Computer Vision and Image Understanding **84,** 264–284 (2001)
doi:10.1006/cviu.2001.0937, available online at http://www.idealibrary.com on IDEAL®

Face Detection Using Multimodal Density Models

Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja

Department of Computer Science and Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, Ilinois 61801
E-mail: mhyang@vision.ai.uiuc.edu, kriegman@cs.uiuc.edu, ahuja@vision.ai.uiuc.edu

Received September 7, 2000; accepted August 15, 2001

We present two methods using multimodal density models for face detection in gray-level images. One generative method uses a mixture of factor analyzers to concurrently perform clustering and, within each cluster, perform local dimensionality reduction. The parameters of the mixture model are estimated using the EM algorithm. A face is detected if the probability of an input sample is above a predefined threshold. The other discriminative method uses Kohonen's self-organizing map for clustering, Fisher's linear discriminant to find an optimal projection for pattern classification, and a Gaussian distribution to model the class-conditional density function of the projected samples for each class. The parameters of the class-conditional density functions are maximum likelihood estimates, and the decision rule is also based on maximum likelihood. A wide range of face images including ones in different poses, with different expressions and under different lighting conditions, is used as the training set to capture variations of the human face. Our methods have been tested on three data sets with a total of 225 images containing 871 faces. Experimental results on the first two data sets show that our generative and discriminative methods perform as well as the best methods in the literature, yet have fewer false detections. Meanwhile, both methods are able to detect faces of nonfrontal views and under more extreme lighting in the third data set.

Key Words: face detection; multimodal density estimation; Fisher's linear discriminant, self-organizing map; factor analysis; mixture of factor analyzers; EM algorithm.

1. INTRODUCTION

Images of human faces are central to intelligent human computer interaction. Many current research topics involve face images, including face recognition, face tracking, pose estimation, facial expression recognition, and gesture recognition. However, most existing solutions assume that human faces in an image or an image sequence have been identified and localized. To build fully automated systems that extract information from

1077-3142/01 $35.00

images with human faces, it is essential to develop robust and efficient algorithms to detect faces.

Given a single image or a sequence of images, the goal of face detection is to identify and locate all of the human faces regardless of their positions, scales, orientations, poses, expressions, occlusions and lighting conditions. This is a challenging problem because faces are nonrigid objects with a high degree of variability in size, shape, color, texture, facial hair, jewelry, makeups, and glasses. Most recent face detection methods can only detect upright, frontal faces under certain lighting conditions. Since the images of a human face lie in a complex subset of the image space that is unlikely to be modeled by a single linear subspace or characterized by a unimodal probability density function, we use multimodal density models to estimate the distribution of face and nonface patterns. Although some methods [19, 35] have applied mixture models for face detection, these use principal component analysis (PCA) for projection, which does not find the optimal subspace maximizing class separation.

Statistical pattern recognition approaches for face detection generally fall into two major categories, generative or discriminative methods, depending on the estimation criteria used for adjusting the model parameters and/or structure. Generative approaches such as Markov random field (MRF) [24], naive Bayes classifier [32], hidden Markov Model (HMM) [25], and higher-order statistics [25] rely on estimating a probability distribution over examples using maximum likelihood (ML) or maximum a posterior (MAP) methods, whereas discriminative methods such as neural networks [28, 35], support vector machines (SVM) [21] and SNoW [39] aim to find a decision surface between face and nonface patterns.[1] Discriminative methods require both positive (face) and negative (nonface) samples to find a decision boundary.

Nevertheless, studies in cognitive psychology have suggested that humans learn to recognize objects (e.g., faces) using positive examples without the need for negative examples [17]. Furthermore, while it is relatively easy to gather a representative set of face samples, it is extremely difficult to collect a representative set of nonface samples. The effectiveness of discriminative methods requires efforts in collecting nonface patterns. On the other hand, generative mixture methods such as a mixture of Gaussians and a mixture of factor analyzers rely on a joint probability distribution over examples, classification labels, and hidden variables (i.e., mixture weights). Although the joint distribution in this approach carries a number of advantages, e.g., in handling incomplete examples, the typical estimation criterion (maximum likelihood or its variants) is nevertheless suboptimal from the classification viewpoint. Furthermore, generative methods usually require data sets larger than those of discriminative methods since most of them involve estimating covariance matrices. Discriminative methods that focus directly on the parametric decision boundary, e.g., SVMs or Fisher's linear discriminant, typically yield better classification results, when they are applicable and properly utilized.

In this paper, we aim to investigate the advantages and disadvantages of generative and discriminative approaches to face detection. In the generative approach, we use only positive examples (i.e., face samples) and aim to estimate a probability distribution of face patterns. Furthermore, we use a mixture method to better model the distribution of face patterns. In the discriminative approach, we use Fisher's linear discriminant to find a decision boundary between face and nonface patterns. We then compare the performance of both methods on several benchmark data sets in order to investigate their pros and cons.

[1] Note that it is possible to incorporate generative methods in discriminative methods and vice versa.

The first detection method is an extension of factor analysis. Factor analysis (FA) is a statistical method for modeling the covariance structure of high-dimensional data using a small number of latent variables. FA is analogous to PCA in several aspects. However PCA, unlike FA, does not define a proper density model for the data since the cost of coding a data point is equal anywhere along the principal component subspace (i.e., the density is unnormalized along these directions). Further, PCA is not robust to independent noise in the features of the data since the principal components maximize the variances of the input data, thereby retaining unwanted variations. Synthetic and real examples in [3, 5, 8, 9] have shown that the projected samples from different classes in the PCA subspace can often be smeared. For the cases where the samples have certain structure, PCA is suboptimal from the classification standpoint. Hinton *et al*. have applied FA to digit recognition, and they compare the performance of PCA and FA models [14]. A mixture model of factor analyzers has recently been extended [11] and applied to face recognition [10]. Both studies show that FA performs better than PCA in digit and face recognition. Since pose, orientation, expression, and lighting condition affect the appearance of a human face, the distribution of faces in the image space can be better represented by a multimodal density model where each modality captures certain characteristics of certain face appearances. We present a probabilistic method that uses a mixture of factor analyzers (MFA) to detect faces with wide variations. The parameters in the mixture model are estimated using the EM algorithm.

The second method that we present uses Fisher's linear discriminant (FLD) to project samples from a high-dimensional image space to a lower-dimensional feature space. Recently, the Fisherface method [3] and others [36, 40] based on linear discriminant analysis have been shown to outperform the widely used Eigenface method [37] in face recognition on several data sets, including the Yale face database where face images are taken under varying lighting conditions. One possible explanation is that FLD provides a better projection than PCA for pattern classification since it aims to find the most discriminant projection direction. Consequently, the classification results in the projected subspace may be superior than other methods. (See [18] for a discussion about training set size). In the second proposed method, we decompose the training face and nonface samples into several subclasses using Kohonen's self-organizing map (SOM). From these relabeled samples, the within-class and between-class scatter matrices are computed, thereby generating the optimal projection based on FLD. For each subclass, we use a Gaussian to model its class-conditional density function where the parameters are estimated based on maximum likelihood [8, 9]. To detect faces, each input image is scanned with a rectangular window in which the class-dependent probability is computed. The maximum likelihood decision rule is used to determine whether a face is detected.

To capture the variations in face patterns, we use a set of 1681 face images from Olivetti [31], UMIST [12], Harvard [13], Yale [3], and FERET [23] databases. Our experimental results on the data sets used in [28, 35] (which consist of 145 images with 619 faces) show that our methods perform as well as the reported methods in the literature, yet with fewer false detections. To further test our methods, we collected a set of 80 images containing 252 faces. This data set is rather challenging since it contains profile views of faces, faces with a wide variety of expressions, and faces with heavy shadows. Our methods are able to detect most of these faces as well. Furthermore, our methods have fewer false detections than other methods.

The remainder of this paper is organized as follows. We review previous work on face detection in Section 2. In Section 3, we describe a mixture of factor analyzers and apply them

to face detection. We then present the second multimodal density model using Kohonen's self-organizing map algorithm for clustering and Fisher's linear discriminant for projection in Section 4. Comprehensive experiments on several benchmark data sets are detailed in Section 5. We also compare the results from our methods with other methods in the literature. Finally, we conclude this paper with comments and future work in Section 6.

