In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 34–58, January, 2002.

Detecting Faces in Images: A Survey

Ming-Hsuan Yang David Kriegman Narendra Ahuja

Abstract

Images containing faces are essential to intelligent vision-based human computer interaction, and research efforts in face processing include face recognition, face tracking, pose estimation, and expression recognition. However, many reported methods assume that the faces in an image or an image sequence have been identified and localized. To build fully automated systems that analyze the information contained in face images, robust and efficient face detection algorithms are required. Given a single image, the goal of face detection is to identify all image regions which contain a face regardless of its three-dimensional position, orientation, and lighting conditions. Such a problem is challenging because faces are non-rigid and have a high degree of variability in size, shape, color, and texture. Numerous techniques have been developed to detect faces in a single image, and the purpose of this paper is to categorize and evaluate these algorithms. We also discuss relevant issues such as data collection, evaluation metrics, and benchmarking. After analyzing these algorithms and identifying their limitations, we conclude with several promising directions for future research.[1]

keywords

Face detection, face recognition, object recognition, view-based recognition, statistical pattern recognition, machine learning.

1 Introduction

With the ubiquity of new information technology and media, more effective and friendly methods for human computer interaction (HCI) are being developed which do not rely on traditional devices such as keyboards, mice, and displays. Furthermore, the ever decreasing price/performance ratio of computing coupled with recent decreases in video image acquisition cost imply that computer vision systems can be deployed in desktop and embedded systems [111] [112] [113]. The rapidly expanding research in face processing is based on the premise that information about a user's identity, state, and intent can be extracted from images, and that computers can then react accordingly, e.g., by observing a person's facial expression. In the last five years, face and facial expression recognition have attracted much attention though they have been studied for more than twenty years by psychophysicists, neuroscientists, and engineers. Many research demonstrations and commercial applications have been developed from these efforts. A first step of any face processing system is detecting the locations in images where faces are present. However, face detection from a single image is a challenging task because of variability in scale, location, orientation (up-right, rotated), and pose (frontal, profile). Facial expression, occlusion, and lighting conditions also change the overall appearance of faces.

We now give a definition of *face detection*: Given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image, and if present, return the image location and extent of each face. The challenges associated with face detection can be attributed to the following factors:

- Pose: The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.
- Presence or absence of structural components: Facial features such as beards, mustaches, and glasses may or may not be present, and there is a great deal of variability amongst these components including shape, color, and size.

[1]This manuscript is a draft of the paper appearing in PAMI 2002. An early version of this paper also appeared at `http://vision.ai.uiuc.edu/mhyang/face-detection-suvey.html` in March, 1999.

- Facial expression: The appearance of faces are directly affected by a person's facial expression.
- Occlusion: Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.
- Image orientation: Face images directly vary for different rotations about the camera's optical axis.
- Imaging conditions: When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

There are many closely related problems to face detection. *Face localization* aims to determine the image position of a single face; this is a simplified detection problem with the assumption that an input image contains only one face [85] [103]. The goal of *facial feature detection* is to detect the presence and location of features such as eyes, nose, nostrils, eyebrow, mouth, lips, ears, etc. with the assumption that there is only one face in an image [28] [54]. *Face recognition* or *face identification* compares an input image (probe) against a database (gallery) and reports a match, if any [163] [133] [18]. The purpose of *face authentication* is to verify the claim of the identity of an individual in an input image [158] [82], while *face tracking* methods continuously estimate the location and possibly the orientation of a face in an image sequence in real time [30] [39] [33]. *Facial expression recognition* concerns identifying the affective states (happy, sad, disgusted, etc) of humans [40] [35]. Evidently, face detection is the first step in any automated system which solves the above problems. It is worth mentioning that many papers use the term "face detection", but the methods and the experimental results only show that a single face is localized in an input image. In this paper, we differentiate *face detection* from *face localization* since the latter is a simplified problem of the former. Meanwhile, we focus on face detection methods rather than tracking methods.

While numerous methods have been proposed to detect faces in a single image of intensity or color images, we are unaware of any surveys on this particular topic. A survey of early face recognition methods before 1991 was written by Samal and Iyengar [133]. Chellapa, Wilson, and Sirohey wrote a more recent survey on face recognition and some detection methods [18].

Among the face detection methods, the ones based on learning algorithms have attracted much attention recently and have demonstrated excellent results. Since these data-driven methods rely heavily on the training sets, we also discuss several databases suitable for this task. A related and important problem is how to evaluate the performance of the proposed detection methods. Many recent face detection papers compare the performance of several methods, usually in terms of detection and false alarm rates. It is also worth noticing that many metrics have been adopted to evaluate algorithms, such as learning time, execution time, the number of samples required in training, and the ratio between detection rates and false alarms. Evaluation becomes more difficult when researchers use different definitions for detection and false alarm rates. In this paper, *detection rate* is defined as the ratio between the number of faces correctly detected and the number faces determined by a human. An image region identified as a face by a classifier is considered to be correctly detected if the image region covers more than certain percentage of a face in the image (See Section 3.3 for details). In general, detectors can make two types of errors: *False negatives* in which faces are missed resulting in low detection rates, and *false positives* in which an image region is declared to be face, but it is not. A fair evaluation should take these factors into consideration since one can tune the parameters of one's method to increase the detection rates while also increasing the number of false detections. In this paper, we discuss the benchmarking data sets and the related issues in a fair evaluation.

With over 150 reported approaches to face detection, the research in face detection has broader implications for computer vision research on object recognition. Nearly all model-based or appearance-based approaches to 3-D object recognition have been limited to rigid objects while attempting to robustly perform identification over a broad range of camera locations and illumination conditions. Face detection can be viewed as a two-class recognition problem in which an image region is classified as being a "face" or "nonface". Consequently, face detection is one of the few attempts to recognize from images (not abstract representations) a class of objects for which there is a great deal of within-class variability (described previously). It is also one of the few classes of objects for which this variability has been captured using large training sets of images, and so some of the detection techniques may be applicable to a much broader class of recognition problems.

