Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 39 (2006) 260–276
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

# Face detection using discriminating feature analysis and Support Vector Machine

Peichung Shih*, Chengjun Liu

*Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA*

Received 21 September 2004; received in revised form 6 July 2005; accepted 6 July 2005

**Abstract**

This paper presents a novel face detection method by applying discriminating feature analysis (DFA) and support vector machine (SVM). The novelty of our DFA–SVM method comes from the integration of DFA, face class modeling, and SVM for face detection. First, DFA derives a discriminating feature vector by combining the input image, its 1-D Haar wavelet representation, and its amplitude projections. While the Haar wavelets produce an effective representation for object detection, the amplitude projections capture the vertical symmetric distributions and the horizontal characteristics of human face images. Second, face class modeling estimates the probability density function of the face class and defines a distribution-based measure for face and nonface classification. The distribution-based measure thus separates the input patterns into three classes: the face class (patterns close to the face class), the nonface class (patterns far away from the face class), and the undecided class (patterns neither close to nor far away from the face class). Finally, SVM together with the distribution-based measure classifies the patterns in the undecided class into either the face class or the nonface class. Experiments using images from the MIT–CMU test sets demonstrate the feasibility of our new face detection method. In particular, when using 92 images (containing 282 faces) from the MIT–CMU test sets, our DFA–SVM method achieves 98.2% correct face detection rate with two false detections.


*Keywords:* Discriminating feature analysis (DFA); Distribution-based measure; Face detection; Support Vector Machine (SVM); The DFA–SVM method

## 1. Introduction

Face detection methods generally learns statistical models of face and nonface images, and then apply two-class classification rules to discriminate between face and nonface patterns [1,2]. As a face must be located and extracted before it can be verified or identified, face detection is the first step towards building an automated face verification or identification system. Face verification mainly concerns authenticating a claimed identity posed by a person, while face identification focuses on recognizing the identity of a person from a database of known individuals [3,4]. An automated vision system that performs the functions of face detection, verification, and identification has great potential in a wide spectrum of applications, such as airport security and access control, building (embassy) surveillance and monitoring, human–computer intelligent interaction and perceptual interfaces, smart environments for home, office, and cars [2,1,5–7].

This paper presents a novel face detection method by applying discriminating feature analysis (DFA) and Support Vector Machine (SVM). The novelty of our DFA–SVM method comes from the integration of DFA, face class modeling, and SVM for face detection. Our DFA–SVM method works as follows: First, DFA derives a discriminating feature vector by combining the input image, its 1-D Haar wavelet representation, and its amplitude projections. While the Haar wavelets produce an effective representation for object detection, the amplitude projections capture the vertical symmetric distributions and the horizontal characteristics of human face images. Second, face class modeling

* Corresponding author.
*E-mail address:* ps9@oak.njit.edu (P. Shih).

0031-3203/$30.00 
doi:10.1016/j.patcog.2005.07.003

statistically estimates the probability density function (PDF) of the face class and defines a distribution-based measure for face and nonface classification. The face class is modeled as a multivariate normal distribution [8], and the distribution-based measure then separates the input patterns into three classes: the face class (patterns close to the face class), the nonface class (patterns far away from the face class), and the undecided class (patterns neither close to nor far away from the face class). Note that the distribution-based measure also derives nonface patterns for SVM training. Finally, the SVM together with the distribution-based measure classifies the patterns in the undecided class into either the face class or the nonface class. Experiments using images from the MIT–CMU test sets show the feasibility of our new face detection method. The DFA–SVM method is trained using 600 FERET facial images [9] and 3813 nonface images that lie close to the face class. When tested using 92 images (containing 282 faces) from the MIT–CMU test sets [10], our DFA–SVM method achieves 98.2% correct face detection rate with two false detections, a performance comparable to the state-of-the-art face detection methods, such as Schneiderman–Kanade's method [11].

## 2. Background

Earlier efforts of face detection research have been focused on correlation or template matching, matched filtering, subspace methods, deformable templates, etc. [12,13]. For comprehensive surveys of these early methods, see Refs. [14,6,7]. Recent face detection approaches, however, emphasize on statistical modeling and machine learning techniques [15,16]. Some representative methods are the probabilistic visual learning method [8], the example-based learning method [17], the neural network-based learning method [10,18], the probabilistic modeling method [11,19], the mixture of linear subspaces method [20], the machine learning approach using a boosted cascade of simple features [21], statistical learning theory and SVM-based methods [22–24], the Markov random field-based methods [25,26], the color-based face detection method [27], and the Bayesian discriminating feature (BDF) method [28].

Moghaddam and Pentland [8] applied unsupervised learning to estimate the density in a high-dimensional eigenspace and derived a maximum likelihood method for single face detection. Rather than using principal component analysis (PCA) for dimensionality reduction, they implemented the eigenspace decomposition as an integral part of estimating the conditional PDF in the original high-dimensional image space. Face detection is then carried out by computing multiscale saliency maps based on the maximum likelihood formulation. Sung and Poggio [17] presented an example-based learning method by means of modeling the distributions of face and nonface patterns. To cope with the variability of face images, they empirically chose six Gaussian clusters to model the distributions for face and nonface patterns, respectively. The density functions of the distributions are then fed to a multiple layer perceptron for face detection. Rowley et al. [10] developed a neural network-based upright, frontal face detection system, which applies a retinally connected neural network to examine small windows of an image and decide whether each window contains a face. The face detector, which was trained using a large number of face and nonface examples, contains a set of neural network-based filters and an arbitrator which merges detections from individual filters and eliminates overlapping detections. In order to detect faces at any degree of rotation in the image plane, the system was extended to incorporate a separate router network, which determines the orientation of the face pattern. The pattern is then derotated back to the upright position, which can be processed by the early developed system [18]. Schneiderman and Kanade [19] proposed a face detector based on the estimation of the posterior probability function, which captures the joint statistics of local appearance and position as well as the statistics of local appearance in the visual world. To detect side views of a face, profile images were added to the training set to incorporate such statistics [11]. Viola and Jones [21] presented a machine learning approach for face detection. The novelty of their approach comes from the integration of a new image representation (integral image), a learning algorithm (based on AdaBoost), and a method for combining classifiers (cascade). Hsu et al. [27] developed a face detection method in color images by detecting skin regions first, and then generating face candidates based on some constraints, such as the spatial arrangement. The face candidates are further verified by constructing eye, mouth, and boundary maps. Liu [28] recently developed a BDF method for multiple frontal face detection. The BDF method applies a DFA procedure for image representation, statistical modeling of face and nonface classes, and the Bayes classifier for face detection.

SVM is a particular implementation of statistical learning theory, which describes an approach known as structural risk minimization by minimizing the risk functional in terms of both the empirical risk and the confidence interval [29]. Osuna et al. [30] pioneered the research of face detection using SVM and demonstrated its generalization capability for face detection. Papageorgiou et al. [31] developed a wavelet-based SVM method for face and pedestrian detection. Romdhani et al. [32] proposed a method for speeding up a nonlinear SVM by evaluating a subset of support vectors. Recently, Bartlett et al. [33] combined an AdaBoost algorithm with an SVM for face detection in video, and Heisele et al. [34] presented a hierarchical face detection method using cascaded SVMs for face detection in a coarse-to-fine fashion.

## 3. Face detection using discriminating feature analysis and Support Vector Machine

The system architecture of our DFA–SVM face detection method is shown in Fig. 1. An input image is first processed by the DFA, which defines a feature vector by combining

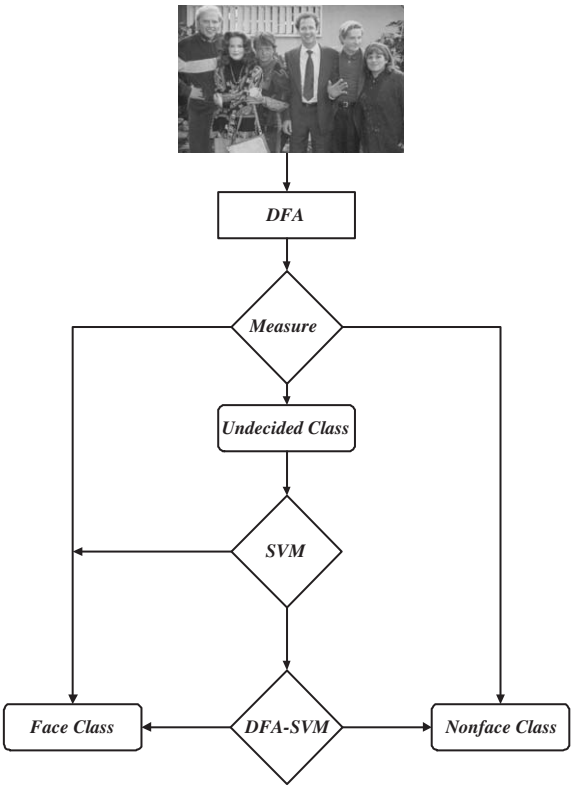



Fig. 1. System architecture of the DFA–SVM face detection method.

the input image, its 1-D Haar wavelet representation, and its amplitude projections. Based on the DFA feature vector, the face class is then modeled using a multivariate normal density and a distribution-based measure is defined for face and nonface classification. The distribution-based measure separates the patterns in the input image into three classes: the face class (patterns close to the face class), the nonface class (patterns far away from the face class), and the undecided class (patterns neither close to nor far away from the face class). Finally, a SVM detects faces in the undecided class and passes patterns to the DFA–SVM decision rule, which further classifies them into either the face class or the nonface class.