2. PREVIOUS WORK

Numerous intensity-based methods have been proposed recently to detect human faces in a single image or a sequence of images. We discuss the most relevant works in this section and will present experiments and comparisons with these methods in Section 5. See [38] for a comprehensive survey on face detection methods.

Sung and Poggio have developed a clustering and distribution-based system for face detection [35]. Their system consists of two components, distribution-based models for face/nonface patterns and a multilayer perceptron classifier. Each face and nonface example is first normalized and processed to a 19×19 pixel pattern. Next, the training patterns are classified into six face and six nonface clusters using a modified k-means algorithm. Each face cluster is represented by a multidimensional Gaussian with a centroid location and a covariance matrix. Two distance metrics are computed between an input image pattern and the 12 prototype clusters. The first distance component is a normalized Mahalanobis distance between the test pattern and the cluster centroid, measured within a lower-dimensional subspace spanned by the cluster's 75 largest eigenvectors. The second distance component is the Euclidean distance between the test pattern and its projection onto the 75-dimensional subspace. This distance component accounts for pattern differences not captured by the first distance component. The last step is to use a multilayer perceptron network to classify face window patterns from nonface patterns using the 12 pairs of distances to each cluster. The classifier is trained using a standard backpropagation algorithm. Note that it is easy to get a representative sample of images that contain faces, but much more difficult to get a representative sample of those that do not. This problem is avoided by a bootstrap method that selectively adds image patterns to the training set as training progresses. They start with a small set of nonface training examples and train the classifier with this database. Then the face detector is applied to a set of random images, and all the nonface patterns that the current system wrongly classifies as faces are collected. These mislabeled nonface patterns are then added to the training database as new nonface examples.

Among all the face detection methods that use neural networks, the most significant work is probably by Rowley *et al*. [26–28]. There are two major components: multiple neural networks (to detect face patterns) and a decision-making module (to render the final decision from multiple detection results). The first component of this method is a neural network that receives a 20×20 pixel region of an image and outputs a score ranging from -1 to 1, indicating the possibility of a nonface or face pattern. To detect faces anywhere in an image, the neural network is applied at every location in the image. To detect faces larger than 20×20 pixels, the input image is repeatedly reduced in size by subsampling, and the filter is applied at each size. The second component of this method is to merge overlapping detection and arbitrate between the outputs of multiple networks. They use simple arbitration schemes such as logic operators (AND/OR) and voting to improve performance. The system by Rowley *et al*. [27] is less computationally expensive than Sung and Poggio's system,

and has a higher detection rate based on a test set of 24 images containing 144 faces. One limitation of both systems is that they can only detect upright, frontal faces. Recently Rowley *et al*. [29] have extended this method to detect rotated faces using a router network that processes each input window to determine the possible face orientation and then rotates the window to a canonical orientation; the rotated window is presented to the neural networks as described above. This system has a detection rate on upright faces lower than that of the upright detector. Nevertheless, the system is able to detect 76.9% of faces over two large test sets with a small number of false positives.

Kullback relative information is employed by Colmenarez and Huang to maximize the information-based discrimination between positive and negative examples of faces [6]. Images from the training set of each class (i.e., face and nonface class) are analyzed as observations of a random process, and are characterized by two probability functions. They used a family of discrete Markov processes to model the face and background patterns and to estimate the probability model. The learning process is converted into an optimization problem to select the Markov process that maximizes the information-based discrimination between the two classes. The likelihood ratio is computed using the trained probability model and used to detect the faces.

Schneiderman and Kanade [32] describe a naive Bayes classifier based on local appearance and position of the face pattern at multiple resolutions. They emphasize local appearance because some local patterns of an object are more unique than others; the intensity patterns around the eyes of a face are much more unique than the pattern found on the cheeks. At each scale, each face sample is decomposed into four rectangular subregions. These subregions are then projected to lower dimensional space using PCA and quantized into a finite set of patterns. The statistics of local appearance are then estimated independently from the samples (i.e., the frequency of each pattern) to encode the uniqueness of local appearance. The reason they adopt the naive Bayes assumption (i.e., no statistical dependency between the subregions) are twofold. First, it provides better estimation of the conditional density functions of these subregions. Second, a naive classifier provides a functional form of the posterior probability to capture the joint statistics of local appearance and position on the object. Under this formulation, their method decides that a face is present when the likelihood ratio is larger than the ratio of prior probabilities. The proposed Bayesian approach shows comparable performance to [28] and their method is able to detect some rotated and profile faces. Recently, Schneiderman and Kanade have extended this method to detect profile faces and cars [33].

Support vector machines have been applied to face detection by Osuna *et al*. [21]. SVMs can be considered as a new paradigm to train polynomial function, neural networks, or radial basis function (RBF). While most training techniques for classifiers such as neural networks and RBF are developed based on the principle of minimizing the training error (i.e., empirical risk), SVMs operate on another induction principle, called structural risk minimization, which minimizes an upper bound on the expected generalization error. In other words, a SVM classifier aims to find an optimal hyperplane such that the classification error of the unseen test patterns is minimized. Training a SVM is equivalent to solving a linearly constrained quadratic programming problem. However, the computation involved is both time and space intensive. In [21] Osuna *et al*. develop an efficient method for training a SVM for a large-scale problem and apply it to face detection. Based on two test sets of 10,000,000 test patterns of 19×19 pixels (Test sets A and B collected by Sung and Poggio [35]), their system has slightly lower error rates and runs approximately 30 times faster

than the system by Sung and Poggio [34]. Poggio *et al*. also apply SVMs to detect faces and pedestrians in the wavelet domain [20, 22].

3. MIXTURE OF FACTOR ANALYZERS

In the first method, we fit a mixture model of factor analyzers to the training samples using the EM algorithm [7] and obtain a distribution of face patterns. To detect faces, each input image is scanned with a rectangular window in which the probability of the current input being a face pattern is calculated. A face is detected if the probability is above a predefined threshold. We describe factor analysis and a mixture of factor analyzers in this section. More details of these models can be found in [2, 11].

3.1. Factor Analysis

Factor analysis is a statistical model in which the observed vector is partitioned into an unobserved systematic part and an unobserved error part. The systematic part is taken as a linear combination of a relatively small number of unobserved factor variables while the components of the error vector are considered as uncorrelated or independent. From another point of view, factor analysis gives a description of the interdependence of a set of variables in terms of the factors without regard to the observed variability. While PCA aims to extract a subspace in which the variance is maximized (thereby minimizing the reconstruction cost), some unwanted variations (due to lighting, facial expressions, viewing points, etc.) may be captured (see [3, 5, 8, 9] for examples). It has been observed that in face recognition the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to the chances in face identity [1]. Therefore, while the PCA projections are optimal in a correlation sense (or for reconstruction from a low-dimensional subspace), these eigenvectors or bases may be suboptimal from the classification viewpoint.

FA is a different way of analyzing the covariance matrix of the inputs that starts with a proper probabilistic model and correctly blends the reconstruction cost. Unlike PCA, the FA model allows different variances to be used for coding the residual errors on different dimensions. For face patterns, we expect that different regions of face images to have different levels of variability that can be modeled as pixel noise. The noise level within a forehead region, for instance, is likely to be smaller than that of an eye region. Therefore, FA is able to better model variation in pixel noise across images, and may provide better probabilistic model for faces. Another difference is that PCA is rotationally symmetric whereas FA is not. For FA, the particular dimensions used to describe the image are special in the sense that the noise corrupting them is taken to be mutually independent. See also [10, 11, 14] for further discussion on the differences between PCA and FA with illustrations on synthetic examples.