Face detection also provides interesting challenges to the underlying pattern classification and learning techniques. When a raw or filtered image is considered as input to a pattern classifier, the dimension of the feature space is extremely large (i.e., the number of pixels in normalized training images). The classes of face and non-face images are decidedly characterized by multimodal distribution functions and effective decision boundaries are likely to be non-linear in the image space. To be effective, either classifiers must be able to extrapolate from a modest number of training samples or be efficient when dealing with a very large number of these high-dimensional training samples.

With an aim to give a comprehensive and critical survey of current face detection methods, this paper is organized as follows. In Section 2 we give a detailed review of techniques to detect faces in a single image. Benchmarking databases and evaluation criteria are discussed in Section 3. We conclude this paper with discussion of several promising directions for face detection in Section 4.

Though we report error rates for each method when available, tests are often done on unique data sets, and so comparisons are often difficult. We indicate those methods that have been evaluated with a publicly available test set. It can be assumed that a unique data set was used if we do not indicate the name of the test set.

2 Detecting Faces In A Single Image

In this section, we review existing techniques to detect faces from a single intensity or color image. We classify single image detection methods into four categories; some methods clearly overlap category boundaries and are discussed at the end of this section.

1. Knowledge-based methods: These rule-based methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. These methods are designed mainly for face localization.

2. Feature invariant approaches: These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use the these to locate faces. These methods are designed mainly for face localization.

3. Template matching methods: Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed for detection. These methods have been used for both face localization and detection.

4. Appearance-based methods: In contrast to template matching, the models (or templates) are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection.

Table 1 summarizes algorithms and representative works for face detection in a single image within these four categories. Below, we discuss the motivation and general approach of each category. This is followed by a review of specific methods including a discussion of their pros and cons. We suggest ways to further improve these methods in Section 4.

2.1 Knowledge-Based Top-Down Methods

In this approach, face detection methods are developed based on the rules derived from the researcher's knowledge of human faces. It is easy to come up with simple rules to describe the features of a face and their relationships. For example, a face often appears in an image with two eyes that are symmetric to each other, a nose and a mouth. The relationships between features can be represented by their relative distances and positions. Facial features in an input image are extracted first, and face candidates are identified based on the coded rules. A verification process is usually applied to reduce false detections.

One problem with this approach is the difficulty in translating human knowledge into well-defined rules. If the rules are detailed (i.e., strict), they may fail to detect faces that do not pass all the rules. If the rules are too general, they may give many false positives. Moreover, it is difficult to extend this approach to detect

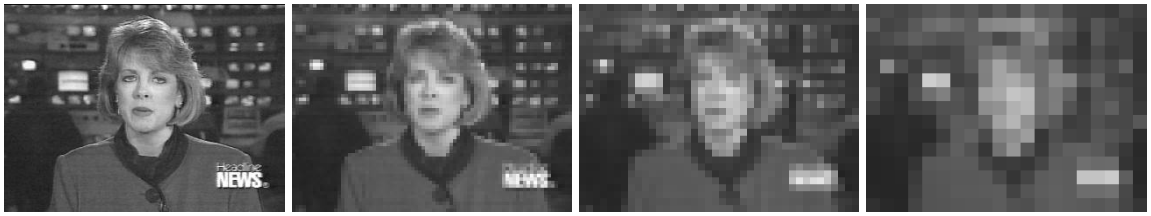
Table 1: Categorization of methods for face detection in a single image

Approach	Representative Works
Knowledge-based	
	Multiresolution rule-based method [170]
Feature invariant	
– Facial Features	Grouping of edges [87] [178]
– Texture	Space Gray-Level Dependence matrix (SGLD) of face pattern [32]
– Skin Color	Mixture of Gaussian [172] [98]
– Multiple Features	Integration of skin color, size and shape [79]
Template matching	
– Predefined face templates	Shape template [28]
– Deformable Templates	Active Shape Model (ASM) [86]
Appearance-based method	
– Eigenface	Eigenvector decomposition and clustering [163]
– Distribution-based	Gaussian distribution and multilayer perceptron [154]
– Neural Network	Ensemble of neural networks and arbitration schemes [128]
– Support Vector Machine (SVM)	SVM with polynomial kernel [107]
– Naive Bayes Classifier	Joint statistics of local appearance and position [140]
– Hidden Markov Model (HMM)	Higher order statistics with HMM [123]
– Information-Theoretical Approach	Kullback relative information [89] [24]

faces in different poses since it is challenging to enumerate all possible cases. On the other hand, heuristics about faces work well in detecting frontal faces in uncluttered scenes.

Yang and Huang used a hierarchical knowledge-based method to detect faces [170]. Their system consists of three levels of rules. At the highest level, all possible face candidates are found by scanning a window over the input image and applying a set of rules at each location. The rules at a higher level are general descriptions of what a face looks like while the rules at lower levels rely on details of facial features. A multi-resolution hierarchy of images is created by averaging and subsampling, and an example is shown in Figure 1. Examples of the coded rules used to locate face candidates in the lowest resolution include: "the center part of the face (the dark shaded parts in Figure 2) has four cells with a basically uniform intensity", "the upper round part of a face (the light shaded parts in Figure 2) has a basically uniform intensity", and "the difference between the average gray values of the center part and the upper round part is significant." The lowest resolution (Level 1) image is searched for face candidates, and these are further processed at finer resolutions. At Level 2, local histogram equalization is performed on the face candidates received from Level 2, followed by edge detection. Surviving candidate regions are then examined at Level 3 with another set of rules that respond to facial features such as the eyes and mouth. Evaluated on a test set of 60 images, this system located faces in 50 of the test images while there are 28 images in which false alarms appear. One attractive feature of this method is that a coarse-to-fine or focus-of-attention strategy is used to reduce the required computation. Although it does not result in a high detection rate, the ideas of using a multiresolution hierarchy and rules to guide search have been used in later face detection works [81].