### 3.1. Discriminating feature analysis

Let $I(i, j) \in \mathbb{R}^{m\times n}$ represent an input image (e.g., training images for the face and the nonface classes, or subimages of test images), and $\mathbf{X} \in \mathbb{R}^{mn}$ be the vector formed by concatenating the rows (or columns) of $I(i, j)$. The 1-D Haar representation of $I(i, j)$ yields two images, $I_h(i, j) \in \mathbb{R}^{(m-1)\times n}$ and $I_v(i, j) \in \mathbb{R}^{m\times(n-1)}$, corresponding to the horizontal and vertical difference images, respectively. $I_h(i, j)$ and $I_v(i, j)$ then form two vectors, $\mathbf{X}_h \in \mathbb{R}^{(m-1)n}$ and $\mathbf{X}_v \in \mathbb{R}^{m(n-1)}$, by concatenating the rows (or columns). The amplitude projections of $I(i, j)$ along its rows and columns form the horizontal (row) and vertical (column) projections, $\mathbf{X}_r \in \mathbb{R}^m$ and $\mathbf{X}_c \in \mathbb{R}^n$, respectively.

The vectors $\mathbf{X}$, $\mathbf{X}_h$, $\mathbf{X}_v$, $\mathbf{X}_r$, and $\mathbf{X}_c$ are normalized to zero mean and unit variance, respectively. The normalized vectors are then concatenated to form a new feature vector $\mathbf{Y} \in \mathbb{R}^N$, where $N = 3mn$. Finally, the feature vector $\mathbf{Y}$ is normalized to zero mean and unit variance to form the discriminating feature vector for face detection. Fig. 2(a) shows the mean face (i.e., the average of the training face images), its 1-D Haar wavelet representation, and its amplitude projections. The first image is the mean face. The second and the third images are the vertical and the horizontal difference images of the mean face, respectively, which correspond to the 1-D Haar wavelet representation. The last two bar graphs draw the vertical (column) and horizontal (row) projections of the mean face, which correspond to the amplitude projections. Fig. 2(b) shows the mean nonface (i.e., the average of the training nonface images), its 1-D Haar wavelet representation, and its amplitude projections. Fig. 2(c) shows the mean support nonface (i.e., the average of the support nonface images, which are chosen from the training nonface images by SVM, see Section 3.3), its 1-D Haar wavelet representation, and its amplitude projections. Note that the images and projections in Fig. 2(b) and (c) resemble their counterparts in Fig. 2(a) due to the fact that the nonface samples lie close to the face class. Furthermore, Fig. 2(c) looks more like a face than Fig. 2(b) does because Fig. 2(c) is the mean of support nonfaces, which lie closer to the face class than other nonfaces do.

### 3.2. Face class modeling and distribution-based measure

Statistical modeling of the face class defines in essence the PDF, of the face class. A common assumption of the PDF is a multivariate normal distribution, which is especially reasonable if one models only the upright frontal faces that are properly aligned to one another [8,28]. Note that the training face images are all upright, frontal, and properly aligned, the density function of the face class is, therefore, modeled as a multivariate normal distribution

$$p(\mathbf{Y}) = \frac{1}{(2\pi)^{N/2}|\mathbf{\Sigma}|^{1/2}} \times \exp\left\{-\frac{1}{2}(\mathbf{Y}-\mathbf{M})^{\mathrm{t}}\mathbf{\Sigma}^{-1}(\mathbf{Y}-\mathbf{M})\right\}, \quad (1)$$

where $\mathbf{Y} \in \mathbb{R}^N$ is the discriminating feature vector, $\mathbf{M} \in \mathbb{R}^N$ and $\mathbf{\Sigma} \in \mathbb{R}^{N\times N}$ are the mean vector and the covariance matrix of the face class, $\omega_f$, respectively. Take the natural logarithm on both sides, we have

$$\ln[p(\mathbf{Y})] = -\tfrac{1}{2}\{(\mathbf{Y}-\mathbf{M})^{\mathrm{t}}\mathbf{\Sigma}^{-1}(\mathbf{Y}-\mathbf{M}) + N\ln(2\pi) + \ln|\mathbf{\Sigma}|\}. \quad (2)$$

The covariance matrix, $\mathbf{\Sigma}$, can be factorized into the following form using PCA [35]:

$$\mathbf{\Sigma} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{\mathrm{t}} \quad \text{with } \mathbf{\Phi}\mathbf{\Phi}^{\mathrm{t}} = \mathbf{\Phi}^{\mathrm{t}}\mathbf{\Phi} = \mathbf{I}_N,$$
$$\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_N\}, \quad (3)$$

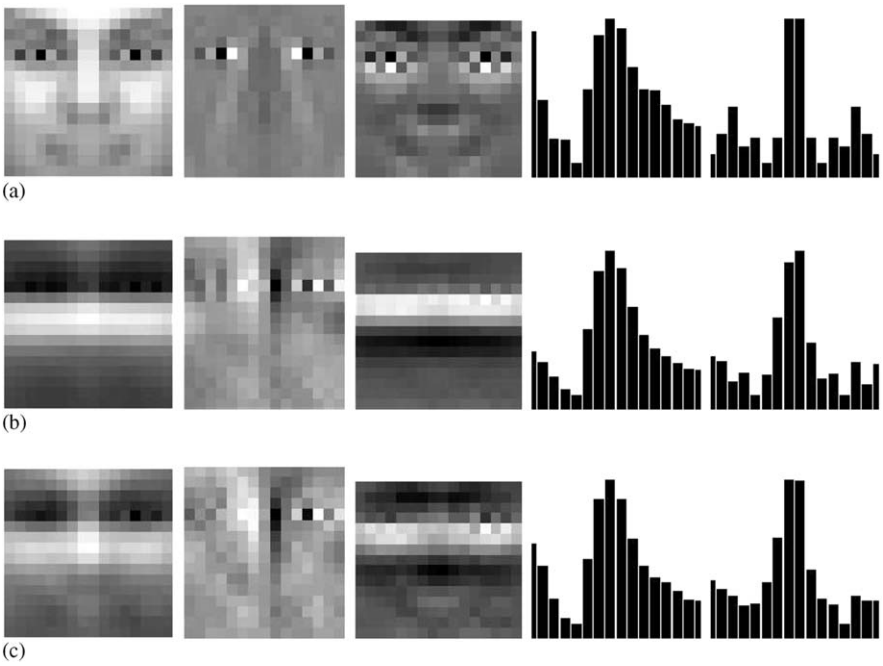



Fig. 2. Discriminating feature analysis of the mean face, the mean nonface, and the mean support nonface. (a) The first image is the mean face, the second and the third images are its 1-D Haar wavelet representation, and the last two bar graphs are its amplitude projections. (b) The mean nonface, its 1-D Haar wavelet representation, and its amplitude projections. (c) The mean support nonface, its 1-D Haar wavelet representation, and its amplitude projections. Note that the images and projections in (b) and (c) resemble their counterparts in (a) due to the fact that nonface samples lie close to the face class.

where $\mathbf{\Phi} \in \mathbb{R}^{N\times N}$ is an orthogonal eigenvector matrix, $\mathbf{\Lambda} \in \mathbb{R}^{N\times N}$ a diagonal eigenvalue matrix with diagonal elements (the eigenvalues) in decreasing order ($\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_N$), and $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is an identity matrix. An important property of PCA is its optimal signal reconstruction in the sense of minimum mean-square error when only a subset of principal components is used to represent the original signal [36]. The principal components are defined by the following vector, $\mathbf{Z} \in \mathbb{R}^N$:

$$\mathbf{Z} = \mathbf{\Phi}^{\mathrm{t}}(\mathbf{Y} - \mathbf{M}). \tag{4}$$

It then follows from Eqs. (2)–(4):

$$\ln[p(\mathbf{Y})] = -\tfrac{1}{2}\{\mathbf{Z}^{\mathrm{t}}\mathbf{\Lambda}^{-1}\mathbf{Z} + N\ln(2\pi) + \ln|\mathbf{\Lambda}|\}. \tag{5}$$

Note that the elements of $\mathbf{Z}$ are the principal components. Due to the optimal signal reconstruction property of PCA, we use only the first $M$ ($M \ll N$) principal components to estimate the PDF. We further adopt a model developed by Moghaddam and Pentland [8] to estimate the remaining $N-M$ eigenvalues, $\lambda_{M+1}, \lambda_{M+2}, \ldots, \lambda_N$, using the average of those values:

$$\rho = \frac{1}{N-M}\sum_{k=M+1}^{N}\lambda_k. \tag{6}$$

Note that from Eq. (4), we have $\|\mathbf{Z}\|^2 = \|\mathbf{Y} - \mathbf{M}\|^2$, where $\|\cdot\|$ denotes the norm operator. This observation shows that the PCA transformation, which is an orthonormal transformation, does not change norm. Now, it follows from Eqs. (5) and (6) that

$$\begin{aligned}\ln[p(\mathbf{Y})] &= -\frac{1}{2}\left\{\sum_{i=1}^{M}\frac{z_i^2}{\lambda_i} + \frac{\|\mathbf{Y}-\mathbf{M}\|^2 - \sum_{i=1}^{M} z_i^2}{\rho}\right.\\ &\quad\left. + \ln\left(\prod_{i=1}^{M}\lambda_i\right) + (N-M)\ln\rho + N\ln(2\pi)\right\},\end{aligned} \tag{7}$$

where $z_i$'s are the elements of $\mathbf{Z}$ defined by Eq. (4).