Formally, an observable data vector $\mathbf{x} \in \mathcal{R}^d$ is modeled using a vector of factors $\mathbf{z} \in \mathcal{R}^p$, where p is generally much smaller than d. The generative model is given by

$$\mathbf{x} = \mathbf{\Lambda}\mathbf{z} + \mathbf{u}, \tag{1}$$

where $\mathbf{\Lambda} \in \mathcal{R}^{d \times p}$ is known as the *factor loading matrix* and $\mathbf{u}$ variables account for

independent noise in each dimension of $\mathbf{x}$. The factors $\mathbf{z}$ model correlations between the elements of $\mathbf{x}$ and are assumed to be random with zero mean, i.e., $E[\mathbf{z}] = 0$. Let the covariance matrix of $\mathbf{z}$ be $\mathbf{\Phi}$ (i.e., $E[\mathbf{z}\mathbf{z}^{\mathrm{T}}] = \mathbf{\Phi}$) and we further assume factors are orthogonal, then $E[\mathbf{z}\mathbf{z}^{\mathrm{T}}] = \mathbf{\Phi} = \boldsymbol{I}$ (where $\boldsymbol{I}$ is an identity matrix). If $\mathbf{\Phi}$ is not diagonal, the factors are said to be oblique. In this paper, we assume the factors are orthogonal, and thus $\mathbf{z}$ is assumed to be $\mathcal{N}(0, \boldsymbol{I})$ distributed (zero-mean independent normals with unit variance). The d-dimensional random variable $\mathbf{u}$ is distributed $\mathcal{N}(0, \mathbf{\Psi})$ where $\mathbf{\Psi}$ is a diagonal matrix, due to the assumption that the observed variables $\mathbf{x}$ are independent given the factors $\mathbf{z}$. In other words,

$$p(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mathbf{\Lambda}\mathbf{z}, \Psi). \tag{2}$$

One crucial assumption is that the components of $\mathbf{u}$ are uncorrelated. In other words, the errors of observation $\mathbf{u}$ and the specific factors $\mathbf{z}$ are by definition uncorrelated. Another viewpoint of this assumption is that the factors are supposed to explain or account for as much of the variance of the observed data as possible. According to this model, $\mathbf{x}$ is therefore distributed with zero mean and covariance $\mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} + \mathbf{\Psi}$, since

$$\mathbf{\Sigma}_{\mathbf{x}} = E[(\mathbf{\Lambda}\mathbf{z} + \mathbf{u})(\mathbf{\Lambda}\mathbf{z} + \mathbf{u})^{\mathrm{T}}] = \mathbf{\Lambda}\boldsymbol{I}\mathbf{\Lambda}^{\mathrm{T}} + \mathbf{\Psi} + \mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} + \mathbf{\Psi}. \tag{3}$$

The goal of factor analysis is to find the $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ that best model the covariance structure of $\mathbf{x}$. The factor variables $\mathbf{z}$ model correlations between the elements of $\mathbf{x}$, while the $\mathbf{u}$ variables account for independent noise in each element $\mathbf{x}$. The p factors play the same role as the principal components in PCA; i.e., they are informative projections of the data. In other words, the columns of $\mathbf{\Lambda}$ can be considered as vectors that span a p-dimensional subspace ($p \leq d$). These p vectors can be considered as coordinate axes in the p-dimensional subspace, and $\mathbf{z}$ can be considered as coordinates of a point in this subspace, often called the *factor space*. Multiplying $\mathbf{\Lambda}$ on the right by a matrix corresponds to taking a projection in the factor space.

Note that $\mathbf{x}$, $\mathbf{z}$, and $\mathbf{u}$ are normally distributed random variables, and $\mathbf{x}$ is a linear combination of $\mathbf{z}$ and $\mathbf{u}$, so the conditional expectation can be computed easily. Given $\mathbf{\Lambda}$ and $\mathbf{\Psi}$, the joint distribution of $\mathbf{x}$ and $\mathbf{z}$ is

$$P\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}0\\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} + \mathbf{\Psi} & \mathbf{\Lambda}\\ \mathbf{\Lambda}^{\mathrm{T}} & \boldsymbol{I}\end{bmatrix}\right). \tag{4}$$

Note that the covariance matrix has been partitioned into matrices according to $\mathbf{x}$ and $\mathbf{z}$, and so these matrices do not have the same size.

Therefore, the expected value of the factors can be computed through the linear combinations

$$E[\mathbf{z} \mid \mathbf{x}] = \beta\mathbf{x} \tag{5}$$

$$\mathrm{Var}[\mathbf{z} \mid \mathbf{x}] = \boldsymbol{I} - \beta\mathbf{\Lambda}, \tag{6}$$

where $\beta = \mathbf{\Lambda}^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{x}}^{-1} = \mathbf{\Lambda}^{\mathrm{T}}(\mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} + \mathbf{\Psi})^{-1}$. The second moment of factors can be computed as

$$E[\mathbf{z}\mathbf{z}^{\mathrm{T}} \mid \mathbf{x}] = \mathrm{Var}(\mathbf{z} \mid \mathbf{x}) + E[\mathbf{z} \mid \mathbf{x}]E[\mathbf{z} \mid \mathbf{x}]^{\mathrm{T}} = \boldsymbol{I} - \beta\mathbf{\Lambda} + \beta\mathbf{x}\mathbf{x}^{\mathrm{T}}\beta^{\mathrm{T}}. \tag{7}$$

Rubin and Thayer presented an EM-based algorithm for maximum likelihood estimation of factor analysis [30]. The idea is to treat the unobservable $\mathbf{z}$ as missing data. Since $\mathbf{x}$ and $\mathbf{z}$ have a joint normal distribution, the sufficient statistics are the means and covariances. The E-step of the algorithm is to obtain the expectation of the covariances on the basis of trial values of the parameters. The M-step is to maximize the likelihood function on the basis of these covariances, which in turn provides updated values of the parameters. The steps alternate, and the procedure converges to the (local) maximum likelihood estimate [7].[2] Specifically, the expected values of factors in (5) and second moment in (7) form the basis of the EM algorithm. We summarize Rubin and Thayer's EM algorithm as follows:

- **E-step**: Given $\mathbf{\Lambda}$ and $\mathbf{\Psi}$, compute $E[\mathbf{z} \mid \mathbf{x}_i]$ and $E[\mathbf{z}\mathbf{z}^{\mathrm{T}} \mid \mathbf{x}_i]$ for each data point $\mathbf{x}_i$ using (5) and (7).
- **M-step**: Update $\mathbf{\Lambda}$ and $\mathbf{\Psi}$

$$\mathbf{\Lambda}^{new} = \left(\sum_{i=1}^{n} \mathbf{x}_i E[\mathbf{z} \mid \mathbf{x}_i]^{\mathrm{T}}\right)\left(\sum_{i=1}^{n} E[\mathbf{z}\mathbf{z}^{\mathrm{T}} \mid \mathbf{x}_i]\right)^{-1} \tag{8}$$

$$\mathbf{\Psi}^{new} = \frac{1}{n}\mathrm{diag}\left\{\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} - \mathbf{\Lambda}^{new} E[\mathbf{z} \mid \mathbf{x}_i]\mathbf{x}_i^{\mathrm{T}}\right\}, \tag{9}$$

where $\mathrm{diag}\{A\}$ means that only the diagonal elements are nonzero, and all the off-diagonal elements of a matrix are set to 0.

3.2. Mixture of Factor Analyzers

In this section, we consider modeling the distribution of face patterns by a mixture of m factor analyzers (indexed by f_i, $j = 1, \ldots, m$) where each factor analyzer has the same number of p factors, and each factor analyzer has a different mean $\boldsymbol{\mu}_j$. The generative model obeys the mixture distribution [11]

$$P(\mathbf{x}) = \sum_{j=1}^{m} \int P(\mathbf{x} \mid \mathbf{z}, f_j) P(\mathbf{z} \mid f_j) P(f_j)\, d\mathbf{z}, \tag{10}$$

where

$$P(\mathbf{z} \mid f_j) = P(\mathbf{z}) = \mathcal{N}(0, \boldsymbol{I}) \tag{11}$$

$$P(\mathbf{x} \mid \mathbf{z}, f_j) = \mathcal{N}(\boldsymbol{\mu}_j + \mathbf{\Lambda}_j\mathbf{z}, \mathbf{\Psi}). \tag{12}$$

The parameters of this mixture model are $\{(\boldsymbol{\mu}_j, \mathbf{\Lambda}_j)_{j=1}^{m}, \boldsymbol{\pi}, \mathbf{\Psi}\}$ where $\boldsymbol{\pi}$ is the vector of adaptable mixing proportions, $\boldsymbol{\pi}_j = P(f_j)$. The latent variables in this model are the factors

[2] Dempster *et al*. [7] proved that each iteration of EM increases the likelihood even if starting from a point where the likelihood is not convex. Also, if an instance of the algorithm converges, it converges to a (local) maximum of the likelihood.