Kotropoulos and Pitas [81] presented a rule-based localization method which is similar to [71] and [170]. First, facial features are located with a projection method that Kanade successfully used to locate the boundary of a face [71]. Let $I(x, y)$ be the intensity value of an $m \times n$ image at position (x, y), the horizontal and vertical projections of the image are defined as $HI(x) = \sum_{y=1}^{n} I(x, y)$ and $VI(y) = \sum_{x=1}^{m} I(x, y)$. The horizontal profile of an input image is obtained first, and then the two local minima, determined by detecting abrupt changes in HI, are said to correspond to the left and right side of the head. Similarly, the vertical profile is obtained, and the local minima are determined for the locations of mouth lips, nose tip and eyes. These detected features constitute a facial candidate. Figure 3(a) shows one example where the boundaries of the face correspond to the local minimum where abrupt intensity changes occur. Subsequently, eyebrow/eyes, nostrils/nose and the mouth detection rules are used to validate these candidates. The proposed method has been tested using a set of faces in frontal views extracted from the European ACTS M2VTS (MultiModal



(a) n=1, original image (b) n=4 (c) n=8 (d) n=16

Figure 1: Original and corresponding low resolution images. Each square cell consists of $n \times n$ pixels in which the intensity of each pixel is replaced by the average intensity of the pixels in that cell.

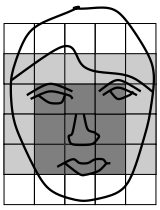


Figure 2: A typical face used in knowledge-based top-down methods: Rules are coded based on human knowledge about the characteristics (e.g., intensity distribution and difference) of the facial regions [170].

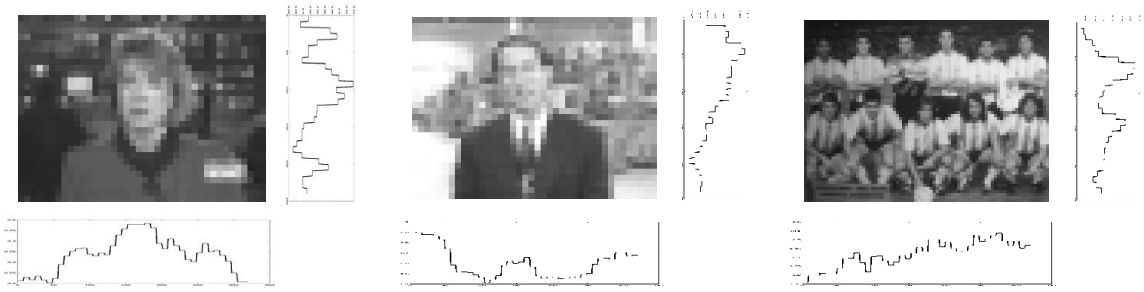
Verification for Teleservices and Security applications) database [116] which contains video sequences of 37 different people. Each image sequence contains only one face in a uniform background. Their method provides correct face candidates in all tests. The detection rate is 86.5% if successful detection is defined as correctly identifying all facial features. Figure 3(b) shows one example in which it becomes difficult to locate a face in a complex background using the horizontal and vertical profiles. Furthermore, this method cannot readily detect multiple faces as illustrated in Figure 3(c). Essentially, the projection method can be effective if the window over which it operates is suitably located to avoid misleading interference.

2.2 Bottom-Up Feature-based Methods

In contrast to the knowledge-based top-down approach, researchers have been trying to find invariant features of faces for detection. The underlying assumption is based on the observation that humans can effortlessly detect faces and objects in different poses and lighting conditions, and so there must exist properties or features which are invariant over these variabilities. Numerous methods have been proposed to first detect facial features and then to infer the presence of a face. Facial features such as eyebrows, eyes, nose, mouth, and hair-line are commonly extracted using edge detectors. Based on the extracted features, a statistical model is built to describe their relationships and to verify the existence of a face. One problem with these feature-based algorithms is that the image features can be severely corrupted due to illumination, noise and occlusion. Feature boundaries can be weakened for faces while shadows can cause numerous strong edges which together render perceptual grouping algorithms useless.

2.2.1 Facial Features

Sirohey proposed a localization method to segment a face from a cluttered background for face identification [145]. It uses an edge map (Canny detector [15]) and heuristics to remove and group edges so that only the ones on the face contour are preserved. An ellipse is then fit to the boundary between the head region and the background. This algorithm achieves 80% accuracy on a database of 48 images with cluttered backgrounds. Instead of using edges, Chetverikov and Lerch presented a simple face detection method using blobs and



(a) n=8 (b) n=8 (c) n=4

Figure 3: Horizontal and vertical profiles. It is feasible to detect a single face by searching for the peaks in horizontal and vertical profiles. However, the same method has difficulty to detect faces in complex backgrounds or multiple faces as shown in 3(b) and 3(c).

streaks (linear sequences of similarly oriented edges) [20]. Their face model consists of two dark blobs and three light blobs to represent eyes, cheekbones, and nose. The model uses streaks to represent the outlines of the faces, eyebrows, and lips. Two triangular configurations are utilized to encode the spatial relationship among the blobs. A low resolution Laplacian image is generated to facilitate blob detection. Next, the image is scanned to find specific triangular occurrences as candidates. A face is detected if streaks are identified around a candidate.

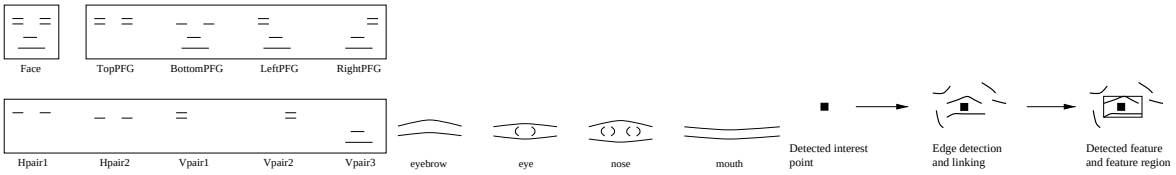
Graf et al. developed a method to locate facial features and faces in gray scale images [54]. After band pass filtering, morphological operations are applied to enhance regions with high intensity that have certain shapes (e.g., eyes). The histogram of the processed image typically exhibits a prominent peak. Based on the peak value and its width, adaptive threshold values are selected in order to generate two binarized images. Connected components are identified in both binarized images to identify the areas of candidate facial features. Combinations of such areas are then evaluated with classifiers, to determine whether and where a face is present. Their method has been tested with head-shoulder images of 40 individuals and with 5 video sequences where each sequence consists of 100 to 200 frames. However, it is not clear how morphological operations are performed and how the candidate facial features are combined to locate a face.