Note that the last three terms, $\ln(\prod_{i=1}^{M}\lambda_i)$, $(N-M)\ln\rho$, and $N\ln(2\pi)$, are constants for all patterns and discarded in our experimental implementations. Eq. (7) thus defines a distribution-based measure for face and nonface classification using the first $M$ principal components, the input image, and the mean face. The distribution-based measure not only classifies the input patterns into the face class (patterns close to the face class), the nonface class (patterns far away from the face class), and the undecided class (patterns neither close to nor far away from the face class), but also derives nonface images for SVM training (see Section 4.1). The SVM together with the distribution-based measure will finally classify the patterns in the undecided class into either the face class or the nonface class.

### 3.3. Support Vector Machine

SVM is a particular realization of statistical learning theory. It describes an approach known as structural risk

minimization, which minimizes the risk functional in terms of both the empirical risk and the confidence interval [29]. The main idea of SVM comes from (i) a nonlinear mapping of the input space to a high-dimensional feature space, and (ii) designing the optimal hyperplane in terms of the maximal margin between the patterns of the two classes in the feature space. SVM, which displays good generalization performance, has been applied extensively for pattern classification, regression, and density estimation.

Let $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_k, y_k)$, $\mathbf{x}_i \in \mathbb{R}^N$, and $y_i \in \{+1, -1\}$ be $k$ training samples in the input space, where $y_i$ indicates the class membership of $\mathbf{x}_i$. Let $\varphi$ be a nonlinear mapping between the input space and the feature space, $\varphi : \mathbb{R}^N \to \mathcal{F}$, i.e., $\mathbf{x} \to \varphi(\mathbf{x})$. The optimal hyperplane in the feature space is defined as follow:

$$w_0 \cdot \varphi(\mathbf{x}) + b_0 = 0. \tag{8}$$

It can be proven [29] that the weight vector $w_0$ is a linear combination of the support vectors, which are the vectors $\mathbf{x}_i$ that satisfy $y_i(w_0 \cdot \varphi(\mathbf{x}_i) + b_0) = 1$:

$$w_0 = \sum_{\text{support vectors}} y_i \alpha_i \varphi(\mathbf{x}_i), \tag{9}$$

where $\alpha_i$'s are determined by maximizing the following functional:

$$L(\alpha) = \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{k} \alpha_i \alpha_j y_i y_j \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j) \tag{10}$$

subject to the following constraints:

$$\sum_{i=1}^{k} \alpha_i y_i = 0, \quad \alpha_i \geqslant 0, \quad i = 1, 2, \ldots, k. \tag{11}$$

From Eqs. (8) and (9), we can derive the linear decision function in the feature space

$$f(\mathbf{x}) = sign\left( \sum_{\text{support vectors}} y_i \alpha_i \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}) + b_0 \right). \tag{12}$$

Note that the decision function (see Eq. (12)) is defined by the dot products in the high-dimensional feature space, where computation might be prohibitively expensive. SVM, however, manages to compute the dot products by means of a kernel function [29]

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j). \tag{13}$$

Three classes of kernel functions widely used in kernel classifiers, neural networks, and SVMs are polynomial kernels, Gaussian kernels, and sigmoid kernels [29]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d, \tag{14}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left( -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right), \tag{15}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa(\mathbf{x}_i \cdot \mathbf{x}_j) + \vartheta), \tag{16}$$

where $d \in \mathbb{N}$, $\sigma > 0$, $\kappa > 0$, and $\vartheta < 0$.

### 3.4. Face detection using distribution-based measure and SVM

The first decision rule applies the distribution-based measure (Eq. (7)) to detect faces that are very close to the face class and exclude patterns that are very far away from the face class. This decision rule separates the patterns in the input image into three classes: the face class ($\omega_f$), the non-face class ($\omega_n$), and the undecided class ($\omega_u$):

$$\mathbf{Y} \in \begin{cases} \omega_f & \text{if } \ln[p(\mathbf{Y})] \geqslant \tau_f, \\ \omega_u & \text{if } \tau_n < \ln[p(\mathbf{Y})] < \tau_f, \\ \omega_n & \text{otherwise,} \end{cases} \tag{17}$$

where $\mathbf{Y}$ is the discriminating feature vector derived from an input pattern, $\ln[p(\mathbf{Y})]$ is the distribution-based measure defined by Eq. (7), and $\tau_f$ and $\tau_n$ are thresholds.

The second decision rule, the SVM decision rule, then applies the SVM classifier to detect faces in the $\omega_u$ class:

$$\mathbf{Y} \in \begin{cases} \omega_f & \text{if } f(\mathbf{Y}) > 0, \\ \omega_n & \text{otherwise,} \end{cases} \tag{18}$$

where $f(\mathbf{Y})$ is the decision function of the SVM classifier defined by Eq. (12). Our experiments, however, show that the SVM classifier alone cannot detect all the faces in the $\omega_u$ class, and some face patterns are misclassified to the $\omega_n$ class.

To improve the face detection performance of our method, we design the third decision rule, the DFA–SVM decision rule, which further checks the patterns assigned to the $\omega_n$ class from the $\omega_u$ class:

$$\mathbf{Y} \in \begin{cases} \omega_f & \text{if } g(\mathbf{Y}) > \tau_s \text{ and } \ln[p(\mathbf{Y})] + cg(\mathbf{Y}) > \tau_t, \\ \omega_n & \text{otherwise,} \end{cases} \tag{19}$$

where $c$ is a positive constant, $\tau_s$ and $\tau_t$ are thresholds, and $g(\mathbf{Y})$ is the decision value of the SVM classifier without the sign function (see Eq. (12)):

$$g(\mathbf{Y}) = \sum_{\text{support vectors}} y_i \alpha_i \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{Y}) + b_0. \tag{20}$$

Note that the functionality of the first term, $g(\mathbf{Y}) > \tau_s$, is to select candidate face patterns misclassified to the $\omega_n$ class by the SVM classifier (see Eq. (18)); the second term, $\ln[p(\mathbf{Y})] + cg(\mathbf{Y}) > \tau_t$, then further determines the true faces from the candidate patterns by linearly combining the decision values from the distribution-based measure and the SVM classifier.

The three decision rules detailed above actually apply a coarse-to-fine classification strategy in the sense that they

are cascaded in an increasing order of detection accuracy and decreasing order of computational efficiency. Regarding detection accuracy, the third decision rule detects faces more accurately than the first two rules do because it combines the classification power of the distribution-based measure and SVM. As to computational efficiency, the first decision rule runs faster than the second and the third one due to the fact that the term $\ln[p(\mathbf{Y})]$ is evaluated by estimating the first $M$ principal components of the input vector $\mathbf{Y}$, while the term $g(\mathbf{Y})$ is obtained directly in the original input space, $\mathbb{R}^N$. Note that $M$ is much smaller than $N$ (see Section 3.2).

## 4. Experiments

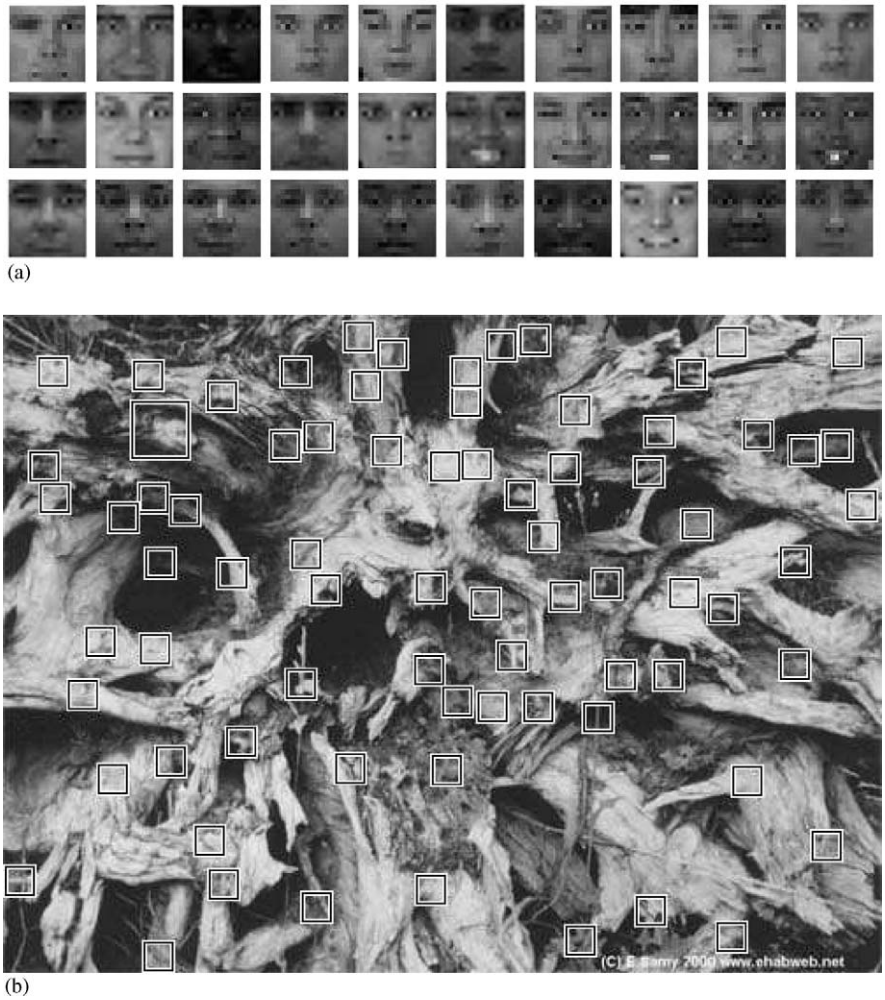
This section details statistical learning, learning the thresholds, face detection performance, and computational complexity of the DFA–SVM method. The training data for the DFA–SVM method comes from the FERET database [9] Batch 15, which contains 600 frontal face images. The testing data comes from the MIT–CMU test sets [10], which include images from diverse sources. Experimental results show that our DFA–SVM method, which is trained on a simple image set yet works on much more complex images, displays robust generalization performance.

