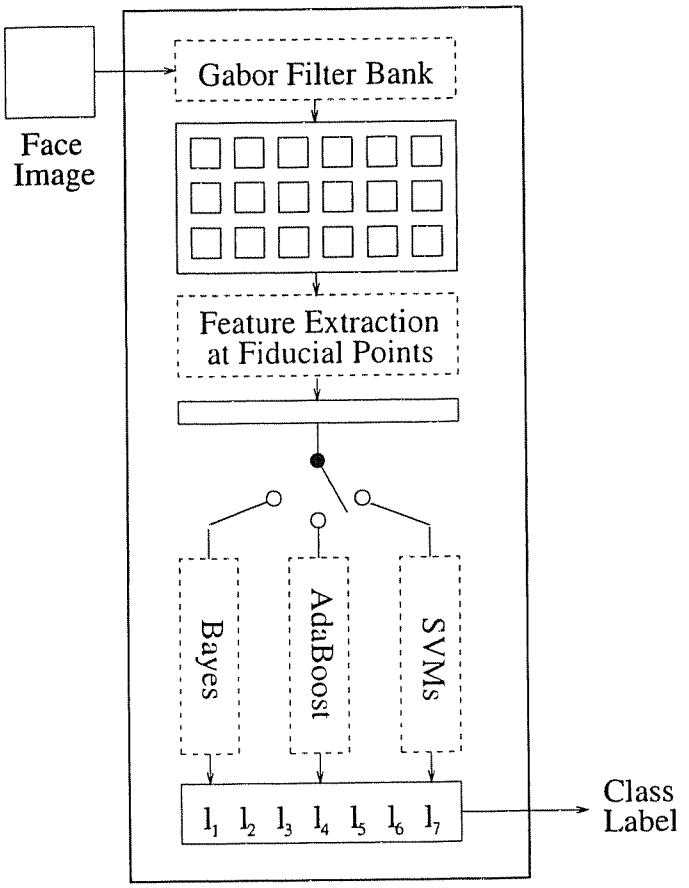


FIG. 1 The framework of the face expression recognition system. Dashed blocks represent operations while solid blocks contain the filtered images, feature vector, and final class labels.

framework for feature selection. Experimental results and comparisons are given in Section 8. Finally, we give some discussions on the results in Section 9.

2. FACIAL FEATURE EXTRACTION

The face database [19] used in our experiments contains 213 images of 10 Japanese women. Each person has two to four images for each of seven expressions: neutral, happy, sad, surprise, anger, disgust, and fear. Each image size is

256 x 256 pixels. A few examples are shown in Fig. 2. For details on the database such as image collection, data description, and human ranking, see [19]. Other researchers who have also used this database include [20] [39] [38].

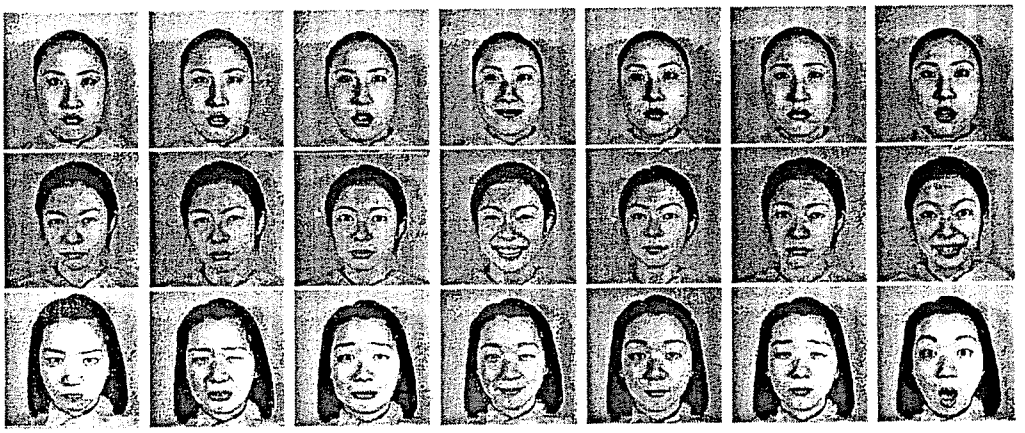


FIG. 2 Some images in the face expression database. From left to right, the expressions are angry, disgust, fear, happy, neutral, sad, and surprise.

Given a set of points detected or marked on a face image, two approaches to facial feature extraction are to use either (1) the geometric positions of the fiducial points, or (2) the Gabor filter coefficients [5] at the fiducial points. It was shown in [20] [39] that the filter coefficients can characterize face expressions better than the geometric positions. Therefore, in our study, we use Gabor filtering for facial feature extraction.

2.1. The Gabor Filter Bank

A two-dimensional Gabor function, $g(x, y)$, and its Fourier transform, $G(u, v)$, can be written as

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi j W x\right] \tag{1}$$

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \tag{2}$$

where W is the frequency of a sinusoidal plane wave along the x-axis, and σ_x and σ_y are the space constants of the Gaussian envelope along the x and y axes, respectively. $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$. Filtering a signal with this basis provides a localized frequency characterization. Filters with arbitrary orientations can be obtained by a rigid rotation of the x-y coordinate system:

$$g'(x, y) = g(x', y'), \tag{3}$$

where

$$x' = x\cos\theta + y\sin\theta, \quad y' = -x\sin\theta + y\cos\theta, \tag{4}$$

and θ is the rotation angle.