Leung, Burl, and Perona developed a probabilistic method to locate a face in a cluttered scene based on local feature detectors and random graph matching [87]. Their motivation is to formulate the face localization problem as a search problem in which the goal is to find the arrangement of certain facial features that is most likely to be a face pattern. Five features (two eyes, two nostrils, and nose/lip junction) are used to describe a typical face. For any pair of facial features of the same type, (e.g., left-eye, right-eye pair), their relative distance is computed, and over an ensemble of images the distances are modeled by a Gaussian distribution. A facial template is defined by averaging the responses to a set of multi-orientation, multiscale Gaussian derivative filters (at the pixels inside the facial feature) over a number of faces in a data set. Given a test image, candidate facial features are identified by matching the filter response at each pixel against a template vector of responses (similar to correlation in spirit). The top two feature candidates with strongest response are selected to search for the other facial features. Since the facial features cannot appear in arbitrary arrangements, the expected locations of the other features are estimated using a statistical model of mutual distances. Furthermore, the covariance of the estimates can be computed. Thus, the expected feature locations can be estimated with high probability. Constellations are then formed only from candidates that lie inside the appropriate locations, and the most face-like constellation is determined. Finding the best constellation is formulated as a random graph matching problem in which the nodes of the graph correspond to features on a face, and the arcs represent the distances between different features. Ranking of constellations is based on a probability density function that a constellation corresponds to a face versus the probability it was generated by an alternative mechanism (i.e., nonface). They used a set

of 150 images for experiments in which a face is considered correctly detected if any constellation correctly locates three or more features on the faces. This system is able to achieve a correct localization rate of 86%.

Instead of using mutual distances to describe the relationships between facial features in constellations, an alternative method for modeling faces was also proposed by the Leung, Burl, and Perona [13] [88]. The representation and ranking of the constellations is accomplished using the statistical theory of shape, developed by Kendall [75] and others [95]. The shape statistics is a joint probability density function over N feature points, represented by (x_i, y_i), for i-th feature under the assumption that the original feature points are positioned in the plane according to a general $2N$-dimensional Gaussian distribution. They applied the same maximum likelihood (ML) method to determine the location of a face. One advantage of these methods is that partially occluded faces can be located. However, it is unclear whether these methods can be adapted to detect multiple faces effectively in a scene.

In [177] [178], Yow and Cipolla presented a feature-based method that uses a large amount of evidence from the visual image and their contextual evidence. The first stage applies a second derivative Gaussian filter, elongated at an aspect ratio of three to one, to a raw image. Interest points, detected at the local maxima in the filter response, indicate the possible locations of facial features. The second stage examines the edges around these interest points and groups them into regions. The perceptual grouping of edges is based on their proximity and similarity in orientation and strength. Measurements of a region's characteristics, such as edge length, edge strength and intensity variance, are computed and stored in a feature vector. From the training data of facial features, the mean and covariance matrix of each facial feature vector are computed. An image region becomes a valid facial feature candidate if the Mahalanobis distance between the corresponding feature vectors is below a threshold. The labeled features are further grouped based on model knowledge of where they should occur with respect to each other. Each facial feature and grouping is then evaluated using a Bayesian network. One attractive aspect is that this method can detect faces at different orientations and poses. The overall detection rate on a test set of 110 images of faces with different scales, orientations and viewpoints is 85% [179]. However the reported false detection rate is 28%, and the implementation is only effective for faces larger than 60×60 pixels. Subsequently, this approach has been enhanced with active contour models [22] [179]. Figure 4 summarizes their feature-based face detection method.



(a) Face model and the components.

(b) Facial feature models.

(c) Feature selection process.

Figure 4: (a) Yow and Cipolla model a face as a plane with 6 oriented facial features (eyebrows, eyes, nose, and mouth) [179]. (b) Each facial feature is modeled as pairs of oriented edges. (c) Feature selection process starts with interest points, followed by edge detection and linking, and tested by a statistical model (Courtesy of K. C. Yow and R. Cipolla).

Takacs and Wechsler described a biologically motivated face localization method based on a model of retinal feature extraction and small oscillatory eye movements [157]. Their algorithm operates on the conspicuity map, or region of interest, with a retina lattice modeled after the magnocellular ganglion cells in the human vision system. The first phase computes a coarse scan of the image to estimate the location of the face, based on the filter responses of receptive fields. Each receptive field consists of a number of neurons which are implemented with Gaussian filters tuned to specific orientations. The second phase refines the conspicuity map by scanning the image area at a finer resolution to localize the face. The error rate on a test set of 426 images (200 subjects from the FERET database) is 4.69%.

Han et al. developed a morphology-based technique to extract what they call eye-analogue segments for face detection [58]. They argue that eyes and eyebrows are the most salient and stable features of human face and thus useful for detection. They define eye-analogue segments as edges on the contours of eyes. First, morphological operations such as closing, clipped difference, and thresholding are applied to extract pixels at which the intensity values change significantly. These pixels become the eye-analogue pixels in their approach. Then, a labeling process is performed to generate the eye-analogue segments. These segments are used to guide the search for potential face regions with a geometrical combination of eyes, nose, eyebrows and mouth. The candidate face regions are further verified by a neural network similar to [127]. Their experiments demonstrate a 94% accuracy rate using a test set of 122 images with 130 faces.

Recently Amit, Geman, and Jedynak presented a method for shape detection and applied it to detect frontal-view faces in still intensity images [3]. Detection follows two stages: focusing and intensive classification. Focusing is based on spatial arrangements of edge fragments extracted from a simple edge detector using intensity difference. A rich family of such spatial arrangements, invariant over a range of photometric and geometric transformations, are defined. From a set of 300 training face images, particular spatial arrangements of edges which are more common in faces than backgrounds are selected using an inductive method developed in [4]. Meanwhile, the CART algorithm [11] is applied to grow a classification tree from the training images and a collection of false positives identified from generic background images. Given a test image, regions of interest are identified from the spatial arrangements of edge fragments. Each region of interest is then classified as face or background using the learned CART tree. Their experimental results on a set of 100 images from the Olivetti (now AT&T) data set [136] report a false positive rate of 0.2% per 1000 pixels and a false negative rate of 10%.