### 4.1. Statistical learning of the DFA–SVM method

The statistical learning of the DFA–SVM method includes three stages: face class modeling, nonface image generation, and SVM. First, the DFA–SVM method models the face class, $\omega_f$, as a multivariate normal distribution and defines a distribution-based measure, $\ln[p(\mathbf{Y})]$, using 600 frontal face images from the FERET database Batch 15 [9]. Note that we also include the mirror images of the training data; hence the total training images for face class modeling is 1200.

Second, nonface training images are derived by choosing subimages from 14 natural scene images that do not contain any face at all. The subimages that lie close to (in the sense of the distribution-based measure, $\ln[p(\mathbf{Y})]$) the face class are chosen as nonface training images:

$$\ln[p(\mathbf{Y})] > \tau_n, \tag{21}$$




(a)

(b)

Fig. 3. Some examples of the face and the nonface training images: (a) examples of the face training images; (b) examples of the nonface training images.

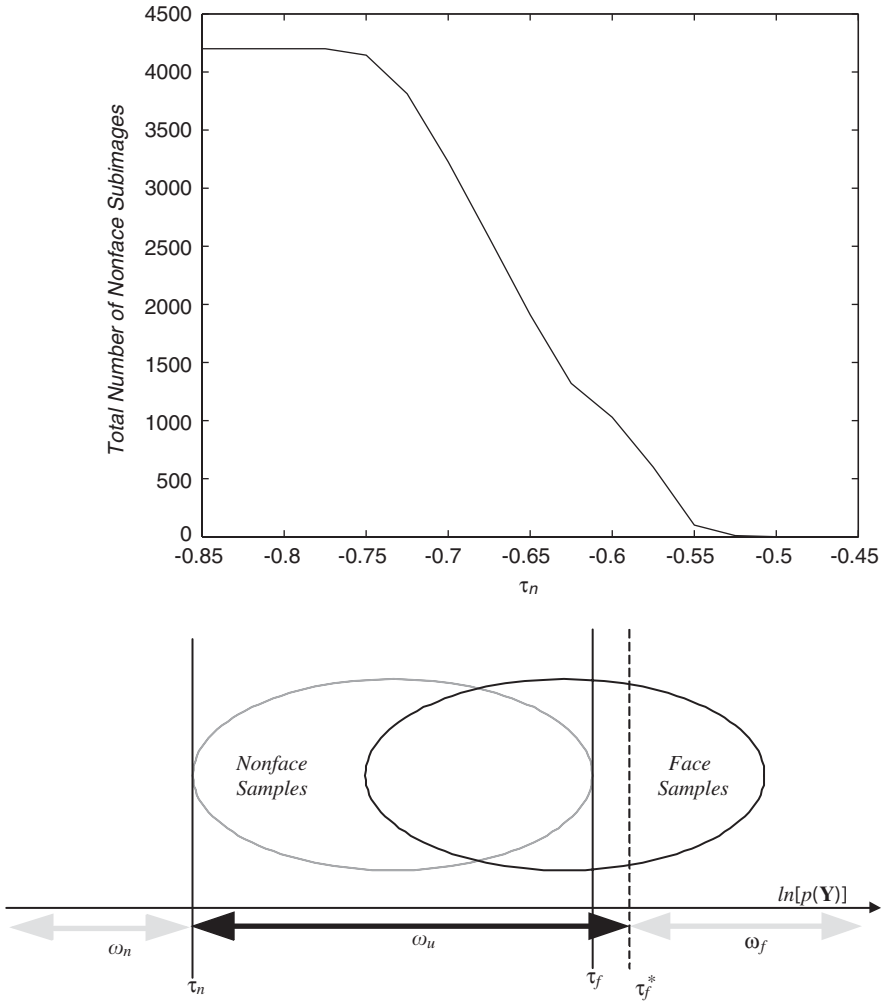



Fig. 4. The learning of thresholds $\tau_n$ and $\tau_f$. (a) The relationship between the total number of nonface subimages and the threshold $\tau_n$. The horizontal axis indicates the value of $\tau_n$ and the vertical axis is the total number of nonface subimages derived from all the 14 natural scene images used in our experiments. (b) The threshold, $\tau_f$, is set to be the largest distribution-based measure of all these nonface images. Note that in our experiments, the threshold $\tau_f$ is increased to $\tau_f^*$ to reduce the number of false detections.

where $\ln[p(\mathbf{Y})]$ is the distribution-based measure (see Eq. (7)) and $\tau_n$ is the threshold (see Eq. (17)). Note that the number of principal components, $M$, defined in Eq. (7) is fixed at 10 throughout our experiments.

Finally, we apply the 1200 face images and 3813 nonface images to train a SVM with a polynomial kernel of degree 2 for face detection. Note that both the face and the nonface images are normalized to a spatial resolution of $16 \times 16$. Fig. 3(a) shows some examples of the face training images that are normalized to $16 \times 16$, and Fig. 3(b) shows some examples of the nonface training images derived from a natural scene image. Note that the nonface images in Fig. 3(b) display different sizes, which correspond to different scales of the original natural scene image when the nonface images are derived. The spatial resolution of all the nonface images, however, is the same, $16 \times 16$.

### 4.2. Learning the thresholds

Thresholds play important roles in our DFA–SVM face detection method as a change of a threshold may affect the system performance significantly. This section describes a general procedure to fine-tune the four thresholds, $\tau_n$, $\tau_f$, $\tau_s$, and $\tau_t$, defined in Section 3.4.

#### 4.2.1. Thresholds $\tau_f$ and $\tau_n$

The learning of thresholds starts with $\tau_n$ as it directly determines the number of nonface training images derived from the 14 natural scene images (see Eq. (21)). Fig. 4(a) plots the relationship between the total number of nonface images and the threshold $\tau_n$. The horizontal axis indicates the value of $\tau_n$, and the vertical axis is the total number of nonface images derived from all the 14 natural scene images. Note that in order to prevent one scene image from unduly

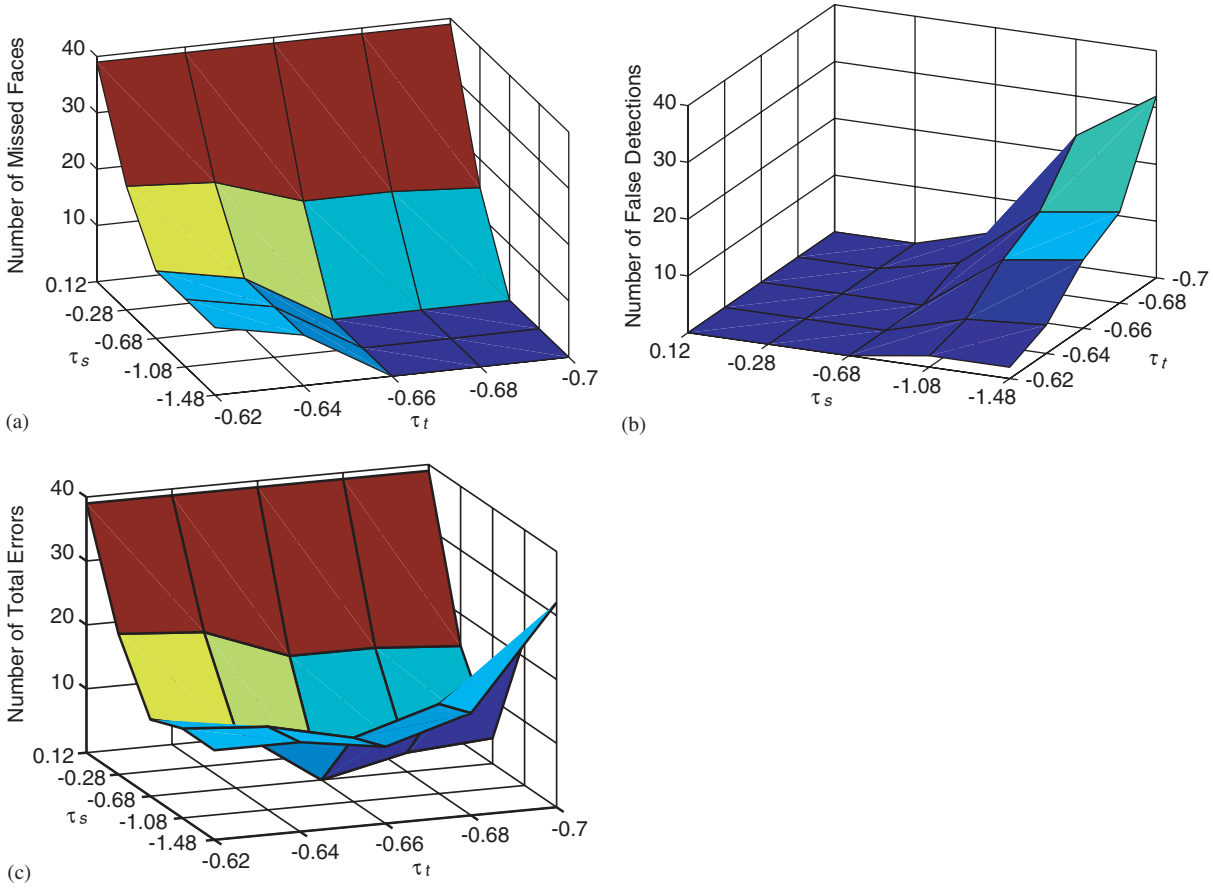



Fig. 5. The learning of thresholds $\tau_s$ and $\tau_t$: (a) the number of missed faces increases when either $\tau_s$ or $\tau_t$ increases; (b) the number of false detections decreases when either $\tau_s$ or $\tau_t$ increases; (c) the number of total errors (missed faces + false detections) versus the thresholds $\tau_s$ and $\tau_t$. The smallest total error occurs at $\{-0.68, -0.66\}$.

dominating the others, we cap the number of nonface images for each scene at 300. As a result, the maximum number of nonface images derived from all the scene images should be 4200, which is shown in Fig. 4(a) by the flat curve segment. Our choice of $\tau_n$ is thus close to the threshold value that leads to this flat curve segment. In particular, the chosen threshold $\tau_n$ generates 3813 nonface images from the 14 natural scene images, and these generated images form the nonface training set for the DFA–SVM face detection method.