In earlier applications of the Gabor filtering technique [5] for face recognition [16] [36] and face expression classification [19] [20] [39] [38], investigators have only varied the scale and orientation of the filters, but kept the Gaussian envelope parameter σ fixed to π or 2π. This methodology is questionable because the area of the energy distribution of the filters varies with scale, so the Gaussian envelope should vary with the filter size. Consequently, we designed the Gabor filter bank based on the filters used perviously for texture segmentation and image retrieval [14] [21].

The Gabor filter bank is designed to cover the entire frequency spectrum [14] [21]. In other words, the Gabor filter set is constructed such that the half-peak magnitude of the filters in the frequency spectrum touch each other. This results in the following formulas to compute the filter parameters σ_u and σ_v:

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{S-1}}, \quad W = a^m U_l, \tag{5}$$

$$\sigma_u = \frac{(a-1)W}{(a+1)\sqrt{2\ln 2}} \tag{6}$$

$$\sigma_v = \tan\left(\frac{\pi}{2K}\right)\left[W - \frac{(2\ln 2)\sigma_u^2}{W}\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{W^2}\right]^{-\frac{1}{2}} \tag{7}$$

where U_l and U_h denote the lower and upper center frequencies of interest. $m \in \{0, 1, \ldots, S-1\}$ and $n \in \{0, 1, \ldots, K-1\}$, are the indices of scale and orientation, respectively. K is the number of orientations and S is the number of scales.

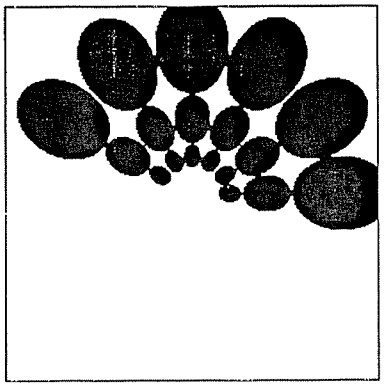


FIG. 3 The filter set in the spatial-frequency domain. There are a total of 18 Gabor filters shown at half-peak magnitude.

In our experiments we used $U_h = \sqrt{2}/4$, $U_l = \sqrt{2}/16$, and used three scales ($S = 3$) and six orientations ($K = 6$). The half-peak supports of the Gabor filter bank are shown in Figure 3. The differences in the strength of the responses of different image regions is the key to the multi-channel approach to face image analysis.

2.2. Feature Extraction

After Gabor filtering, the amplitude values at selected fiducial points on the face images are used as the features. Automatically extracting these points [16] [36] is still an open problem [39]. In order to focus this study on the classifier performance, we manually marked the fiducial points in each image. Typical positions of 34 fiducial points are shown in Figure 4. Thus for each face image, the extracted feature vector is of dimension 612 (34 x 3 x 6).

FIG. 4 34 fiducial points on a face image.

2.3. Data Normalization

There are two common ways for performing data normalization. The first is very simple, just shifting each component to a given range, for example between 0 and 1:

$$\widetilde{\mathbf{x}}_{ik} = \frac{\mathbf{x}_{ik} - \min_k}{\max_k - \min_k} \tag{8}$$

where $\max_k$ and $\min_k$ correspond to the biggest and smallest values in dimension k of the training data, respectively, and $\mathbf{x}_{ik}$ is the ith feature vector in dimension k. This approach is not robust because some noisy, bigger values may overshadow the smaller, but real, data. A better approach is to use Gaussian normalization, by computing the mean, μ_k, and standard deviation, σ_k, in each dimension of the training data. Then normalize the original data to a $N(0,1)$ distribution as follows:

$$\widetilde{\mathbf{x}}_{ik} = \frac{\mathbf{x}_{ik} - \mu_k}{\sigma_k} \tag{9}$$

We chose to use the second approach in our data normalization. In addition to normalizing the training data, the test data was also normalized using the same μ_k and σ_k.

After processing the data, we trained the classifiers and then used the test data for face expression recognition experiments. We leave the problem of feature selection to Section 7 after the introduction of the classifiers and the multi-class classification schemes.

3. BAYES CLASSIFIER

The Bayes classifier yields the minimum error rates when the underlying probability density function (pdf's) are known [10]. The *a posteriori* probability of pattern x belonging to class ω_c is given by Bayes' rule:

$$P(\omega_c|\mathbf{x}) = \frac{P(\omega_c)p(\mathbf{x}|\omega_c)}{p(\mathbf{x})} \tag{10}$$

where $P(\omega_c)$ is the *a priori* probability, $p(\mathbf{x}|\omega_c)$ the conditional probability density function of ω_c, and $p(\mathbf{x})$ is the mixture density. The maximum *a posteriori* (MAP) decision is

$$\omega_c^* = \arg\max_c P(\omega_c|\mathbf{x}), \quad c = 1, 2, \ldots, C \tag{11}$$

The Bayes classifier can be used for both two-class and multi-class classifications.