2.2.2 Texture

Human faces have a distinct texture that can be used to separate them from different objects. Augusteijn and Skufca developed a method that infers the presence of a face through the identification of face-like textures [6]. The texture are computed using second-order statistical features (SGLD) [59] on subimages of 16×16 pixels. Three types of features are considered: skin, hair and others. They used a cascade correlation neural network [41] for supervised classification of textures and a Kohonen self-organizing feature map [80] to form clusters for different texture classes. To infer the presence of a face from the texture labels, they suggest using votes of the occurrence of hair and skin textures. However, only the result of texture classification is reported, not face localization or detection.

Dai and Nakano also applied SGLD model to face detection [32]. Color information is also incorporated with the face-texture model. Using the face texture model, they design a scanning scheme for face detection in color scenes, in which the orange-like parts including the face areas are enhanced. One advantage of this approach is that it can detect faces which are not upright or have features such as beards and glasses. The reported detection rate is perfect for a test set of 30 images with 60 faces.

2.2.3 Skin Color

Human skin color has been used and proven to be an effective feature in many applications from face detection to hand tracking. Although different people have different skin color, several studies have shown that the major difference between lies largely in their intensity rather than their chrominance [54] [55] [172]. Several color spaces have been utilized to label pixels as skin including RGB [66] [67] [137], normalized RGB [102] [29] [149] [172] [30] [105] [171] [77] [151] [120], HSV (or HSI) [138] [79] [147] [146], YCrCb [167] [17], YIQ [31] [32], YES [131], CIE XYZ [19], and CIE LUV [173].

Many methods have been proposed to build a skin color model. The simplest model is to define a region of skin tone pixels using Cr, Cb values [17], i.e., $R(Cr, Cb)$, from samples of skin color pixels. With carefully chosen thresholds, $[Cr_1, Cr_2]$ and $[Cb_1, Cb_2]$, a pixel is classified to have skin tone if its values (Cr, Cb) fall within the ranges, i.e., $Cr_1 \leq Cr \leq Cr_2$ and $Cb_1 \leq Cb \leq Cb_2$. Crowley and Coutaz used a histogram $h(r, g)$ of (r, g) values in normalized RGB color space to obtain the probability of obtaining a particular RGB-vector given that the pixel observes skin [29] [30]. In other words, a pixel is classified to belong to skin color if $h(r, g) \geq \tau$ where τ is a threshold selected empirically from the histogram of samples. Saxe and Foulds proposed an iterative skin identification method that uses histogram intersection in HSV color space [138].

An initial patch of skin color pixels, called the control seed, is chosen by the user and is used to initiate the iterative algorithm. To detect skin color regions, their method moves through the image, one patch at a time, and presents the control histogram and the current histogram from the image for comparison. Histogram intersection [155] is used to compare the control histogram and current histogram. If the match score or number of instances in common (i.e., intersection) is greater than a threshold, the current patch is classified as being skin color. Kjeldsen and Kender defined a color predicate in HSV color space to separate skin regions from background [79] . In contrast to the nonparametric methods mentioned above, Gaussian density functions [14] [77] [173] and a mixture of Gaussians [66] [67] [174] are often used to model skin color. The parameters in a unimodal Gaussian distribution are often estimated using maximum likelihood [14] [77] [173]. The motivation for using a mixture of Gaussians is based on the observation that the color histogram for the skin of people with different ethnic background does not form a unimodal distribution, but rather a multimodal distribution. The parameters in a mixture of Gaussians are usually estimated using an EM algorithm [66] [174]. Recently, Jones and Rehg conducted a large-scale experiment in which nearly 1 billion labeled skin tone pixels are collected (in normalized RGB color space) [69]. Comparing the performance of histogram and mixture models for skin detection, they find histogram models to be superior in accuracy and computational cost.

Color information is an efficient tool for identifying facial areas and specific facial features if the skin color model can be properly adapted for different lighting environments. However, such skin color models are not effective where the spectrum of the light source varies significantly. In other words, color appearance is often unstable due to changes in both background and foreground lighting. Though the color constancy problem has been addressed through the formulation of physics-based models [45], several approaches have been proposed to use skin color in varying lighting conditions. McKenna, Raja, and Gong presented an adaptive color mixture model to track faces under varying illumination conditions [99]. Instead of relying on a skin color model based on color constancy, they used a stochastic model to estimate an object's color distribution on-line and adapt to accommodate changes in the viewing and lighting conditions. Preliminary results show that their system can track faces within a range of illumination conditions. However, this method cannot be applied to detect faces in a single image.

Skin color alone is usually not sufficient to detect or track faces. Recently several modular systems using a combination of shape analysis, color segmentation and motion information for locating or tracking heads and faces in an image sequence have been developed [55] [173] [172] [99] [147]. We review these methods in the next section.

2.2.4 Multiple Features

Recently, numerous methods that combine several facial features have been proposed to locate or detect faces. Most of them utilize global features such as skin color, size, and shape to find face candidates, and then verify these candidates using local, detailed features such as eye brows, nose, and hair. A typical approach begins with the detection of skin-like regions as described in Section 2.2.3. Next, skin-like pixels are grouped together using connected component analysis or clustering algorithms. If the shape of a connected region has an elliptic or oval shape, it becomes a face candidate. Finally, local features are used for verification. However, others such as [17, 63], have used different sets of features.