After generating the nonface images, a reasonable choice of the threshold, $\tau_f$, is the largest distribution-based measure of all these nonface images (see Fig. 4(b)). However, Eq. (17) is applied for coarse detection, whose purpose is to reliably classify face and nonface patterns while leaving difficult patterns in the undecided class. We therefore increase the threshold from $\tau_f$ to $\tau_f^*$ (see Fig. 4(b)) to reduce the number of false detections.

#### 4.2.2. Thresholds $\tau_s$ and $\tau_t$

The thresholds, $\tau_s$ and $\tau_t$, in Eq. (19) are used for the fine detection, whose functionality is to classify the undecided patterns into either the face class or the nonface class. The values of these two thresholds are determined through a numerical analysis procedure. Let $\Gamma_s$ be a set of values of $\tau_s$ and $\Gamma_t$ be a set of values of $\tau_t$, we evaluated the face detection performance of each combination of $\Gamma_s \times \Gamma_t$ using a subset (24 images containing 54 faces) of the MIT–CMU test sets. Note that the parameter, $c$, defined in Eq. (19) is fixed at 0.05 throughout our experiments.

Figs. 5(a), (b), and (c) show the number of missed faces versus the thresholds ($\tau_s$ and $\tau_t$), the number of false detections versus the thresholds, and the number of total errors (missed faces + false detections) versus the thresholds, respectively. Fig. 5(a) shows that the number of missed face increases when either $\tau_s$ or $\tau_t$ increases. Fig. 5(b) shows, however, that the number of false detections decreases when either $\tau_s$ or $\tau_t$ increases. Finally, Fig. 5(c) shows that the smallest total error occurs at $\{-0.68, -0.66\}$. We therefore set $\tau_s$ and $\tau_t$ to be $-0.68$ and $-0.66$, respectively.

### 4.3. Face detection performance of the DFA–SVM method

The data used to test our DFA–SVM method for face detection comes from the MIT–CMU test sets [10].

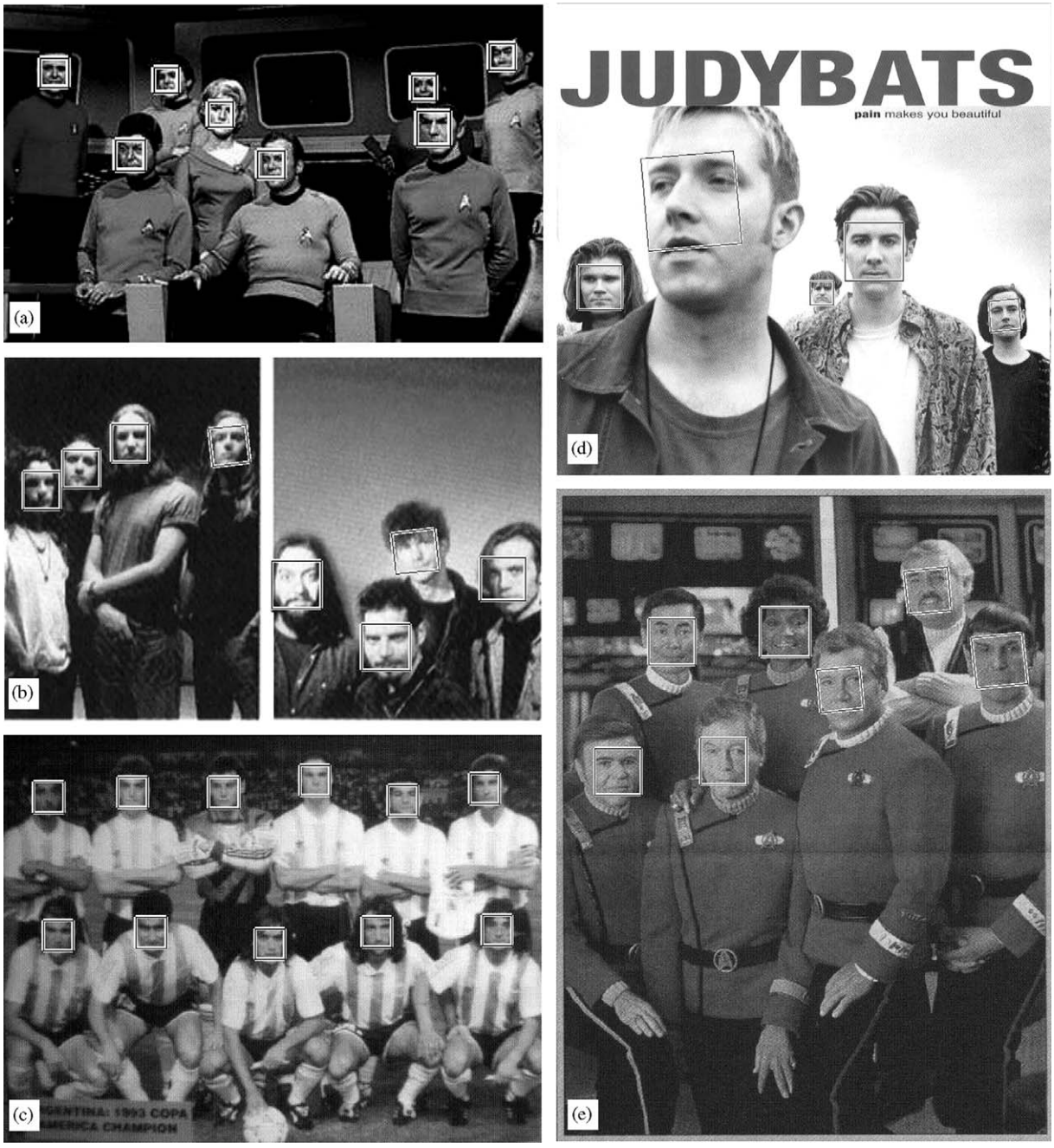



Fig. 6. Examples of detecting multiple faces using the DFA–SVM method.

Specifically, the test set includes 92 images containing 282 faces.[1] As our method addresses detection of frontal and real human faces, those MIT–CMU test images containing large pose-angled faces, line-drawn faces, poker faces, masked faces, or cartoon faces are not included in our experiments. Note that the test images used in our

[1] The 92 images are listed as follows: *aerosmith-double.gif*, *albert.gif*, *Argentina.gif*, *audrey1.gif*, *audrey2.gif*, *audrybt1.gif*, *baseball.gif*, *bksomels.gif*, *blues-double.gif*, *brian.gif*, *bttf301.gif*, *bwolen.gif*, *cfb.gif*,*churchill-downs.gif*, *class57.gif*, *cluttered-tahoe.gif*, *cnn1085.gif*, *cnn1160.gif*, *cnn1260.gif*, *cnn1630.gif*, *cnn1714.gif*, *cnn2020.gif*, *cnn2221.gif*,*cnn2600.gif*, *cpd.gif*, *crimson.gif*, *ds9.gif*, *ew-courtney-david.gif*, *ew-friends.gif*,*fleetwood-mac.gif*, *frisbee.gif*, *Germany.gif*, *giant-panda.gif*, *gigi.gif*, *gpripe.gif*, *harvard.gif*, *hendrix2.gif*, *henry.gif*, *jackson.gif*, *john.coltrane.gif*, *judybats.gif*, *kaari-stef.gif*, *kaari1.gif*, *kaari2.gif*,*karen-and-rob.gif*, *knex0.gif*, *knex37.gif*, *kymberly.gif*, *lacrosse.gif*,*larroquette.gif*, *madaboutyou.gif*, *married.gif*, *me.gif*,*mom-baby.gif*,*mona-lisa.gif*, *music-groups-double.gif*, *natalie1.gif*, *nens.gif*, *newsradio.gif*, *oksana1.gif*, *original1.gif*, *original2.gif*, *pittsburgh-park.gif*, *police.gif*, *sarah4.gif*, *sarah_live_2.gif*, *seinfeld.gif*, *shumeet.gif*, *soccer.gif*, *speed.gif*, *tahoe-and-rich.gif*, *tammy.gif*, *tommyrw.gif*, *tori-crucify.gif*, *tori-entweekly.gif*, *tori-live3.gif*, *torrance.gif*, *tp-reza-girosi.gif*, *tp.gif*, *tree-roots.gif*, *trek-trio.gif*, *trekcolr.gif*, *tress-photo-2.gif*, *tress-photo.gif*, *u2-cover.gif*, *uprooted-tree.gif*, *voyager2.gif*, *wall.gif*, *window.gif*, *wxm.gif*, *yellow-pages.gif*, *ysato.gif*.


Fig. 7. Examples of detecting rotated faces using the DFA–SVM method.

experiments are from diverse sources, while the training images are from only one database, i.e., the frontal face images of the FERET Batch 15 [9]. The test data is thus able to test the generalization performance of our DFA–SVM method.