$\mathbf{z}$ and the mixture indicator variable f_j, where $f_j = 1$ when the data point is generated by the jth factor analyzer.

Given a set of training images, the EM algorithm [11] is used to estimate the parameters, $\{(\boldsymbol{\mu}_j, \boldsymbol{\Lambda}_j)_{j=1}^m, \boldsymbol{\pi}, \boldsymbol{\Psi}\}$. For the E-step of the EM algorithm, we need to compute expectations of all the interactions of the hidden variables that appear in the log likelihood,

$$E[f_j \mathbf{z} \mid \mathbf{x}_i] = E[f_j \mid \mathbf{x}_i] E[\mathbf{z} \mid f_j, \mathbf{x}_i] \tag{13}$$

$$E[f_j \mathbf{z}\mathbf{z}^{\mathrm{T}} \mid \mathbf{x}_i] = E[f_j \mid \mathbf{x}_i] E[\mathbf{z}\mathbf{z}^{\mathrm{T}} \mid f_j, \mathbf{x}_i]. \tag{14}$$

Defining

$$h_{ij} = E[f_j \mid \mathbf{x}_i] \propto P(\mathbf{x}_i, f_j) = \boldsymbol{\pi}_j \mathcal{N}\left(\mathbf{x}_i - \boldsymbol{\mu}_j, \boldsymbol{\Lambda}_j \boldsymbol{\Lambda}_j^{\mathrm{T}} + \Psi\right) \tag{15}$$

and using (5) and (12), we obtain

$$E[f_j \mathbf{z} \mid \mathbf{x}_i] = h_{ij} \beta_j (\mathbf{x}_i - \boldsymbol{\mu}_j), \tag{16}$$

where

$$\beta_j \equiv \boldsymbol{\Lambda}_j^{\mathrm{T}} \left(\boldsymbol{\Lambda}_j \boldsymbol{\Lambda}_j^{\mathrm{T}} + \boldsymbol{\Psi}\right)^{-1}. \tag{17}$$

Similarly, using (7) and (14), we obtain

$$E[f_j \mathbf{z}\mathbf{z}^{\mathrm{T}} \mid \mathbf{x}_i] = h_{ij} \left(\boldsymbol{I} - \beta_j \boldsymbol{\Lambda}_j + \beta_j (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} \beta_j^{\mathrm{T}}\right). \tag{18}$$

The EM algorithm for mixture of factor analyzers can be stated as

- **E-step**: Compute $E[f_j \mid \mathbf{x}_i]$, $E[\mathbf{z} \mid f_j, \mathbf{x}_i]$, and $E[\mathbf{z}\mathbf{z}^{\mathrm{T}} \mid f_j, \mathbf{x}_i]$ for all data points i and mixture components j.
- **M-step**: Solve a set of linear equations for $\boldsymbol{\pi}_j$, $\boldsymbol{\Lambda}_j$, $\boldsymbol{\mu}_j$, and $\boldsymbol{\Psi}$.

The mixture of factor analyzers is essentially a reduced dimensionality mixture of Gaussians. Each factor analyzer fits a Gaussian to a portion of the data, weighted by the posterior probabilities, h_{ij}. Since the covariance matrix for each Gaussian is specified through the lower-dimensional factor-loading matrices, the model has $mpd + d$, rather than $md(d+1)/2$ parameters dedicated to modeling the covariance structure in high dimensions.

Note that since the EM algorithm involves matrix inversion in computing β and the condition number (the ratio of the largest singular value of $\mathbf{x}$ to the smallest) of input vector $\mathbf{x}$ is usually large (i.e., nearly singular), it is necessary to project input vectors $\mathbf{x}$ to a lower-dimensional space using PCA. Let $\mathbf{x}' = W_{\mathrm{PCA}}^{\mathrm{T}} \mathbf{x}$, we model the distribution of $\mathbf{x}'$ using the above equations, and we call the factors in the projected factor-loading matrix $W_{\mathrm{PCA}}^{\mathrm{T}} \boldsymbol{\Lambda}$ the *Factorfaces*. These Factorfaces give a description or explanation of the interdependence of faces in the training set without regard to the observed variability. Figure 1 shows 20 Factorfaces in one factor analyzer determined when estimating the mixture model on the training set. Note that each factor loosely resembles a face. In other words, each input face is modeled by a linear combination of the mixture of Factorfaces and noise. Figure 2 shows 20 Factorfaces in another factor analyzer. Note that these Factorfaces capture variations of faces in a 45° pose. This is the direct result of the mixture of factor analyzers since

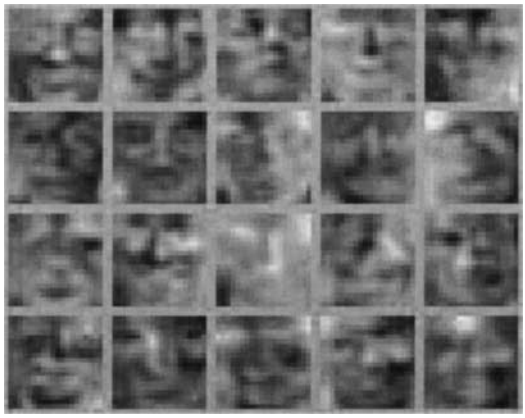

FIG. 1. Factorfaces: The Factorfaces (i.e., factors) in a factor analyzer explain observable face images based on the generative model. Each face is represented by a linear combination of a mixture of Factorfaces and noise. These Factorfaces capture variations in faces in frontal views. Note that the blurry results are due to the size of a face image being 20×20 pixels.

it concurrently performs clustering and, with each cluster, performs local dimensionality reduction.

3.3. Detecting Face Patterns

To detect faces, each input image is scanned with a rectangular window with d pixels, and the probability of the window's contents being a face pattern is estimated using (10). A face is detected if this probability is above a predefined threshold. In order to detect faces of different scales, each input image is repeatedly subsampled by a factor of 1.2, and the above mentioned detection process is applied. If the size of the face image in the training set is $v \times v$ pixels and the above iterative process is applied, we can detect faces up to $6.2v \times 6.2v$ pixels.

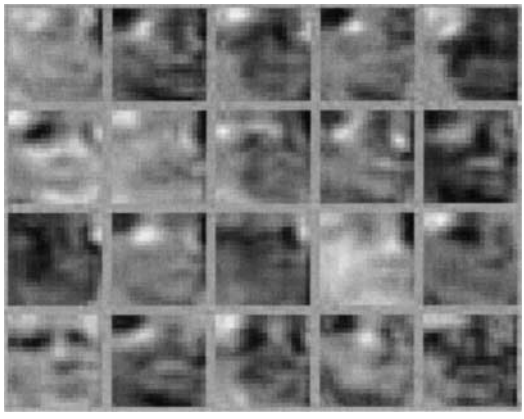

FIG. 2. Factorfaces: The Factorfaces in a factor analyzer explain observable face images based on the generative model. Each face is represented by a linear combination of a mixture of Factorfaces and noise. Note that these Factorfaces capture facial features of faces in 45° pose. Note that the blurry results are due to the size of a face image being 20×20 pixels.

4. MIXTURE OF LINEAR SPACES USING FISHER'S LINEAR DISCRIMINANT

In the second mixture model, we use Kohonen's self-organizing map [16] to first divide the face samples into c_1 face classes and then to divide the nonface samples into c_2 nonface classes. Next, Fisher projection is computed based on all c ($c = c_1 + c_2$) classes to maximize the ratio of the between-class scatter (variance) and the within-class scatter (variance). The now-labeled training set is projected from a high-dimensional image space to a lower-dimensional (i.e., $c - 1$ dimensional) feature space, and a Gaussian distribution is used to model the class-conditional density function for each class where the parameters are estimated using maximum likelihood. For detection, the conditional probability of each sample given each class is computed, and the maximum likelihood principle is used to decide in which class the sample belongs. In our experiments, we choose 25 face and 25 nonface classes because of the size of training set. In making this decision, we face the typical "curse of dimensionality" problem [4], which states that the amount of data required to construct a reliable estimate of the true solution increases exponentially with dimension. If the number of classes is too small, the clustering results may be poor. On the other hand, we may not have enough samples to estimate the class-conditional density function well if we choose a large number of classes.