Yachida et al. presented a method to detect faces in color images using fuzzy theory [19] [169] [168]. They used two fuzzy models to describe the distribution of skin and hair color in CIE XYZ color space. Five (one frontal, and four side views) head-shape models are used to abstract the appearance of faces in images. Each shape model is a 2-D pattern consisting of $m \times n$ square cells where each cell may contain several pixels. Two properties are assigned to each cell: the skin proportion and the hair proportion, which indicate the ratios of the skin area (or the hair area) within the cell to the area of the cell. In a test image each pixel is classified as hair, face, hair/face, and hair/background based on the distribution models, thereby generating skin-like and hair-like regions. The head shape models are then compared with the extracted skin-like and hair-like regions in a test image. If they are similar, the detected region becomes a face candidate. For verification, eye-eyebrow and nose-mouth features are extracted from a face candidate using horizontal edges.

Sobottka and Pitas proposed a method for face localization and facial feature extraction using shape and color [147]. First, color segmentation in HSV space is performed to locate skin-like regions. Connected components are then determined by region growing at a coarse resolution. For each connected component,

the best fit ellipse is computed using geometric moments. Connected components that are well approximated by an ellipse are selected as face candidates. Subsequently these candidates are verified by searching for facial features inside of the connected components. Features, such as eyes and mouths, are extracted based on the observation that they are darker than the rest of a face. In [159] [160], a Gaussian skin color model is used to classify skin color pixels. To characterize the shape of the clusters in the binary image, a set of 11 lowest-order geometric moments is computed using Fourier and radial Mellin transforms. For detection, a neural network is trained with the extracted geometric moments. Their experiments show a detection rate of 85% based on a test set of 100 images.

The symmetry of face patterns has also been applied to face localization [131]. Skin/non-skin classification is carried out using the class-conditional density function in YES color space followed by smoothing in order to yield contiguous regions. Next, an elliptical face template is used to determine the similarity of the skin color regions based on Hausdorff distance [65]. Finally, the eye centers are localized using several cost functions which are designed to take advantage of the inherent symmetries associated with face and eye locations. The tip of the nose and the center of the mouth are then located by utilizing the distance between the eye centers. One drawback is that it is effective only for a single frontal-view face and when both eyes are visible. A similar method using color and local symmetry was presented in [151].

In contrast to pixel-based methods, a detection method based on structure, color and geometry was proposed in [173]. First, multiscale segmentation [2] is performed to extract homogeneous regions in an image. Using a Gaussian skin color model, regions of skin tone are extracted and grouped into ellipses. A face is detected if facial features such as eyes and mouth exist within these elliptic regions. Experimental results show that this method is able to detect faces at different orientations with facial features such as beard and glasses.

Pentland and Kauth proposed a blob representation to extract a compact, structurally meaningful description of multispectral satellite imagery [74]. A feature vector at each pixel is formed by concatenating the pixel's image coordinates to the pixel's spectral (or textural) components; pixels are then clustered using this feature vector to form coherent connected regions, or "blobs." To detect faces, each feature vector consists of the image coordinates and normalized chrominance, i.e., $X = (x, y, \frac{r}{r+g+b}, \frac{g}{r+g+b})$ [149], [105]. A connectivity algorithm is then used to grow blobs, and the resulting skin blob whose size and shape is closest to that of a canonical face is considered as a face.

Range and color have also been employed for face detection by Kim et al. [77]. Disparity maps are computed and objects are segmented from the background with a disparity histogram using the assumption that background pixels have the same depth and they outnumber the pixels in the foreground objects. Using a Gaussian distribution in normalized RGB color space, segmented regions with a skin-like color are classified as faces. Similar approach has been proposed by Darrell et al. for face detection and tracking [33].

2.3 Template Matching

In template matching, a standard face pattern (usually frontal) is manually predefined or parameterized by a function. Given an input image, the correlation values with the standard patterns are computed for the face contour, eyes, nose, and mouth independently. The existence of a face is determined based on the correlation values. This approach has the advantage of being simple to implement. However, it has proven to be inadequate for face detection since it cannot effectively deal with variation in scale, pose and shape. Multiresolution, multiscale, subtemplates, and deformable templates have subsequently been proposed to achieve scale and shape invariance.

2.3.1 Predefined Templates

An early attempt to detect frontal faces in photographs is reported by Sakai, Nagao, and Fujibayashi [132]. They used several subtemplates for the eyes, nose, mouth, and face contour to model a face. Each subtemplate is defined in terms of line segments. Lines in the input image are extracted based on greatest gradient change and then matched against the subtemplates. The correlations between subimages and contour template are computed first to detect candidate locations of faces. Then matching with the other subtemplates is performed at the candidate positions. In other words, the first phase determines focus of attention or region

of interest, and the second phase examines the details to determine the existence of a face. The idea of focus of attention and subtemplates has been adopted by later works on face detection.

Craw, Ellis, and Lishman presented a localization method based on a shape template of a frontal-view face (i.e., the outline shape of a face) [27]. A Sobel filter is first used to extract edges. These edges are grouped together to search for the template of a face based on several constraints. After the head contour has been located, the same process is repeated at different scales to locate features such as eyes, eyebrows, and lips. Craw, Tock, and Bennett later described a localization method using a set of 40 templates to search for facial features and a control strategy to guide and assess the results from the template-based feature detectors [28].

Govindaraju et al. presented a two stage face detection method in which face hypotheses are generated and tested [52] [53] [51]. A face model is built in terms of features defined by the edges. These features describe the curves of the left side, the hair-line, and the right side of a frontal face. The Marr-Hildreth edge operator is used to obtain an edge map of an input image. A filter is then used to remove objects whose contours are unlikely to be part of a face. Pairs of fragmented contours are linked based on their proximity and relative orientation. Corners are detected to segment the contour into feature curves. These feature curves are then labeled by checking their geometric properties and relative positions in the neighborhood. Pairs of feature curves are joined by edges if their attributes are compatible (i.e., if they could arise from the same face). The ratios of the feature pairs forming an edge is compared with the golden ratio, and a cost is assigned to the edge. If the cost of a group of three feature curves (with different labels) is low, the group becomes a hypothesis. When detecting faces in newspaper articles, collateral information, which indicates the number of persons in the image, is obtained from the caption of the input image to select the best hypotheses [52]. Their system reports a detection rate of approximately 70% based on a test set of 50 photographs. However the faces must be upright, unoccluded, and frontal. The same approach has been extended by extracting edges in the wavelet domain by Venkatraman and Govindaraju [165].