In face expression recognition there are often not enough samples to reliably estimate the conditional density function for each class. A compromise is to assume that the within-class densities can be modelled as normal distributions, and all the within-class covariance matrices are identical and diagonal. Liu and Wechsler [18] used this simplification for face recognition. Here we evaluate this approach for the problem of face expression recognition. The parameters of the normal distributions

are estimated as follows,

$$\mu_c = \frac{1}{N_c} \sum_{j=1}^{N_c} \mathbf{x}_j^{(c)}, \quad c = 1, 2, \ldots, C \tag{12}$$

where $\mathbf{x}_j^{(c)}$, $j = 1, 2, \ldots, N_c$, represents the samples from class ω_c, and

$$\Sigma_I = \Sigma_c = diag\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_D^2\} \tag{13}$$

where D is the feature dimension. Each component σ_i^2 can be estimated by the sample variance in the one-dimensional feature subspace

$$\sigma_i = \frac{1}{C} \sum_{c=1}^{C} \left\{ \frac{1}{N_c - 1} \sum_{j=1}^{N_c} \left(x_{ji}^{(c)} - \mu_{ci} \right)^2 \right\} \tag{14}$$

where $x_{ji}^{(c)}$ is the ith element of the sample $\mathbf{x}_j^{(c)}$, μ_{ci} the ith element of μ_c, and C the number of classes.

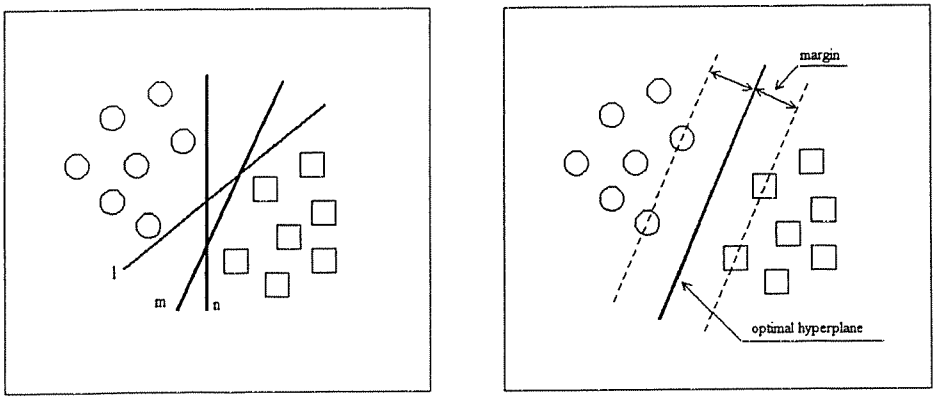
4. SUPPORT VECTOR MACHINE

4.1. Basic Theory of Support Vector Machines

Given a set of training vectors belonging to two separate classes, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, where $\mathbf{x}_i \in R^n$ and $y_i \in \{-1, +1\}$, one wants to find a hyperplane $\mathbf{wx} + b = 0$ to separate the data. Fig. 5(a) shows an example and several possible hyperplanes, but there is only one (shown in Fig. 5(b)) that maximizes the margin (i.e., the distance between the hyperplane and the nearest data point of each class). This linear classifier is called the optimal separating hyperplane (OSH).

The solution to the optimization problem of SVMs is given by the saddle point of the Lagrange functional,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{l} \alpha_i \{ y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] - 1 \} \tag{15}$$



(a) (b)

FIG. 5 Classification between two classes using hyperplanes: (a) Arbitrary hyperplanes l, m and n. (b) The optimal separating hyperplane with the largest margin identified by the dashed lines, passing the support vectors.

where α_i are the Lagrange multipliers. Classical Lagrangian duality enables the *primal* problem (15) to be transformed to its *dual* problem, which is easier to solve. The solution is given by

$$\bar{\mathbf{w}} = \sum_{i=1}^{l} \bar{\alpha}_i y_i \mathbf{x}_i, \qquad \bar{b} = -\frac{1}{2}\bar{\mathbf{w}} \cdot [\mathbf{x}_r + \mathbf{x}_s] \tag{16}$$

where $\mathbf{x}_r$ and $\mathbf{x}_s$ are any two support vectors with $\bar{\alpha}_r, \bar{\alpha}_s > 0$, $y_r = 1$, and $y_s = -1$.

To solve a non-separable problem, Cortes and Vapnik [3] introduced slack variables $\xi_i \geq 0$ and a penalty function, $F(\xi) = \sum_{i=1}^{l} \xi_i$, where the ξ_i measure the mis-classification error. The solution is identical to the separable case except for a modification of the Lagrange multipliers as $0 \leq \alpha_i \leq M$, $i = 1, \ldots, l$. The choice of M is not strict in practice, and we set $M = 100$ in all our experiments. See [33] for more details on the non-separable case.

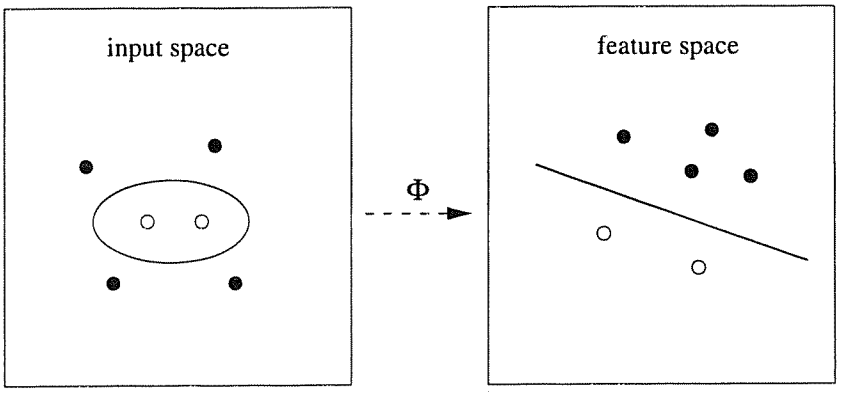


FIG. 6 Nonlinear mapping of an SVM from the input space to a high-dimensional feature space.