The basic search procedure of the DFA–SVM method is sliding a $16 \times 16$ window across all possible locations and scales in a test image. Specifically, the window is shifted pixel-by-pixel and the incremental scale is a factor of 1.125. Note that detecting faces using a scale of $k \times k$ actually involves two operations: first resize the image by a ratio of $16/k$, and then slide a $16 \times 16$ window in the resized image. Fig. 6 (a) shows the detection results of searching from the scale of $12 \times 12$ to $120 \times 120$. Note that even though the sizes of faces appeared in Fig. 6(a) are smaller than $20 \times 20$, the DFA–SVM method searches at a wide

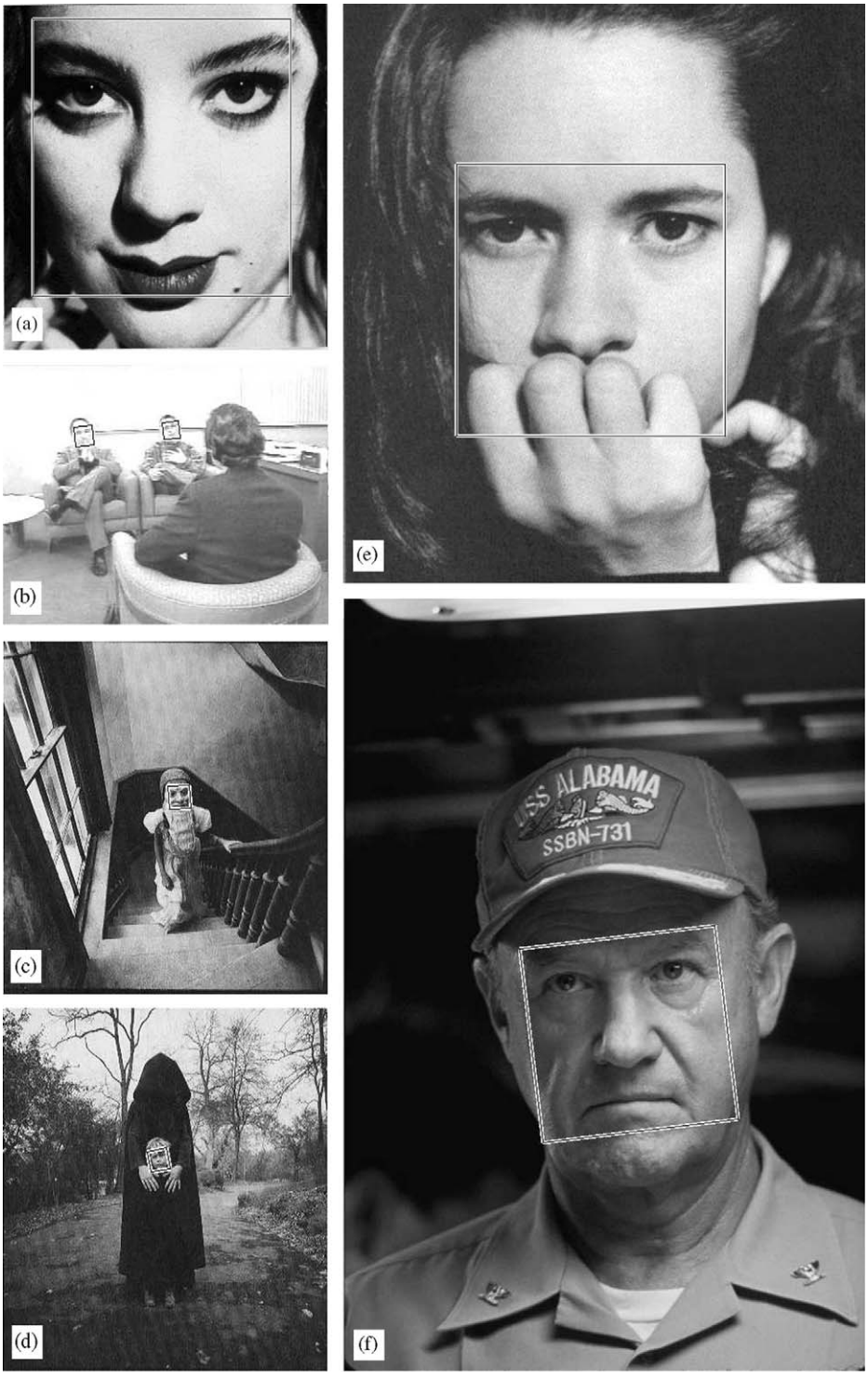



Fig. 8. Examples of detecting faces that are either very large or very small using the DFA–SVM method.

spectrum of scales without leading to any false detection. Similarly, Figs. 6(b)–(e) display examples of multiple face detection in different scales. Note that Figs. 6(b), (c), and (e) show face detection performance of the DFA–SMV method in low contrast images, and Figs. 6(b), (d), and (e) show the detections of slightly rotated faces. All the faces in Fig. 6 are successfully detected by the DFA–SMV method.

Fig. 7 shows some examples of detecting multiple frontal faces across different rotation angles. All the faces in Fig. 7 are correctly detected, but there is one false detection in

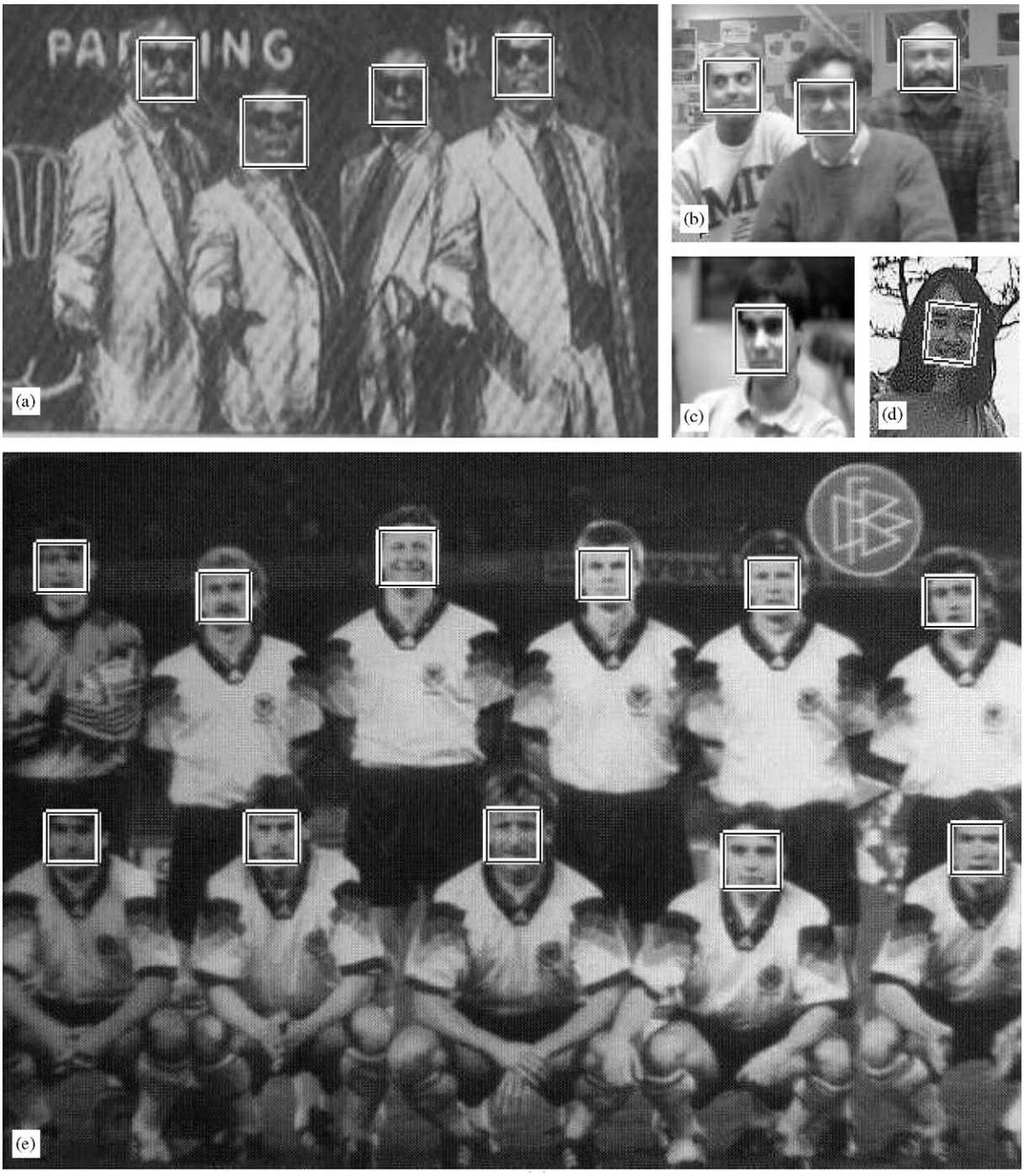



Fig. 9. Examples of detecting faces in low quality images using the DFA–SVM method.

Fig. 7(a). The false detection occurs because the configuration of that pattern resembles a human face. Note that because our DFA–SVM face detection method is trained using only upright frontal faces, the rotated faces are detected by rotating test images to predefined degrees, such as ±5°, ±10°, ±15°, ±20°, and ±25°. For example, in Fig. 7(a), the DFA–SVM face detection method searches from the scale of 21 × 21 to 210 × 210 with 15° and −25° rotation angles.

Fig. 8 shows examples of detecting faces that are either very large or very small. Fig. 7(a), for example, displays the detection of the largest face in the test set (360 × 360) and Fig. 8(b) shows the detection of the smallest face in the test set (13 × 13). Faces in Figs. 8(c)–(f) are detected at the scales of 16 × 16, 16 × 16, 290 × 290, and 210 × 210, respectively.