4.1. Labeling Samples Using SOM

While Fisher's linear discriminant provides effective projections when two classes are unimodal or linearly separable, it may not be effective when classes can be characterized by a multimodal density function. Hence we decompose the training set (using Kohonen's SOM algorithm) into subclasses that can be characterized by a well-behaved density function, and then apply multidiscriminant analysis to the subclasses. In our experiments, we divide the face sample images into 25 classes. After training, the final weight vector for each node is the centroid of the class, i.e., the prototype vector, which corresponds to the prototype of each class. The same procedure is applied to nonface samples. Figure 3 shows the prototypical face of each class. It is clear that the sample face images with different poses and under different lighting conditions (intensity increases from the lower right corner to the upper left corner) have been classified into different classes. Note that the SOM algorithm also places the prototypes in the two-dimensional feature map, shown in Fig. 3, in accordance to their adjacency relationships in the image space. In other words, prototype vectors corresponding to nearby points on the feature map grid have nearby locations in the high-dimensional image space (e.g., nearby prototypes have similar intensity and pose). As mentioned at the beginning of this section, the number of classes has a direct relation with the number of samples. We partitioned the face (and nonface) samples with several configurations, e.g., 4×4, 5×6, and 7×7, and found that the SOM map with 5×5 grids performs best in our experiments.

Figure 4 shows the the resulting SOM map of nonface subclasses. The prototype vectors appear to be random dots that can be explained by the fact that the nonface training set consists of a wide variety of samples that do not seem to have common structural information.

4.2. Fisher's Linear Discriminant

While PCA is commonly used to project face patterns from a high-dimensional image space to a lower-dimensional feature space, a drawback is that it defines a subspace such

FIG. 3. Prototype of each face class: Each prototype vector corresponds to a centroid of a face subclass. The SOM algorithm also places the prototype vectors according to their adjacency relationships. In other words, the prototypes that are similar to each other are placed next to each other in the two-dimensional map.

that it has the greatest variance of the projected sample vectors among all the subspaces. However, such projection may not be effective for classification since large and unwanted variations may be retained. Consequently, the projected samples for each class may not be well clustered, and instead the samples may be smeared together [3, 10, 14]. Fisher's linear

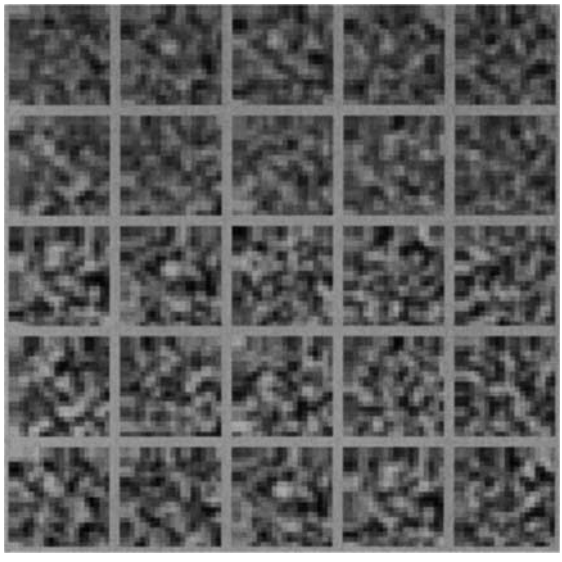

FIG. 4. Prototype of each nonface class: Each prototype vector corresponds to a centroid of a nonface subclass. These prototypes appear to be similar to random dots that do not contain any structural information. The results can be explained by the wide variety of nonface images.

discriminant is an example of a class-specific method that finds the optimal projection for classification. Rather than finding a projection that maximizes the projected variance, FLD determines a projection, $z = \boldsymbol{W}_{\mathrm{FLD}}^{\mathrm{T}} x$, that maximizes the ratio between the between-class scatter (variance) and the within-class scatter (variance). Consequently, classification is simplified in the projected space.

Consider a c-class problem with N samples; let the between-class scatter matrix be defined as

$$\boldsymbol{S}_{\mathrm{B}} = \sum_{i=1}^{c} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\mathrm{T}} \tag{19}$$

and the within-class scatter matrix be defined as

$$\boldsymbol{S}_{\mathrm{W}} - \sum_{i=1}^{c} \sum_{\mathbf{x}_k \in X_i} (\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^{\mathrm{T}} = \sum_{i=1}^{c} \boldsymbol{S}_{Wi}, \tag{20}$$

where $\boldsymbol{\mu}$ is the mean of all samples, $\boldsymbol{\mu}_i$ is the mean of class X_i, $\boldsymbol{S}_{\mathrm{W}i}$ is the covariance of class X_i, and N_i is the number of samples in class X_i.

The optimal projection $\boldsymbol{W}_{\mathrm{FLD}}$ is chosen as the matrix with orthonormal columns that maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected sampled, i.e.,

$$\boldsymbol{W}_{\mathrm{FLD}} = \arg\max_{\boldsymbol{W}} \frac{|\boldsymbol{W}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{B}} \boldsymbol{W}|}{|\boldsymbol{W}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{W}} \boldsymbol{W}|} = [\boldsymbol{w}_1 \quad \boldsymbol{w}_2 \quad \cdots \quad \boldsymbol{w}_m], \tag{21}$$

where $\{\boldsymbol{w}_i | i = 1, 2, \ldots, m\}$ is the set of generalized eigenvectors of $\boldsymbol{S}_{\mathrm{B}}$ and $\boldsymbol{S}_{\mathrm{W}}$, corresponding to the m largest generalized eigenvalues $\{\lambda_i \mid i = 1, 2, \ldots, m\}$. However, the rank of $\boldsymbol{S}_{\mathrm{B}}$ is $c - 1$ or less because it is the sum of c matrices of rank one or less. Thus, the upper bound on m is $c - 1$ [8, 9]. Similarly, the rank of $\boldsymbol{S}_{\mathrm{W}}$ is at most $N - c$. For a set of N sample images of n pixels where N is usually smaller than n, the within-scatter matrix $\boldsymbol{S}_{\mathrm{W}} \in \mathcal{R}^{n \times n}$ is always singular. This means that the projected within-scatter matrix can be zero if the projection matrix is not chosen properly. It is suggested in [3, 36, 40] that we can avoid this problem by first projecting the image set to a lower-dimensional space using PCA so that the resulting within-class scatter matrix $\boldsymbol{S}_{\mathrm{W}}$ is nonsingular before computing the optimal projection $\boldsymbol{W}_{\mathrm{FLD}}$. In other words, we first project the image set from N-dimensional space to $(N - c)$-dimensional space and then compute the optimal projection matrix using (21). Let $\mathbf{x}' = \boldsymbol{W}_{\mathrm{PCA}}^{\mathrm{T}} \mathbf{x}$ where $\boldsymbol{W}_{\mathrm{PCA}}$ is $n \times (N - C)$ matrix computed from

$$\boldsymbol{W}_{\mathrm{PCA}} = \arg\max_{\boldsymbol{W}} |\boldsymbol{W}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{T}} \boldsymbol{W}|, \tag{22}$$

where $\boldsymbol{S}_{\mathrm{T}} = \boldsymbol{S}_{\mathrm{W}} + \boldsymbol{S}_{\mathrm{B}}$ is the total scatter matrix. Next, we compute $\boldsymbol{W}_{\mathrm{FLD}}$ using $\mathbf{x}'$. Consequently, $\boldsymbol{W}_{\mathrm{FLD}}$ is an $(N - c) \times m$ matrix computed by

$$\boldsymbol{W}_{\mathrm{FLD}} = \arg\max_{\boldsymbol{W}} \frac{\left|\boldsymbol{W}^{\mathrm{T}} \boldsymbol{W}_{\mathrm{PCA}}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{B}} \boldsymbol{W}_{\mathrm{PCA}} \boldsymbol{W}\right|}{\left|\boldsymbol{W}^{\mathrm{T}} \boldsymbol{W}_{\mathrm{PCA}}^{\mathrm{T}} \boldsymbol{S}_{\mathrm{W}} \boldsymbol{W}_{\mathrm{PCA}} \boldsymbol{W}\right|}. \tag{23}$$

Using $\boldsymbol{W}_{\mathrm{FLD}}$, we cam project $\mathbf{x}$ from a n-dimensional space to $\mathbf{x}''$ in a $(c - 1)$-dimensional space spanned by nonzero eigenvectors.