4.2. Non-Linear SVM

SVMs can realize non-linear discrimination by kernel mapping [33]. When the samples in the input space can not be separated by any linear hyperplane, they may be linearly separated in a non-linearly mapped feature space. Note that here the feature space of the SVMs is different from the image feature space.

There are a few kernel functions that have been used previously for nonlinear mapping [33], with the Gaussian radial basis function (GRBF) the most commonly used. We used the GRBF kernel in our experiments with the form $K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{(\mathbf{x}-\mathbf{y})^2}{\gamma^2}\right)$, where parameter γ is the width of the Gaussian function.

For a given kernel function, the SVM classifier is now given by

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{l} \bar{\alpha}_i y_i K(\mathbf{x}_i, \mathbf{x}) + \bar{b}\right) \tag{17}$$

5. ADABOOST

Boosting is a method for combining a collection of weak classification functions (weak learners) to form a stronger classifier. AdaBoost is an adaptive algorithm

that boosts a sequence of classifiers, in that the weights are updated dynamically according to the errors in previous learning [9]. AdaBoost belongs to the class of large margin classifiers. The original AdaBoost method [9] works on all given features. Recently, Tieu and Viola [31] adapted the AdaBoost algorithm for natural image retrieval, and later for face detection [34]. They let the weak learner work using a single feature at a time. So after T rounds of boosting, T features are selected together with the T weak classifiers. Tieu and Viola's AdaBoost algorithm [31] is briefly described below:

AdaBoost Algorithm

Input: 1) n training examples, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, with $y_i = 1$ or 0;

2) the number of iterations, T.

Initialize weights $w_{1,i} = \frac{1}{2l}$ or $\frac{1}{2m}$ for $y_i = 1$ or 0, respectively, with $l + m = n$.

Do for $t = 1, \ldots, T$:

1. Train one hypothesis h_j for each feature j with w_t, and error $\epsilon_j = Pr_i^{w_t}[h_j(x_i) \neq y_i]$.

2. Choose $h_t(\cdot) = h_k(\cdot)$ such that $\forall j \neq k, \epsilon_k < \epsilon_j$. Let $\epsilon_t = \epsilon_k$.

3. Update: $w_{t+1,i} = w_{t,i}\beta_t^{e_i}$, where $e_i = 1$ or 0 for example x_i classified correctly or incorrectly, respectively, with $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ and $\alpha_t = \log \frac{1}{\beta_t}$.

4. Normalize the weights so they are a distribution, $w_{t+1,i} \longleftarrow \frac{w_{t+1,i}}{\sum_{j=1}^{n} w_{t+1,j}}$.

Output the final hypothesis

$$h_f(x) = \begin{cases} 1 & if \quad \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases} \tag{18}$$

6. MULTI-CLASS CLASSIFICATION

Previous sections described the basic theory of Bayes, SVMs and AdaBoost for two-class classification. It is very easy to extend Bayes decision to multi-class cases, but difficult for SVM and AdaBoost. With SVM and AdaBoost, a multi-class system can be built from two-class classifiers. There are two main schemes used for this purpose. One is the one-against-all strategy to classify each class against all the remaining classes; The other scheme is the one-against-one strategy that classifies between each pair. For the former, the number of positive and negative examples are extremely unbalanced. Hence we take the latter approach. The problem becomes how to combine the binary classification results to obtain the final decision. A classical method is to use a voting strategy that considers all pairs of classes, hence there will be $C(C-1)/2$ comparisons. Another strategy is to use a binary tree structure to get the final classification, where the total number of comparisons is $C-1$. We use the voting scheme for pairwise Bayes decision and AdaBoost because the $C(C-1)/2$ comparisons are not a heavy computational burden for them. For SVMs, however, we use a binary tree structure to reduce the number of comparisons required by this more computation-intensive classifier.

6.1. Voting Scheme

For each test example, there are a total of $C(C-1)/2$ pairs of classifications computed. The output of the $C(C-1)/2$ classifiers is used to construct a matrix, as shown in Fig. 7. Each element is equal to 1 or 0, where $\phi_{i,j}(\mathbf{x}) = 1$ if $\mathbf{x}$ is classified as class i and $\phi_{i,j}(\mathbf{x}) = 0$ if $\mathbf{x}$ is classified as class j. All elements on the main diagonal are zeros.

The outputs of the pairwise classifiers are combined to obtain the final decision.

$$\begin{pmatrix} 0 & \phi_{1,2} & \phi_{1,3} & \cdots & \phi_{1,C} \\ \phi_{2,1} & 0 & \phi_{2,3} & \cdots & \phi_{2,C} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \phi_{C,1} & \phi_{C,2} & \phi_{C,3} & \cdots & 0 \end{pmatrix}$$

FIG. 7 The pairwise classification results $\phi_{i,j}$ are listed in a $C \times C$ matrix for the C class classification problem. The values of $\phi_{i,j}$ are equal to 1 or 0. If $\phi_{i,j} = 1$, $\phi_{j,i} = 0$.