Fig. 9 shows examples of detecting faces in low quality images. Again, all the faces are correctly detected by the DFA–SVM method. In particular, Fig. 9(a) displays


Fig. 10. Examples of detecting faces with illumination and slightly pose-angled variations using the DFA–SVM method.

detection of faces with dark glasses; Figs. 9(b)–(e) show detections of faces with blurred facial features.

Fig. 10 shows examples of detecting faces with illumination and slightly pose-angled variations. Specifically, Figs. 10(a), (b), and (e) show images containing slightly pose-angled faces, and our DFA–SVM method successfully detects all the faces in these images. However, the DFA–SVM method cannot detect large pose-angled faces, such as the one shown in Fig. 10(d). The reason of such a missed detection is that our DFA–SVM method is only trained by the upright frontal face images, i.e., 600 FERET frontal face images, which do not include any pose-angled face. Fig. 10(c) shows an image with uneven lighting, and the face in this image displays one side brighter than the other side. Still, the face is successfully detected by the DFA–SVM method.

Among the state-of-the-art face detection methods, Schneiderman–Kanade's method [11] is publicly available: *http://vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi*. We

therefore compare our DFA–SVM face detection method with Schneiderman–Kanade's method [11]. This method has two thresholds, the frontal detection threshold and the profile detection threshold, which control the number of faces detected and the number of false detections. Table 1 shows the comparative face detection performance of Schneiderman–Kanade's method and our DFA–SVM method. Note that the two numbers in the parentheses correspond to the frontal detection threshold and the profile detection threshold, respectively. Experimental results show that Schneiderman–Kanade's method achieves 96.1% face detection accuracy with 41 false detections when the thresholds are set to be 1.0. Note that the number of false detections of Schneiderman–Kanade's method counted here only refers to the frontal face false detections, and it does not include the false detections caused by profile face detection. The face detection rate of Schneiderman–Kanade's method decreases when the thresholds get larger. Our DFA–SVM method, achieving 98.2% face detection accuracy, thus compares favorably against the state-of-the-art face detection methods, such as Schneiderman–Kanade's method [11].

Table 1
Comparative face detection performance of Schneiderman–Kanade's method and our DFA–SVM face detection method using 92 images containing 282 faces from the MIT–CMU test sets

| Method | Faces detected | False detections | Detection rate (%) |
|---|---|---|---|
| Schneiderman–Kanade's method (1.0, 1.0) | 271 | 41 | 96.1 |
| Schneiderman–Kanade's method (2.0, 2.0) | 264 | 5 | 93.6 |
| Schneiderman–Kanade's method (3.0, 2.0) | 255 | 1 | 90.4 |
| The DFA–SVM method | 277 | 2 | 98.2 |

### 4.4. Computational efficiency

We apply two criteria to improve the computational efficiency of the DFA–SVM method: the single response criterion and the early exclusion criterion. The single response criterion avoids multiple responses to a singe face, while the early exclusion criterion applies heuristic procedures to exclude subimages that cannot be face.

The single response criterion states that a face should be marked by a single square boundary. Let the DFA–SVM method scan the test images by moving a $16 \times 16$ window from top to bottom, and then from left to right. For the sake of simplicity, we use the upper left pixel to represent a $16 \times 16$ subimage in the following discussion. Fig. 11(a) shows the idea of the single response criterion. Suppose a face is detected at $p$, the DFA–SVM method searches its $7 \times 7$ neighborhood to find a face that lies closest to the face class. Note that due to the predefined searching order, half of these neighbors have already been visited, and the unsearched neighbors are the pixels inside the area $A$. Among the 24 neighbors in the area $A$, suppose $q$ represents a face that lies closest to the face class, then the 542 pixels inside the area $B$ should not be searched again because any detection inside this area will overlap with the face detected at $q$. As a result, the single response criterion improves the computational efficiency by eliminating the large search area, $B$.

The single response criterion also benefits face detection in multiple scales. Fig. 11(b) shows the idea of the single response criterion applied to multiple scale face detection. Suppose a face is detected at the scale of $a \times a$, when the DFA–SVM method tries to detect faces at another scale, say $b \times b$, the area $R$ should not be searched because any

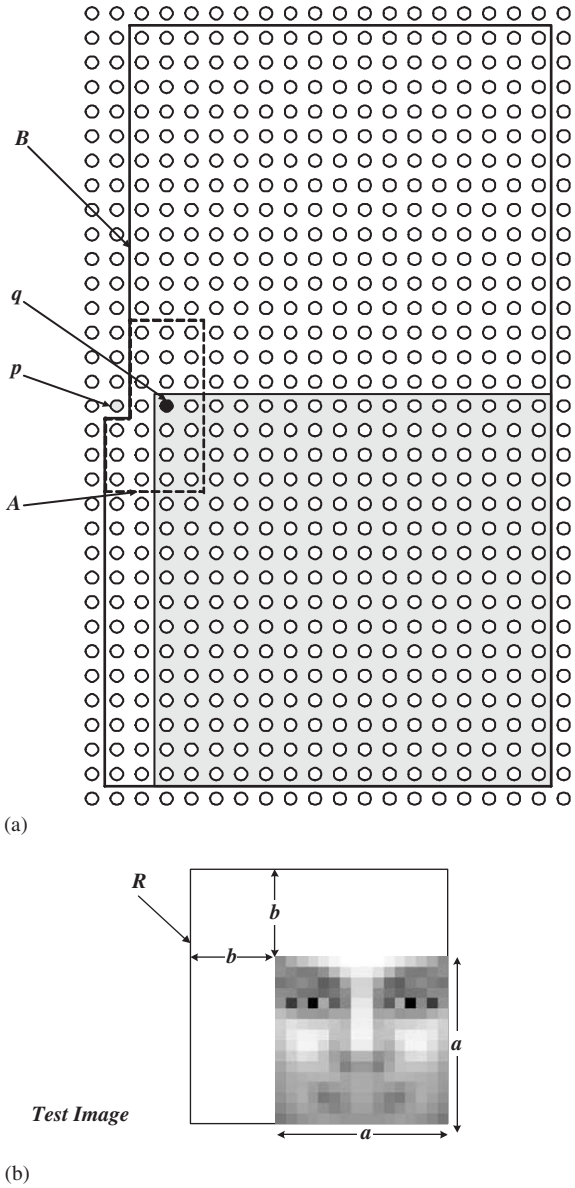



Fig. 11. The single response criterion: (a) a face detected at $q$ eliminates the area $B$ from being searched; (b) a face detected at the scale of $a \times a$ eliminates the area $R$ from being searched when the face detector scans at the scale of $b \times b$.

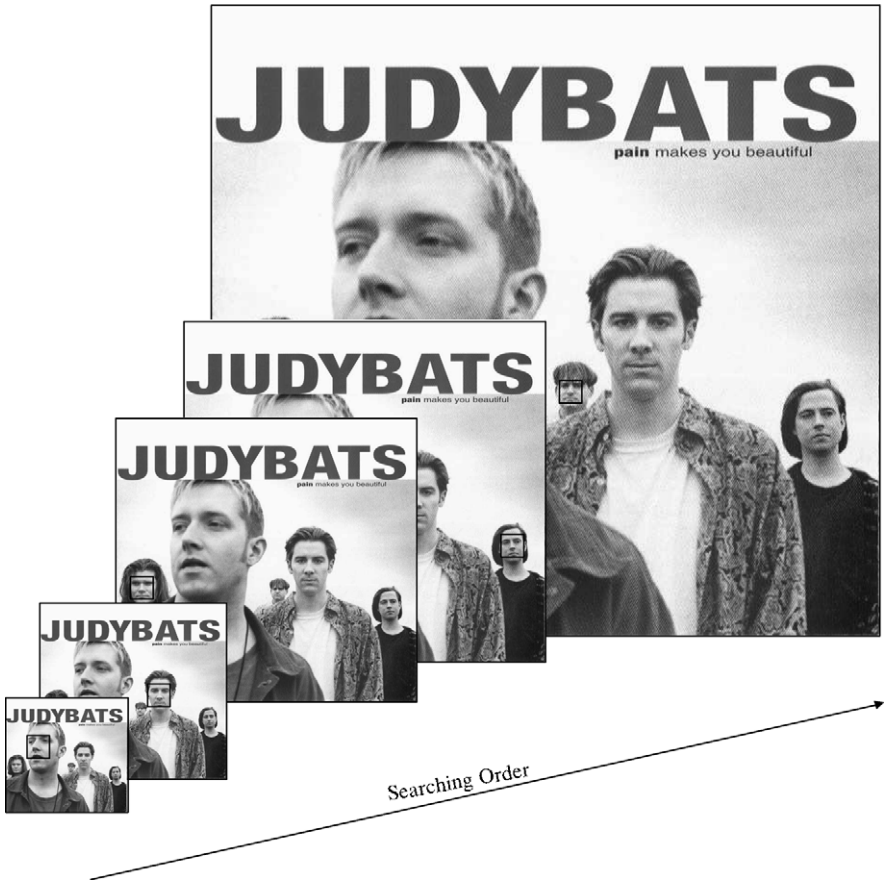



Fig. 12. Multiple scale face detection. The DFA–SVM method detects faces in the order of descending scales, and such order benefits the computational efficiency by eliminating as much search region as possible at each detection scale.

detection inside this area will overlap with the face detected at the scale of $a \times a$. Note that the multiple scale searching in the DFA–SVM method always starts with the largest search scale and then goes to the smaller ones. The single response criterion is thus able to eliminate as much search region as possible and improve the overall computational efficiency. Fig. 12 demonstrates the idea of multiple scale face detection of our DFA–SVM method. From the largest to the smallest scale, faces are detected at the scales of $120 \times 120$, $80 \times 80$, $60 \times 60$, $40 \times 40$, and $30 \times 30$, respectively. Note that the area $R$ should be shrunk by one or two pixels in order to detect closely adjacent or slightly overlapping faces as shown in Fig. 10(d).