Tsukamoto, Lee, and Tsuji presented a qualitative model for face pattern (QMF) [161] [162]. In QMF, each sample image is divided into a number of blocks, and qualitative features are estimated for each block. To parameterize a face pattern, "lightness" and "edgeness" are defined as the features in this model. Consequently, this blocked template is used to calculate "faceness" at every position of an input image. A face is detected if the faceness measure is above a predefined threshold.

Silhouettes have also been used as templates for face localization [134]. A set of basis face silhouettes is obtained using principal component analysis (PCA) on face examples in which the silhouette is represented by an array of bits. These eigen-silhouettes are then used with a generalized Hough transform for localization. A localization method based on multiple templates for facial components was proposed in [150]. Their method defines numerous hypotheses for the possible appearances of facial features. A set of hypotheses for the existence of a face is then defined in terms of the hypotheses for facial components using the Dempster-Shafer theory [34]. Given an image, feature detectors compute confidence factors for the existence of facial features. The confidence factors are combined to determine the measures of belief and disbelief about the existence of a face. Their system is able to locate faces in 88 images out of 94 images.

Sinha used a small set of spatial image invariants to describe the space of face patterns [143] [144]. His key insight for designing the invariant is that while variations in illumination change the individual brightness of different parts of faces (such as eyes, cheeks, and forehead), the relative brightness of these parts remain largely unchanged. Determining pair-wise ratios of the brightness of a few such regions and retaining just the "directions" of these ratios (i.e., is one region brighter or darker than the other?) provides a robust invariant. Thus, observed brightness regularities are encoded as a ratio template which is a coarse spatial template of a face with a few appropriately chosen subregions that roughly correspond to key facial features such as the eyes, cheeks and forehead. The brightness constraints between facial parts are captured by an appropriate set of pairwise brighter-darker relationships between subregions. A face is located if an image satisfies all the pairwise brighter-darker constraints. The idea of using intensity differences between local adjacent regions has later been extend to a wavelet-based representation for pedestrian, car and face detection [109]. Sinha's method has been extended and applied to face localization in an active robot vision system [139] [10]. Figure 5 shows the enhanced template with 23 defined relations. These defined relations are furthered classified into 11 essential relations (solid arrows) and 12 confirming relations (dashed arrows). Each arrow in the figure indicates a relation, with the head of the arrow denoting the second region (i.e., the denominator of

the ratio). A relation is satisfied for face temple if the ratio between two regions exceeds a threshold; and a face is localized if the number of essential and confirming relations exceeds a threshold.

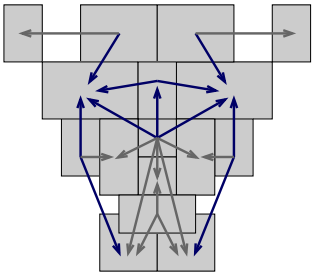


Figure 5: A 14 $\times$ 16 pixel ratio template for face localization based on Sinha method. The template is composed of 16 regions (the gray boxes) and 23 relations (shown by arrows) [139] (Courtesy of B. Scassellati).

A hierarchical template matching method for face detection was proposed by Miao et al. [100]. At the first stage, an input image is rotated from -20° to 20° in steps of 5° in order to handle rotated faces. A multiresolution image hierarchy is formed (See Figure 1), and edges are extracted using the Laplacian operator. The face template consists of the edges produced by six facial components: two eyebrows, two eyes, one nose, and one mouth. Finally, heuristics are applied to determine the existence of a face. Their experimental results show better results in images containing a single face (frontal or rotated) than images with multiple faces.

2.3.2 Deformable Templates

Yuille, Hallinan, and Cohen used deformable templates to model facial features that fit an a priori elastic model to facial features (e.g., eyes) [180] . In this approach, facial features are described by parameterized templates. An energy function is defined to link edges, peaks and valleys in the input image to corresponding parameters in the template. The best fit of the elastic model is found by minimizing an energy function of the parameters. Although their experimental results demonstrate good performance in tracking non-rigid features, one drawback of this approach is that the deformable template must be initialized in the proximity of the object of interest.

In [84], a detection method based on snakes [73] [90] and templates was developed. An image is first convolved with a blurring filter and then a morphological operator to enhance edges. A modified n-pixel (n is small) snake is used to find and eliminate small curve segments. Each face is approximated by an ellipse, and a Hough transform of the remaining snakelets is used to find a dominant ellipse. Thus, sets of four parameters describing the ellipses are obtained and used as candidates for face locations. For each of these candidates, a method similar to the deformable template method [180] is used to find detailed features. If a substantial number of the facial features are found and if their proportions satisfy ratio tests based on a face template, a face is considered to be detected. Lam and Yan also used snakes to locate the head boundaries with a greedy algorithm in minimizing the energy function [85].

Lanitis, Taylor, and Cootes described a face representation method with both shape and intensity information [86]. They start with sets of training images in which sampled contours such as the eye boundary, nose chin/cheek are manually labeled, and a vector of sample points is used to represent shape. They used a point distribution model (PDM) to characterize the shape vectors over an ensemble of individuals, and an approach similar to Kirby and Sirovich [78] to represent shape-normalized intensity appearance. A face-shape PDM can be used to locate faces in new images by using active shape model (ASM) search to estimate the face location and shape parameters. The face patch is then deformed to the average shape, and intensity parameters are extracted. The shape and intensity parameters can be used together for classification. Cootes and Taylor applied a similar approach to localize a face in an image [25]. First, they define a rectangular regions of the image containing instances of the feature of interest. Factor analysis [5] is then applied to fit these training features and obtain a distribution function. Candidate features are determined if the probabilistic measures are above a threshold, and are verified using the ASM. After training this method with 40

images, it is able to locate 35 faces in 40 test images. The ASM approach has also been extended with two Kalman filters to estimate the shape-free intensity parameters and to track faces in image sequences [39].