In the voting scheme, a count $c(\omega_i|\mathbf{x})$ of the number of pairwise classifiers that label $\mathbf{x}$ in class ω_i is calculated by

$$c(\omega_i|\mathbf{x}) = \sum_j \phi_{i,j}(\mathbf{x}) \tag{19}$$

The input $\mathbf{x}$ is assigned the class label for which the count is maximum, i.e.,

$$\omega_c^* = \arg\max_i c(\omega_i|\mathbf{x}), \quad i = 1, 2, \cdots, C \tag{20}$$

6.2. Binary Tree Scheme

In order to reduce the number of comparisons for SVMs in multi-class cases, we construct bottom-up a binary tree that has been used successfully in multi-class face recognition [12]. Suppose there are seven classes in the data set, the decision tree is shown in Fig. 8, where the numbers 1-7 encode the class labels. The numbers are arbitrary without any meaning associated with the ordering. From a comparison between each pair, one class is chosen representing the "winner" of the current two classes. The selected classes (from the lowest level of the binary tree) move up to the next higher level for another round of tests. Finally, a unique class label

appears at the top of the tree. This process corresponds to a single-elimination tournament.

The binary tree structure reduces the number of comparisons to $C - 1$ instead of the $C(C-1)/2$ comparisons in the voting scheme. This benefit is especially useful when the number of classes is very large. If a test image can be classified correctly, the output of the binary tree is the same no matter how it is arranged. On the contrary, when a test example is classified incorrectly, the output depends, in general, on the ordering of the classes at the leaves.

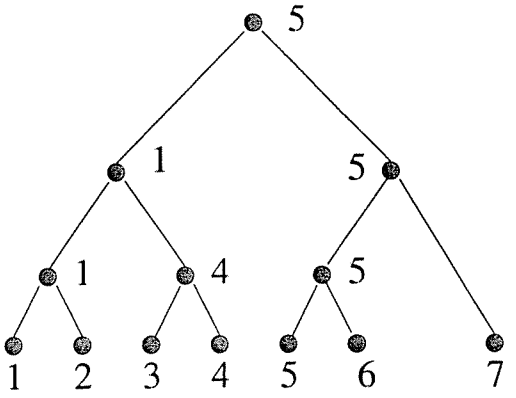


FIG. 8 The binary tree structure of a 7-class classification problem. At each non-leaf node a comparison is made between the classes at its two child nodes, and the winner class is then associated with the current node. This process continues until a class is selected at the root node.

Note that although there are only $C - 1$ comparisons, SVMs still need to be trained using $C(C-1)/2$ pairs because the classes associated with non-leaf nodes can not be determined in advance, and instead depend on the test example.

7. PAIRWISE FEATURE SELECTION

In the representation of face expression images using a bank of Gabor filters, the issue of selecting a good subset of the extracted features (in our case 612) must be addressed. Traditionally, feature selection is defined as follows: given a set of candidate features, select a subset that performs best under some classification system [15]. In the past decade, many researchers have studied search algorithms for feature selection. Jain and Zongker [15] evaluated different search algorithms for feature subset selection and found that the sequential forward floating selection (SFFS) algorithm proposed by Pudil *et al.* [25] performed best. However, SFFS is very time consuming when the number of features is large. Vailaya [32] reported that using SFFS to select 67 features from 600 for a two-class problem (indoor vs. outdoor images), took 12 days to compute. Moreover, it is difficult to select the best features when the number of training samples is small [15], which is often the case for face expression recognition.

Because of the difficulties of using traditional feature selection methodologies for face data, we propose another idea called "feature ranking" to distinguish it from feature selection. In feature ranking, all features are assumed independent and a criterion is used to compare the discriminative capabilities of each feature. The feature ranking approach simplifies and speeds up the process to select a subset of the features. We will select the best features for discriminating between each pair of expression types. The pairwise framework was first presented in [13] for face recognition because the features useful for a pair of classes may not be appropriate for some other pair of classes. A simple criterion,

$$r_{ij}^{d} = \frac{\| \mu_{id} - \mu_{jd} \|}{\sigma_d} \tag{21}$$

is used to rank the d^{th} feature for $d = 1, 2, \cdots, D$, in discriminating between expression classes i and j. μ_{id} (μ_{jd}) is the mean value of class i (j), and σ_d is the variance of the samples of the d^{th} feature. The larger the value of r_{ij}^d, the more discriminative the dth feature for distinguishing between classes i and j.

Using r_{ij}^d to rank the features in descending order according to their discriminability, combined with a user-specified number, N, of features to use, the system selects the first N features to train the classifiers and also to recognize a new face.

Feature ranking is executed for each pair of classes. It provides some knowledge regarding the importance of certain features over others for a specific classification problem. The top features in the list are expected to have higher discrimination capability. Using fewer number of features reduces dimensionality without losing much discrimination power.