To further improve the computational efficiency, the early exclusion criterion defines heuristic procedures to exclude subimages that cannot be face at all. Fig. 13 shows a $16 \times 16$ subimage with five labeled regions corresponding to the left eye (A), the nose bridge (B), the right eye (C), the nose (D), and the mouth (E). The early exclusion criterion first calculates the variances in the regions D and E, respectively, and excludes the subimage as face candidate if either variance is smaller than its predefined threshold. If the subimage is not excluded so far, the early exclusion criterion then calculates the mean values in the regions A, B, and C. Based on these three mean values, two average values, $m_A$ and $m_C$, are derived by averaging the intensity values of the pixels whose intensity values are smaller than the mean values of the regions A and C, respectively; and another average value, $m_B$, is computed by taking the average of the intensity values of the pixels whose intensity values are larger

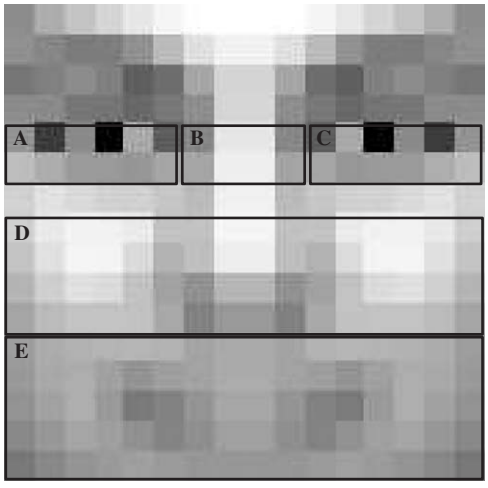



Fig. 13. A $16 \times 16$ subimage with five labeled regions corresponding to the left eye (A), the nose bridge (B), the right eye (C), the nose (D), and the mouth (E).

than the mean value of the region B. Finally, the early exclusion criterion eliminates the subimage as face candidate if $m_B < \kappa m_A$ or $m_B < \kappa m_C$, where $\kappa$ is a control factor. On a computer with 1.2 GHz Pentium III processor and 128 MB main memory, our DFA–SVM face detection method can process a $240 \times 320$ gray scale image in half a second.

## 5. Conclusions

This paper presents a novel face detection method by integrating DFA, face class modeling, and SVM. Discriminating feature analysis derives a feature vector by combining the input image, its 1-D Haar wavelet representation, and its amplitude projections. Face class modeling, then, estimates the PDF of the face class and defines a distribution-based measure for face and nonface classification. And finally, SVM together with the distribution-based measure classifies the patterns in an input image into either the face class or the nonface class. Experiments using images from the MIT–CMU test sets show the feasibility of our new face detection method.

The DFA–SVM method differs from the Bayesian discriminating feature (BDF) method [28] in the following aspects: (i) The BDF method applies the Bayes classifier for face and nonface classification, while the DFA–SVM method applies the distribution-based measure and SVM for face and nonface classification. (ii) The BDF method parametrically models both the face class and the nonface class, while the DFA–SVM method does not model the nonface class at all. (iii) The DFA–SVM method applies a coarse-to-fine classification strategy, the single response criterion, and the early exclusion criterion to improve computational efficiency.

Our future research will be focused on extending our DFA–SVM method to detect pose-angled and partially occluded faces. Towards that end, we will expend our training set to incorporate more facial variations and pose-angled images. We will also investigate the optimal combination of the distribution-based measure and the SVM output values. One possibility is to apply stochastic search algorithms, such as genetic algorithms, to search for the optimal combination.

## Acknowledgements

This work was partially supported by the TSWG R&D Contract N41756-03-C-4026.

## References

[1] A. Pentland, Looking at people: sensing for ubiquitous and wearable computing, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 107–119.
[2] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[3] C. Liu, Gabor-based kernel PCA with fractional power polynomial models for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 572–581.
[4] C. Liu, Enhanced independent component analysis and its application to content based face image retrieval, IEEE Trans. Systems Man Cybernet., Part B: Cybernetics 34 (2) (2004) 1117–1127.
[5] J. Daugman, Face and gesture recognition: overview, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 675–676.
[6] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: a survey, Proc. IEEE 83 (5) (1995) 705–740.
[7] A. Samal, P.A. Iyengar, Automatic recognition and analysis of human faces and facial expression: a survey, Pattern Recognition 25 (1) (1992) 65–77.
[8] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 696–710.
[9] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, The FERET database and evaluation procedure for face recognition algorithms, Image Vision Comput. 16 (1998) 295–306.
[10] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 23–38.
[11] H. Schneiderman, T. Kanade, A statistical method for 3D object detection applied to faces and cars, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2000, pp. 746–751.
[12] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, in: Proceedings of the Computer Vision and Pattern Recognition, 1994, pp. 84–91.
[13] A.L. Yuille, Deformable templates for face recognition, J. Cognitive Neurosci. 3 (1) (1991) 59–70.
[14] K.K. Sung, Learning and example selection for object and pattern detection, Ph.D. Thesis, AI Lab, MIT, 1996.
[15] E. Hjelmas, B.K. Low, Face detection: a survey, Comput. Vision Image Understanding 83 (2001) 236–274.
[16] M.H. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 24 (1) (2002) 34–58.
[17] K.K. Sung, T. Poggio, Example-based learning for view-based human face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 39–51.
[18] H.A. Rowley, S. Baluja, T. Kanade, Rotation invariant neural network-based face detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, 1998, pp. 38–44.
[19] H. Schneiderman, T. Kanade, Probabilistic modeling of local appearance and spatial relationships for object recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, 1998, pp. 45–51.
[20] M.H. Yang, N. Ahuja, D. Kriegman, Face detection using mixtures of linear subspaces, in: Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition, Grenoble, France, 2000, pp. 70–76.
[21] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, 2001, pp. 511–518.
[22] B. Heisele, T. Poggio, M. Pontil, Face detection in still gray images, A.I. memo AIM-1687, Artificial Intelligence Laboratory, MIT, 2000.
[23] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE Trans. Pattern Anal. Mach. Intell. 23 (4) (2001) 349–361.
[24] P. Ho, Rotation invariant real-time face detection and recognition system, A.I. memo AIM-2001-010, Artificial Intelligence Laboratory, MIT, 2001.
[25] S.C. Dass, A.K. Jain, Markov face models, in: The Eighth IEEE International Conference on Computer Vision, Vancouver, Canada, 2001, pp. 680–687.
[26] R.J. Qian, T.S. Huang, Object detection using hierarchical MRF and MAP estimation, in: Proceedings of the Computer Vision and Pattern Recognition, 1997, pp. 186–192.

[27] R. Hsu, M. Abdel-Mottaleb, A. Jain, Face detection in color images, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 696–706.
[28] C. Liu, A Bayesian discriminating features method for face detection, IEEE Trans. Pattern Anal. Mach. Intell. 25 (6) (2003) 725–740.
[29] Y.N. Vapnik, The Nature of Statistical Learning Theory, second ed., Springer, Berlin, 1999.
[30] E. Osuna, R. Freund, F. Girosi, Training support vector machines: an application to face detection, in: Proceedings of the Computer Vision and Pattern Recognition, Puerto Rico, 1997.
[31] C.P. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, in: Proceedings of the International Conference on Computer Vision, Bombay, India, 1998, pp. 555–562.
[32] S. Romdhani, P. Torr, B. Schölkopf, A. Blake, Computationally efficient face detection, in: The Eighth IEEE International Conference on Computer Vision, Vancouver, Canada, 2001.
[33] M.S. Bartlett, G. Littlewort, I. Fasel, J.R. Movellan, Real time face detection and facial expression recognition: development and applications to human computer interaction, in: Proceedings of the Computer Vision and Pattern Recognition, Madison, Wisconsin, 2003.
[34] B. Heisele, T. Serre, S. Prentice, T. Poggio, Hierarchical classification and feature reduction for fast face detection with support vector machines, Pattern Recognition 36 (9) (2003) 2007–2017.
[35] C. Liu, H. Wechsler, Robust coding schemes for indexing and retrieval from large face databases, IEEE Trans. Image Process. 9 (1) (2000) 132–137.
[36] C. Liu, H. Wechsler, Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition, IEEE Trans. Image Process. 11 (4) (2002) 467–476.

**About the Author**—PEICHUNG SHIH received his M.S. degree with summa cum laude in Information Systems in 2002 from New Jersey Institute of Technology. He is a Ph.D. candidate in Computer Science at New Jersey Institute of Technology. His research interests lie in the field of image and video processing, computer vision, and pattern recognition with applications toward face recognition. His present work includes developing novel face detection systems by applying computer vision concepts and statistical learning theory, and designing robust face recognition models by utilizing color information and genetic computations.

**About the Author**—CHENGJUN LIU received the Ph.D. from George Mason University in 1999, and he is presently an Assistant Professor of Computer Science at New Jersey Institute of Technology. His research interests are in computer vision, pattern recognition, image processing, evolutionary computation, and neural computation. His recent research has been concerned with the development of novel and robust methods for image/video retrieval and object detection, tracking and recognition based upon statistical and machine learning concepts.