4.3. Class-Conditional Density Function

Once $\boldsymbol{W}_{\text{FLD}}$ is computed, the now-labeled training set is projected to the $c-1$ dimensional feature space, i.e., $\mathbf{x}'' = \boldsymbol{W}_{\text{FLD}}^{\text{T}}\,\mathbf{x}$, and a Gaussian distribution is used to model each class-conditional density (CCD) function, i.e., $P(\mathbf{x}'' \mid X_i) = \mathcal{N}(\boldsymbol{\mu}_{X_i}, \boldsymbol{\Sigma}_{X_i})$, where $i = 1, \ldots, c$. The parameters, $\{\boldsymbol{\mu}_{X_i}, \Sigma_{X_i}\}$ of each CCD are the maximum likelihood estimates, i.e.,

$$\hat{\boldsymbol{\mu}}_{X_i} = \boldsymbol{W}_{\text{FLD}}^{\text{T}}\mu_i \tag{24}$$

and

$$\hat{\boldsymbol{\Sigma}}_{X_i} = \frac{1}{|X_i|}\boldsymbol{W}_{\text{FLD}}^{\text{T}}\boldsymbol{S}_{\text{W}i}\boldsymbol{W}_{\text{FLD}}, \tag{25}$$

where $|X_i|$ is the number of samples in class X_i.

4.4. Detecting Face Patterns

Each input image is scanned with a rectangular window to determine whether a face exists in the window. The decision rule for deciding whether an input window contains a face is based on maximum likelihood,

$$X^* = \arg\max_{i\in 1,\ldots,c} P(\mathbf{x}'' \mid X_i). \tag{26}$$

Given $\mathbf{x}''$, a face pattern is detected if X^* is a class label belonging to a face subclass. Otherwise, a nonface pattern is detected. To detect faces at different scales, each input image is repeatedly subsampled by a factor of 1.2 and scanned through for 10 iterations.

5. EXPERIMENTS

For training, we use a set of 1681 face images (collected from Olivetti [31], UMIST [12], Harvard [13], Yale [3], and FERET [23] databases) that has wide variations in pose, facial expression, and lighting conditions. In the first method, we collected nonface samples from 4000 images containing landscapes, trees, buildings, etc. We conducted experiments to find W_{PCA}, the appropriate number of factor analyzers, the appropriate number of factors in each factor analyzer, and a probability threshold. The $\boldsymbol{W}_{\text{PCA}}$ matrix projects face patterns from a 400-dimensional input image space to a 80-dimensional subspace before estimating the mixture of factor analyzers. We experimented and tuned the parameters (number of factor analyzers, number of factors, thresholds) during the training process. The best detection rate resulted from from a mixture of 36 factor analyzers in which each factor analyzer has 20 factors. In the second method, we start with 8422 nonface examples Although it is extremely difficult to collect a representative set of nonface examples, the bootstrap method similar to [35] is used to include more nonface examples during training. Each face sample is manually cropped and normalized such that it is aligned vertically and its size is 20×20 pixels. To make the detection method less sensitive to scale and rotation variation, 10 face examples are generated from each original sample. The images are produced by randomly rotating the images by up to 15° and randomly scaling them between 80 and 120%. Figure 5 shows some of the 16810 images generated by the this procedure.

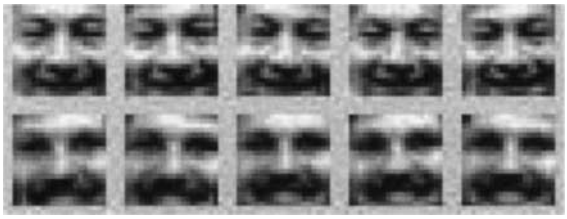

FIG. 5. Example face images that are generated by randomly rotating the original images by a small degree and scaling.

5.1. Empirical Results

We test both methods on the three sets of images collected by Rowley *et al*. [28], Sung and Poggio [35], and ourselves. Since faces are usually detected at multiple scales and positions, we use heuristics similar to [28] to merge all the overlapping windows so that each detected face is bounded by a single window in the final detection result. We first compute the centroid c of all the overlapping windows that have detected faces. The resulting c defines the centroid of a surrounding window that contains a detected face.

In our experiments, a detected face is a successful detection if the subimage contains both eyes and the mouth. Otherwise, it is a false detection. The detection rate is the ratio between the number of successful detections and the number of faces in the test set. Table 1 shows the detection rates of our methods and the reported results of several detection methods described in Section 2 on the test set in [28]. Experimental results on test set 1, which consists of 125 images (483 faces) excluding 5 images of hand-drawn faces, show that our methods have a detection performance comparable with that of other methods, yet with fewer false detections.

Table 2 shows the our experimental results on the test set of Sung and Poggio [35], which consists of 20 images excluding 3 images of line-drawn faces (136 faces). Consistent with the results from test set 1, both of our methods consistently perform well and have few false detections.

Test set 3 consists of 80 images (252 faces), collected from the World Wide Web, with different poses, expressions and faces with heavy shadows. This data set is available at http://vision.ai.uiuc.edu/mhyang/face-dataset.html. The detection rates are 86.7% and 88.2% for MFA- and FLD-based methods. The number of false detections are 45 and 40,

TABLE 1
Experimental Results on Images from Test Set 1
(125 Images with 483 Faces) in [28]

Method	Detection rate	False detection
Mixture of factor analyzers	92.3%	82
Fisher's linear discriminant	93.6%	74
Distribution-based [35]	N/A	N/A
Neural network [28]	92.5%	862
Naive Bayes classifier [32]	93.0%	88
Kullback relative information [6]	98.0%	12758
Support vector machine [21]	N/A	N/A

TABLE 2
Experimental Results on Images from Test Set 2
(20 Images with 136 Faces) in [35]

Method	Detection rate	False detection
Mixture of factor analyzers	89.4%	3
Fisher's linear discriminant	91.5%	1
Distribution-based [35]	81.9%	13
Neural network [28]	90.3%	42
Naive Bayes classifier [32]	91.2%	12
Kullback relative information [6]	N/A	N/A
Support vector machine [21]	74.2%	20

respectively. Both methods perform similarly well in detecting these faces, though the FLD-based method performs slightly better than the method using a mixture of factor analyzers. Figure 6 shows the results of our methods on some test images. For clarity, each face is shown with the window that is closest to the center of the face, though in some test images there exist multiple detections of the same face.

Note that there is a false detection in the upper left corner of the image in Fig. 6 since one window resembles a face. Also note that our methods can detect, up to a certain degree, some profile faces and faces with heavy shadows. However occluded, rotated faces or faces with sunglasses cannot be detected effectively by both methods due to lack of such examples in the training sets. None of the existing detection methods can effectively detect these types of faces, although a recent method is able to detect faces rotated in the image [29].

Figure 7 shows more results of our methods on test images. These results show that faces of different scales in cluttered backgrounds can be detected by our methods.

5.2. Discussion

Although Tables 1 and 2 show the performance of several methods on the same data set, it is still difficult to have a fair evaluation. There are at least three factors that complicate the assessment of these appearance-based methods. First, these learning methods usually use different training sets and different tuning parameters. The number of training examples has a direct effect on the classification performance. However this factor is often ignored when evaluating methods. The second factor is the training and execution time. Although the training time is usually ignored by most systems, it may be important for real-time applications that require training on different data sets. Finally, the number of scanned window locations in these methods vary a lot because they are designed to operate in different environments (i.e., to detect faces within a size range). For example, Colmenarez and Huang [6] argue that their method scans more windows than others and thus the number of false detections is higher than others. Furthermore, the criteria adopted in the reported detection rates is usually not clearly described in most systems. It is clear that a uniform criteria should be adopted in a fair assessment of these methods. Nevertheless, based on the reported results, our methods have a detection performance comparable with that of other methods, yet with fewer false detections.