The top N features are used by the Bayes classifier during training. In AdaBoost, features are selected one by one according to the classification error of the weak learner in the previous step [31]. For SVMs, feature selection is not trivial [11]; while the general framework of pairwise feature selection [13] should work, the problem is how to select the features for the SVMs in each pair of classes. The ranking strategy is too simple to select appropriate features for SVMs. The reason may be because of the difference in the optimization criterion used in SVMs versus other methods such as a Bayes classifier. This means that some features may be good for a Bayes classifier but poor for an SVM, which requires a nice representation for the margin between two classes. Consequently, in our study, all the features were used for SVMs in both training and classification. Some appropriate feature selection methods [35] for SVMs can be explored in the future for use in face expression recognition.

8. EXPRESSION RECOGNITION EXPERIMENTS

In this section, we evaluate the three classifiers for face expression recognition and compare their performance based on recognition accuracy. In addition, we also compare our experimental results with previous approaches that used the same database.

8.1. Experimental Results

Our experimental procedure used 10-fold cross validation because the face database is relatively small (213 image). That is, the face expression database was divided randomly into 10 parts, from which the data from 9 parts were used for training the classifiers and the last part was used for testing. We do this kind of data separation for 10 runs, and the average recognition accuracy over these 10 runs is reported as the final recognition accuracy for each method.

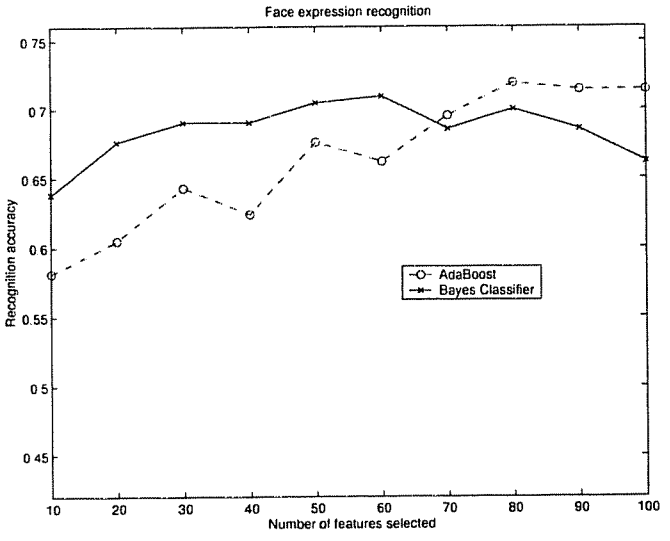


FIG. 9 Experimental results of the simplified Bayes decision and Adaboost as a function of the number of selected features.

For the given training data, each of the three classifiers was trained separately and independently. The input parameters required were minimal. Specifically, the only parameter for the Bayes classifier and AdaBoost was the number of features or the number of boosting rounds. For linear SVM, the parameter was set to $M = 100$. For non-linear SVMs with the GRBF kernel, we experimentally set the width parameter to obtain the best performance. A pairwise framework is used in feature selection and classification for both the Bayes classifier and AdaBoost.

First, the Bayes classifier and the AdaBoost algorithm were compared. Fig. 9 shows the results as the number of features was varied in both methods. It is clear that only a small number of features is sufficient for both algorithms. The Bayes classifier reached its best performance of 70.95% with 60 features, and the performance deteriorated slightly if more features were used. Using all 612 features the recognition accuracy was 63.33% (shown in Fig. 10 as B612, which means Bayes using all features). This demonstrates that over-fitting is a serious problem for the Bayes method, and indicates that feature selection is necessary for this classifier.

For the AdaBoost method, peak performance was 71.9% using 80 features. As shown in Fig. 9, using more features slightly lowered recognition accuracy. In Fig. 10, the recognition results of various approaches are shown, where B60 means Bayes using 60 features, and A80 means AdaBoost using 80 (boosted) features. Note that the number of features in Fig. 9 means how many features used for each pair of classes, but which features were used for each pair was generally not the same. We also tested recognition performance using normalized features (Eq. 9), but this made no difference for either Bayes or AdaBoost.

We evaluated SVMs using the binary tree tournament scheme, and show results in the right half of Fig. 10, where SVM-L denotes linear SVM, and nSVM-L means

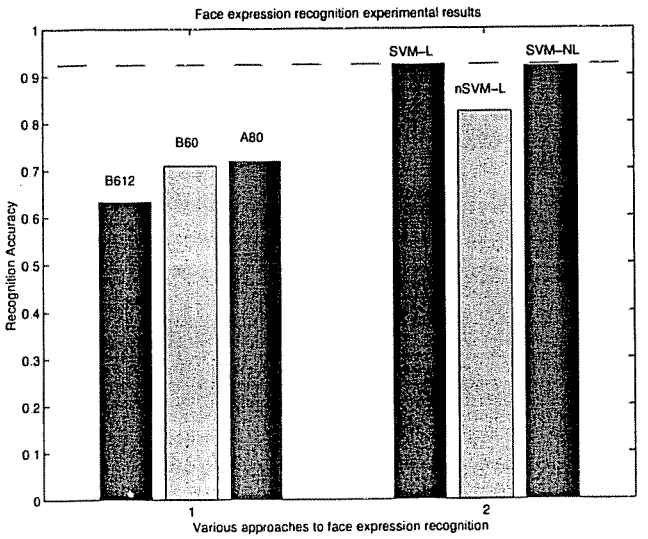


FIG. 10 Experimental results of various approaches for face expression recognition on a common database. See context for the description of each approach.

linear SVM classification using normalized features. Normalization reduced recognition accuracy for linear SVMs. Specifically, the recognition accuracy of SVM-L was 92.4% while it was 82.4% for nSVM-L. It was expected that recognition performance would be improved or at least not become worse with data normalization. But here we get the opposite situation for linear SVMs. Why does data normalization of face expression features deteriorate the performance for linear SVMs, and have no positive effect on Bayes and AdaBoost? It is difficult to address this problem theoretically. Our intuitive interpretation is that normalization moves the point positions in feature space that are sensitive to the margin computation, hence reducing the generalization capability for the SVM classifiers. Besides linear SVMs, we also experimented with non-linear SVMs on the same problem, termed SVM-NL in Fig. 10, in which there is one parameter, i.e., the Gaussian width γ of the GRBF kernel function. We chose the parameter γ experimentally to a value (can

take any value between 5 and 7) where recognition results were best. The accuracy of SVM-NL was 91.9%, which is comparable to the performance by linear SVMs. This indicates again the special characteristics of the face expression data. In other words, the non-linear mapping using the GRBF kernel function showed no benefit for face expression recognition using this data set.

Comparing all three classifiers together, we conclude from Fig. 10 that the recognition accuracies of AdaBoost and Bayes are comparable, and their performance is worse than that based on linear or non-linear SVMs. Overall, linear SVMs without data normalization give the highest recognition accuracy in our experiments.

Finally, we did not quantitatively compare the computation times for each algorithm. Roughly speaking, the Bayes classifier is fastest and SVMs are the slowest, with AdaBoost taking more time than the Bayes approach, but it is much faster than SVMs.

8.2. Comparison with Previous Approaches

Besides comparisons among these three classifiers, we also compared recognition performance with other methods [39] [38] [20] that used the same database. In [39] [38] a neural network approach was used. With the same evaluation criterion, their reported result was 90.1%, which is higher than the Bayes and AdaBoost approaches, but lower than the 92.4% by linear SVMs and the 91.9% by non-linear SVMs. They also removed certain problematic images in the database, and reported a recognition accuracy of 92.2%, which is still not higher than linear SVMs applied to the entire database. This demonstrates the good generalization capability of linear SVMs on face expression data.

In [20], a result of 92% using linear discrimination analysis (LDA) was reported,

but they only included 9 people's face images, hence only 193 instead of all 213 images were used. In conclusion, linear SVMs have good generalization capability for face expression recognition compared with other classifiers.

9. DISCUSSION AND FUTURE RESEARCH DIRECTIONS

The Bayes classifier is a classical technique, however with a small database, it is difficult to estimate the co-variance matrix. Assuming an independently identical distribution (i.i.d.), we simplified the problem of class density estimation. From our experiments, the problem of over-fitting is a serious one when using Bayes with high dimensional data. The pairwise feature selection framework improves the performance to some extent.

Support Vector Machines have been applied recently to vision problems such as face detection and recognition. Our experiments further demonstrate their good generalization ability for face expression recognition. The recognition accuracy of SVMs is high and it can be taken as a benchmark for comparing other methods for face expression recognition. On the other hand, we have some new observations. Why did data normalization reduce the performance of SVMs on face expression images so much? In addition, why does the non-linear GRBF kernel function not show any improvement over linear SVMs? It may be necessary to do a deeper analysis of kernel SVMs for face expression data.

The AdaBoost algorithm is a relatively new method for solving computer vision problems. Our experiments on face expression recognition show that AdaBoost does not currently produce high recognition accuracy. But this does not mean the demise of the AdaBoost approach. On the contrary, more research on boosting-like techniques should be investigated.

Feature selection is a classical problem in pattern recognition. Our pairwise framework simplifies the problem of both feature selection and classification. The question is how to select relevant features for each pair of classes. Our simple feature ranking approach worked well for Bayes and AdaBoost (in which feature selection is incorporated into the boosting process). However, how to select features for SVMs is still an open problem. For example, it has been shown that feature selection for SVMs for image retrieval is difficult [11]. Recently, Weston *et al.* [35] addressed the feature selection problem for SVMs, but it is not clear if that method will work for a large variety of data. Different feature selection techniques may be necessary for SVMs to work on different types of data.

10. CONCLUSIONS

We have investigated experimentally three representative classifier methods: Bayes, SVM, and AdaBoost for face expression recognition. Linear SVMs without feature selection and without data normalization gave the best recognition accuracy. Simple feature selection using a pairwise class discrimination framework works well for the Bayes and AdaBoost methods. But more work is needed to improve the performance of Bayes and AdaBoost so they are comparable to SVMs for face expression recognition.

11. ACKNOWLEDGEMENTS

The authors are grateful to M. Lyons for providing us with the face expression database. The support of the National Science Foundation under Grant No. IIS-9988426 is gratefully acknowledged.

REFERENCES

[1] M. Bartlett, P. Viola, T. Sejnowski, L. Larsen, J. Hager, and P. Ekman, Classifying facial action, in D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, MIT Press, Cambridge, MA, 1996.

[2] C. J. C. Burges, Simplified support vector decision rules, in *Proc. 13th Int. Conf. Machine Learning*, 71-77, 1996.

[3] C. Cortes and V. Vapnik, Support vector networks, *Machine Learning*, 20, 273-297, 1995.

[4] G. Cottrell and J. Metcalfe, Face, gender and emotion recognition using holons. In *Advances in Neural Information Processing Systems 3*, 564-571, 1991.