Based on the motivation for the mixture of factor analyzers model, one might expect it to have a performance better than that of the mixture of Gaussians [35], and the experimental

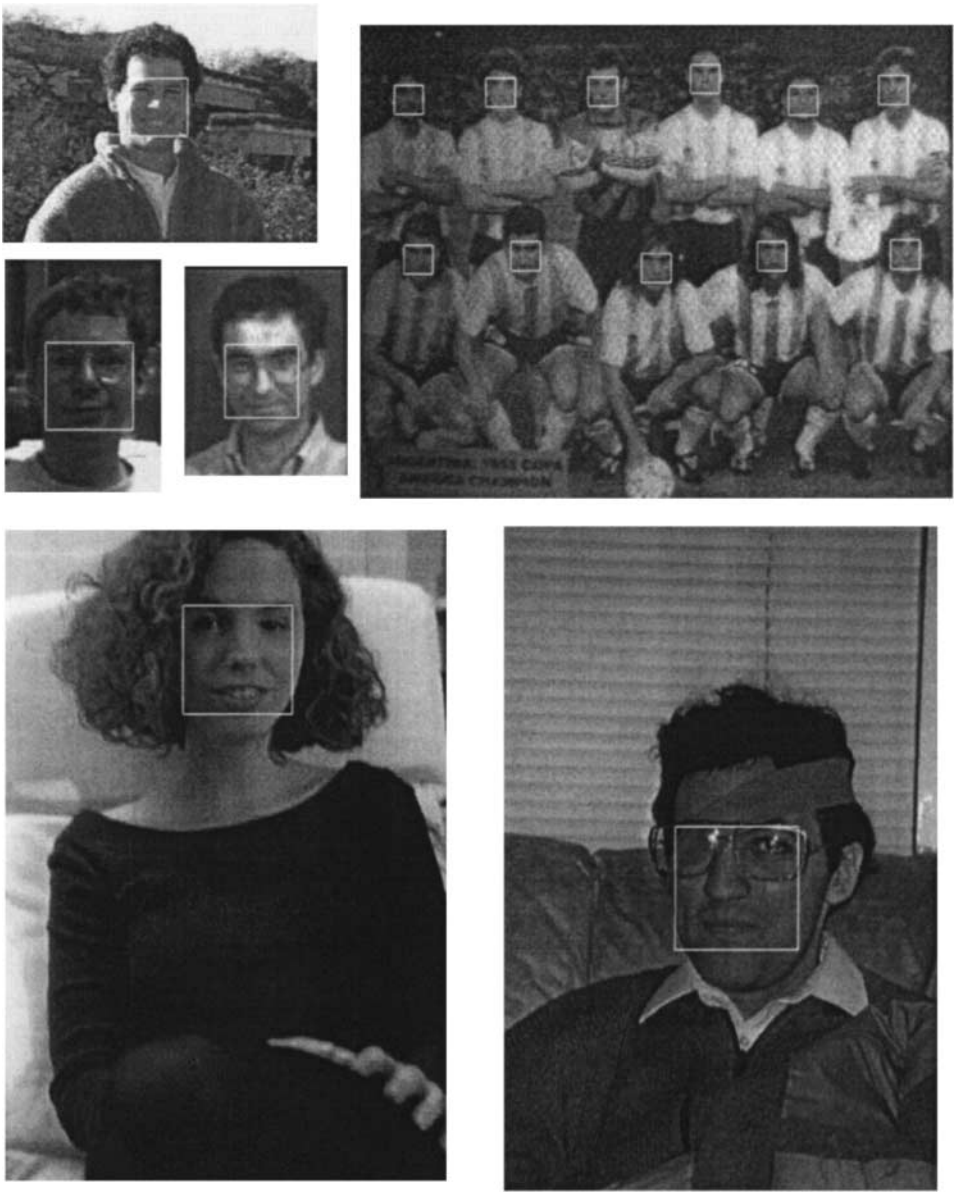

FIG. 6. Sample experimental results using the proposed methods. The bottom two test images are from data set 1 and the the top right image is from data set 2. The remaining 3 images on the top left corner are from data set 3. Both methods successfully detect all faces in these images where each detected face is shown with an enclosing window. However, the first method using a mixture of factor analyzers has one false positive as shown in the upper left image, whereas the second method does not have any false positive in that image. Also note that a profile face in the lower left image is detected but one partially occluded face is not.

results on two benchmark data sets show this to be true. These empirical results can be attributed to several reasons. First, FA explicitly models covariance structure, which allows FA to model variation in pixel noise across the face images, whereas PCA cannot. For modeling faces, FA is likely to be immune to the fact that different dimensions in the input vectors might have different intrinsic amounts of noise and look for the structure in

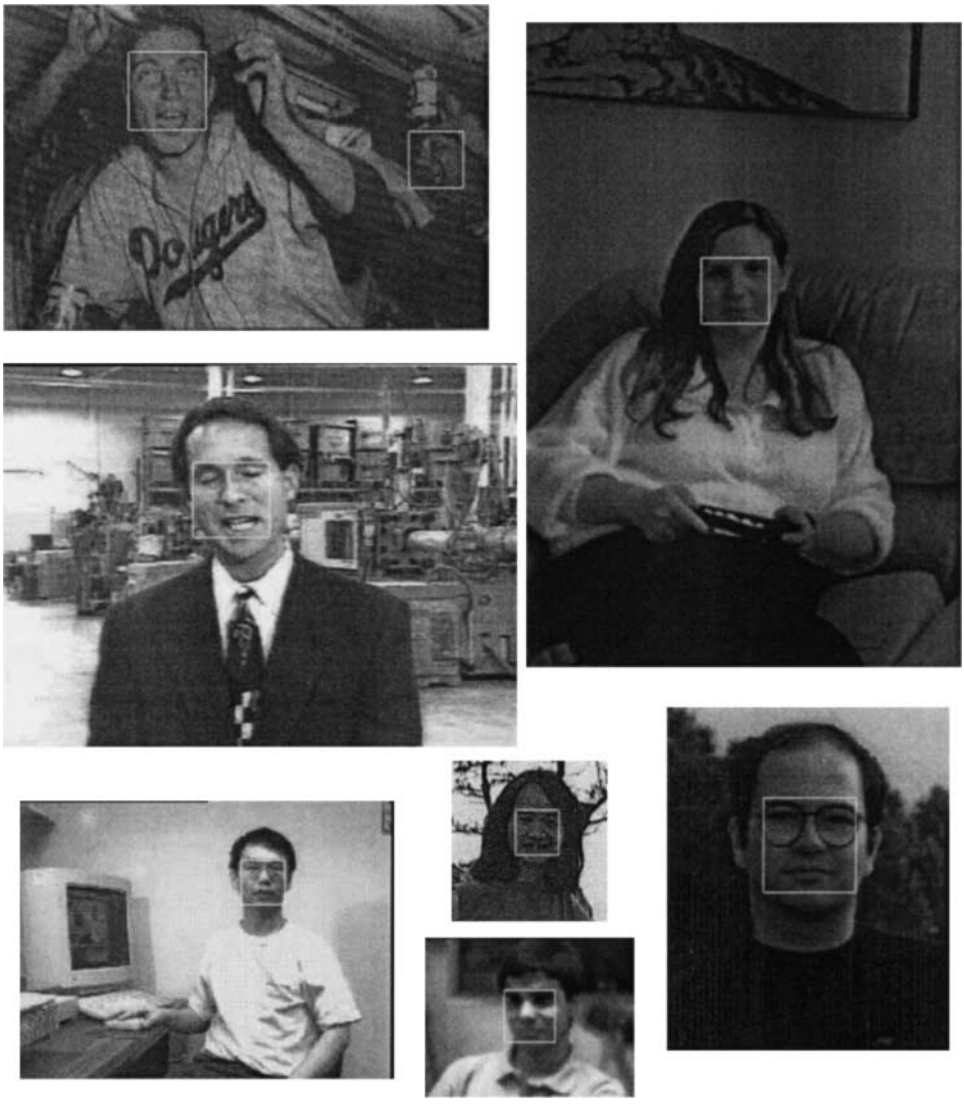

FIG. 7. Sample experimental results on images from data set 1. Both methods successfully detect all faces in these images without false positives, and since the localization is qualitatively similar, only the results for the first method are presented. A detected face is shown with an enclosing window. Note that faces against cluttered backgrounds are detected without false positives and that a profile face is detected in the upper left image.

faces. Second, PCA extracts a linear subspace that maximizes the variance retained in the subspace, but does not model the off-space noise nor the variance in the subspace, whereas in FA, the discrepancies (i.e., noise) between the model and the image are independent from one pixel to the next, given the factors.