[5] J, Daugman, Uncertainty relation for resolution in space, spatial frequency and orientation optimized by tow-dimensional visual cortical filters, *J. Optical Society of America, vol. A*, 1160-1169, 1985.

[6] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, Classifying facial actions, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(10), 974 -989, 1999.

[7] R. P. W. Duin, Classifiers in almost empty spaces, *Proc. of Int. Conf. on Pattern Recognition*, vol. 2, 1-7, 2000.

[8] I. Essa and A. Pentland, Coding, analysis, interpretation, and recognition of facial expressions, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7), 757-763, 1997.

[9] Y. Freund and R. E. Schapire, A decision-theoretic generalization of online learning and an application to boosting. *J. Comp. and Sys. Sci.*, 55(1), 119-139, 1997.

[10] K. Fukunage, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, second edition, 1991.

[11] G. D. Guo, A. K. Jain, W. Y. Ma, and H. J. Zhang, Learning similarity measure for natural image retrieval with relevance feedback, *Proc. CVPR*, vol. 1, 731-736, 2001.

[12] G. D. Guo, S. Z. Li, and K. L. Chan, Face recognition by support vector machines, *Image and Vision Computing*, 19(9-10), 631-638, 2001.

[13] G. D. Guo, H. J. Zhang, and S. Z. Li, Pairwise face recognition, *Proc. ICCV*, vol. 2, 282-287, 2001.

[14] A. K. Jain and F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, *Pattern Recognition*, 16(12), 1167-1186, 1991.

[15] A. Jain and D. Zongker, Feature selection: Evaluation, application, and small sample performance, *IEEE Trans. PAMI*, 19(2), 153-158, 1997.

[16] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Trans. Computers*, 42(3), 300-311, 1993.

[17] A. Lanitis, C. Taylor, and T. Cootes, Automatic interpretation and coding of face images using flexible models, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7), 743-756, 1997.

[18] C. Liu and H. Wechsler, Probablistic reasoning models for face recognition, *Proc. CVPR*, 827-832, 1998.

[19] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, Coding facial expressions with gabor wavelets. In *Proc. Third IEEE Int. Conf. Automatic Face and Gesture Recognition*, 200-205, 1998.

[20] M. J. Lyons, J. Budynek, and S. Akamatsu, Automatic Classification of single facial images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(12), 1357-1362, 1999.

[21] B. S. Manjunath and W. Y. Ma, Texture features for browsing and retrieval of image data. *IEEE Trans. PAMI*, 18(8), 837-842, 1996.

[22] K. Mase, Recognition of facial expression from optical flow. *IEICE Transactions E*, 74(10), 3473-3483, 1991.

[23] C. Padgett and G. Cottrell, Identifying emotion in static images, *Proc. 2nd Joint Symp. on Neural Computation*, vol. 5, 91-101, 1997.

[24] M. Pantie and L. J. M. Rothkrantz, Automatic analysis of facial expressions: The state of the art, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(12), 1424 -1445, 2000.

[25] P. Pudil, J. Novovicova and J. Kittler, Floating search methods in feature selection, *Pattern Recognition Letters*, 15, 1119-1125, 1994.

[26] A. Rahardja, A. Sowmya, and W. Wilson, A neural network aproach to component versus holistic recognition of facial expressions in images, In *Intelligent Robots and Computer Vision X: Algorithms and Techniques, SPIE Proceedings*, vol. 1607, 62-70, 1991.

[27] A. Samal and P. A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: A survey, *Pattern Recognition*, 25, 65-77, 1992.

[28] M. Suma, N. Sugie, and K. Jujimora, A preliminary note on pattern recognition of human emotional expression, *Proc. 4th Int. Joint Conf. on Pattern Recognition*, 408-410, 1978.

[29] B. Scholkopf, *et al.*, Input space versus feature space in kernel-based methods, *IEEE Trans. Neural Networks*, 10(5), 1000-1017, 1999.

[30] Y.-I. Tian, T. Kanade, and J. F. Cohn, Recognizing action units for facial expression analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(2), 97-115, 2001.

[31] K. Tieu and P. Viola, Boosting image retrieval, *Proc. Computer Vision and Pattern Recognition Conf.*, vol. 1, 228-235, 2000.

[32] A. Vailaya, Semantic classification in image database, Ph.D. thesis, Michigan State University, 2000.

[33] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[34] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, *Proc. CVPR*, vol. 1, 511-518, 2001.

[35] J. Weston, Feature selection for SVMs, *In Advances in Neural Information Processing Systems*, vol. 13, 2000.

[36] L. Wiskott, J. -M. Fellous, N. Kruger, and C. von der Malsburg, Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):775-779, 1997.

[37] Y. Yacoob and L. Davis, Recognizing facial expressions by spatio-temporal analysis, in *Proc. Int. Conf. Pattern Recognition*, vol. 1, 747-749, 1994.

[38] Z. Zhang, Feature-based facial expression recognition: Sensitivity analysis and experiments with a multi-layer perceptron, *Int. Journal of Pattern Recognition and Artificial Intelligence*, 13(6):893-911, 1999.

[39] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-lyaer perceptron. *Proc. Third IEEE Int. Conf. Automatic Face and Gesture Recognition*, 454-459, 1998.