In this study, we do not find much difference between two reported methods in the experiments with benchmark data sets, although the method using Fisher's linear discriminant performs slightly better than that using a mixture of factor analyzers. However, the actual computation time for learning in the second method using Fisher's linear discriminant is higher than that for the first method. One reason is that the second method involves

clustering and computation of several covariance matrices for positive and negative examples, whereas the first method uses only face images to estimate a multimodal density function for positive samples in which the clustering and dimensionality procedures are performed concurrently. In terms of running time, the method using a mixture of factor analyzers is slightly faster than the other method in our experiments. Finally, it would be interesting to merge the proposed methods with the "Anti-faces" techniques [15] to potentially improve the detection speed without compromising detection rates.

6. CONCLUDING REMARKS

We have described two methods using multimodal density models to detect human faces regardless of their pose, facial expression, and lighting conditions. The first method fits a mixture of factor analyzers to estimate the density function of face images, and the second method uses a self-organizing map to partition the training set into classes and Fisher's linear discriminant to find the optimal projection for classification. Experimental results on three sets of images demonstrate that both methods perform as well as the best algorithms in detecting upright frontal faces, yet with fewer false detections. Furthermore, our methods are able to detect faces with shadows and a range of poses.

The contributions of this paper can be summarized as follows. First, we introduce projection methods that have the potential to perform better than PCA in classification problems. The classification results in our experiments show that these methods perform well against several state-of-the-art systems. Second, we apply multimodal density models such that the each modality can better capture the variations of face patterns. Although some methods have used mixture models for face detection [19, 35], PCA is used for projection, and as mentioned earlier this does not maximize class separation as FLD does. On the other hand, it is not clear how well SVM performs in face detection since the study in [21] applied SVMs to a rather small test set of 136 faces. It will be of great interest to compare our methods with SVMs on a large test set since SVMs aim to find an optimal hyperplane that minimizes the generalization error under the theoretical upper bounds.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their comments and suggestions. The authors also thank Henry Rowley, for providing images. M.-H. Yang was supported by ONR grant N00014-00-1-009 and Ray Ozzie Fellowship. D. J. Kriegman was supported in part by NSF ITR CCR 00-86094 and NIH R01-EY 12691-01. N. Ahuja was supported in part by ONR grant N00014-00-1-009.

REFERENCES

1. Y. Adini, Y. Moses, and S. Ullman, Face recognition: The problem of compensating for changes in illumination direction, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 721–732.
2. T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, Wiley, New York, 1984.
3. P. Belhumeur, J. Hespanha, and D. Kriegman, Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 711–720.
4. R. Bellman, *Adaptive Control Process: A Guided Tour*, Princeton Univ. Press, Princeton, NJ, 1961.
5. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford Univ. Press, Oxford, 1995.
6. A. J. Colmenarez and T. S. Huang, Face detection with information-based maximum discrimination, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 782–787, 1997.

7. A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. R. Stat. Soc.* **39**, 1977, 1–38.
8. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
9. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley–Interscience, New York, 2001.
10. B. J. Frey, A. Colmenarez, and T. S. Huang, Mixtures of local subspaces for face recognition, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 32–37, 1998.
11. Z. Ghahramani and G. E. Hinton, The em algorithm for mixtures of factor analyzers, Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1996.
12. D. B. Graham and N. M. Allinson, Characterizing virtual eigensignatures for general purpose face recognition, in *Face Recognition: From Theory to Applications* (H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, Eds.), Vol. 163, NATO ASI Series F, Computer and Systems Sciences, pp. 446–456. Springer-Verlag, Berlin, 1998.
13. P. Hallinan, *A Deformable Model for Face Recognition Under Arbitrary Lighting Conditions*, Ph.D. thesis, Harvard University, Cambridge, MA, 1995.
14. G. E. Hinton, P. Dayan, and M. Revow, Modeling the manifolds of images of handwritten digits, *IEEE Trans. Neural Networks* **8**, 1997, 65–74.
15. D. Keren, M. Osadchy, and C. Gotsman, Anti-faces for detections, in *Proceedings of the Sixth European Conference on Computer Vision*, Vol. 1, pp. 134–148, 2000.
16. T. Kohonen, *Self-Organizing Map*, Springer-Verlag, Berlin, 1996.
17. D. D. Lee and H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature* **41**, 1999, 788–791.
18. A. Martinez and A. Kak, PCA versus LDA, *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 2001, 228–233.
19. B. Moghaddam and A. Pentland, Probabilistic visual learning for object recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 696–710.
20. M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, Pedestrian detection using wavelet templates, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 193–199, 1997.
21. E. Osuna, R. Freund, and F. Girosi, Training support vector machines: An application to face detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.
22. C. Papageorgiou, M. Oren, and T. Poggio, A general framework for object detection, in *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pp. 555–562, 1998.
23. P. J. Phillips, H. Moon, S. Rizvi, and P. Rauss, The feret evaluation, in *Face Recognition: From Theory to Applications* (H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, Eds.), Vol. 163, NATO ASI Series F, Computer and Systems Sciences, pp. 244–261. Springer-Verlag, Berlin, 1998.
24. R. J. Qian and T. S. Huang, Object detection using hierarchical MRF and MAP estimation, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 186–192, 1997.
25. A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri, Finding faces in photographs, in *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pp. 640–645, 1998.
26. H. Rowley, S. Baluja, and T. Kanade, Human face detection in visual scenes, in *Advances in Neural Information Processing Systems* (D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds.), Vol. 8, pp. 875–881, MIT Press, Cambridge, 1996.
27. H. Rowley, S. Baluja, and T. Kanade, Neural network-based face detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 203–208, 1996.
28. H. Rowley, S. Baluja, and T. Kanade, Neural network-based face detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 1998, 23–38.
29. H. Rowley, S. Baluja, and T. Kanade, Rotation invariant neural network-based face detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 38–44, 1998.
30. D. Rubin and D. Thayer, EM algorithms for ML factor analysis, *Psychometrika* **47**, 1982, 69–76.
31. F. S. Samaria, *Face Recognition Using Hidden Markov Models*, Ph.D. thesis, University of Cambridge, Cambridge, UK, 1994.

32. H. Schneiderman and T. Kanade, Probabilistic modeling of local appearance and spatial relationships for object recognition, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 45–51, 1998.
33. H. Schneiderman and T. Kanade, A statistical method for 3D object detection applied to faces and cars, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 746–751, 2000.
34. K.-K. Sung, *Learning and Example Selection for Object and Pattern Detection*, Ph.D. thesis, MIT AI Lab, Cambridge, MA, 1996.
35. K.-K. Sung and T. Poggio, Example-based learning for view-based human face detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 1998, 39–51.
36. D. Swets and J. Weng, Using discriminant eigenfeatures for image retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 1996, 831–836.
37. M. Turk and A. Pentland, Eigenfaces for recognition, *J. Cognitive Neurosci.* **3**, 1991, 71–86.
38. M.-H. Yang, D. Kriegman, and N. Ahuja, Detecting faces in images: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, submitted.
39. M.-H. Yang, D. Roth, and N. Ahuja, A SNoW-based face detector, in *Advances in Neural Information Processing Systems* (S. A. Solla, T. K. Leen, and K.-R. Müller, Eds.), Vol. 12, pp. 855–861, MIT Press, Cambridge, MA, 2000.
40. W. Zhao, R. Chellappa, and A. Krishnaswamy, Discriminant analysis of principal components for face recognition, in *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pp. 336–341, 1998.