2.4 Appearance-Based Methods

Contrasted to the template matching methods where templates are pre-defined by experts, the "templates" in appearance-based methods are learned from examples in images. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and nonface images. The learned characteristics are in the form of distribution models, or discriminant functions that are consequently used for face detection. Meanwhile, dimensionality reduction is usually carried out for the sake of computation efficiency and detection efficacy.

Many appearance-based methods can be understood in a probabilistic framework. An image or feature vector derived from an image is viewed as a random variable $\mathbf{x}$, and this random variable is characterized for faces and nonfaces by the class-conditional density functions $p(\mathbf{x}|face)$ and $p(\mathbf{x}|nonface)$. Bayesian classification or maximum likelihood can be used to classify a candidate image location as face or nonface. Unfortunately, a straightforward implementation of Bayesian classification is infeasible because of the high dimensionality of $\mathbf{x}$, because $p(\mathbf{x}|face)$ and $p(\mathbf{x}|nonface)$ are multimodal, and because it is not yet understood if there are natural parameterized forms for $p(\mathbf{x}|face)$ and $p(\mathbf{x}|nonface)$. Hence, much of the work in appearance-based method concerns empirically validated parametric and non-parametric approximations to $p(\mathbf{x}|face)$ and $p(\mathbf{x}|nonface)$.

Another approach in appearance-based methods is to find a discriminant function (i.e., decision surface, separating hyperplane, threshold function) between face and nonface classes. Conventionally image patterns are projected to a lower dimensional space and then a discriminant function is formed (usually based on distance metrics) for classification [163], or a nonlinear decision surface can be formed using multilayer neural networks [128]. Recently, support vector machines and other kernel methods have been proposed. These methods implicitly project patterns to a higher dimensional space and then form a decision surface between the projected face and nonface patterns [107].

2.4.1 Eigenfaces

An early example of employing eigenvectors in face recognition was done by Kohonen [80] in which a simple neural network is demonstrated to perform face recognition for aligned and normalized face images. The neural network computes a face description by approximating the eigenvectors of the image's autocorrelation matrix. These eigenvectors are later known as Eigenfaces.

Kirby and Sirovich demonstrated that images of faces can be linearly encoded using a modest number of basis images [78]. This demonstration is based on the Karhunen-Loève transform [72] [93] [48], which also goes by other names, e.g., principal component analysis [68], and the Hotelling transform [50]. The idea is arguably proposed first by Pearson in 1901 [110] and then by Hotelling in 1933 [62]. Given a collection of n by m pixel training images represented as a vector of size $m \times n$, basis vectors spanning an optimal subspace are determined such that the mean square error between the projection of the training images onto this subspace and the original images is minimized. They call the set of optimal basis vectors eigenpictures since these are simply the eigenvectors of the covariance matrix computed from the vectorized face images in the training set. Experiments with a set of 100 images show that a face image of 91×50 pixels can be effectively encoded using only 50 eigenpictures, while retaining a reasonable likeness (i.e., capturing 95% of the variance).

Turk and Pentland applied principal component analysis to face recognition and detection [163]. Similar to [78], principal component analysis on a training set of face images is performed to generate the Eigenpictures (here called Eigenfaces) which span a subspace (called the face space) of the image space. Images of faces are projected onto the subspace and clustered. Similarly, nonface training images are projected onto the same subspace and clustered. Since images of faces do not change radically when projected onto the face space, while the projection of nonface images appear quite different. To detect the presence of a face in a scene, the distance between an image region and the face space is computed for all locations in the image. The distance from face space is used as a measure of "faceness", and the result of calculating the distance from face space is a "face map". A face can then be detected from the local minima of the

face map. Many works on face detection, recognition, and feature extractions have adopted the idea of eigenvector decomposition and clustering.

2.4.2 Distribution-Based Methods

Sung and Poggio developed a distribution-based system for face detection [152] [154] which demonstrated how the distributions of image patterns from one object class can be learned from positive and negative examples (i.e., images) of that class. 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 image and treated as a 361-dimensional vector or pattern. Next, the patterns are grouped into six face and six nonface clusters using a modified k-means algorithm as shown in Figure 6. Each cluster is represented as a multidimensional Gaussian function with a mean image and a covariance matrix. Figure 7 shows the distance measures in their method. Two distance metrics

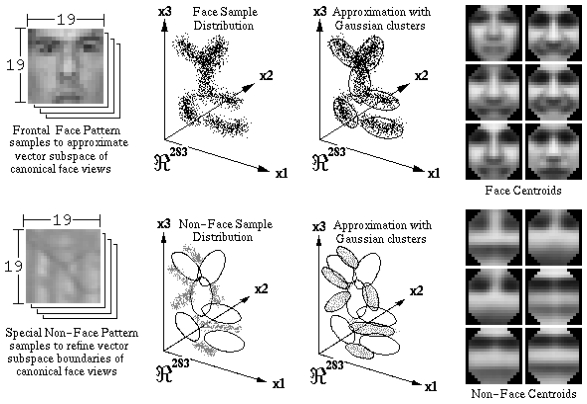


Figure 6: Face and nonface clusters used by Sung and Poggio [154]. Their method estimates density functions for face and nonface patterns using a set of Gaussians. The centers of these Gaussians are shown on the right (Courtesy of K.-K. Sung and T. Poggio).

are computed between an input image pattern and the prototype clusters. The first distance component is the 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 (MLP) network to classify face window patterns from nonface patterns using the twelve pairs of distances to each face and nonface cluster. The classifier is trained using standard backpropagation from a database of 47,316 window patterns. There are 4,150 positive examples of face patterns and the rest are nonface patterns. Note that it is easy to collect a representative sample face patterns, but much more difficult to get a representative sample of nonface patterns. This problem is alleviated by a bootstrap method that selectively adds images to the training set as training progress. Starting with a small set of nonface examples in the training set, the MLP classifier is trained with this database of examples. Then, they run the face detector on a sequence of random images and collect all the nonface patterns that the current system wrongly classifies as faces. These false positives are then added to the training database as new nonface examples. This bootstrap method avoids the problem of explicitly collecting a representative sample of nonface patterns and has been used in later works [107] [128]. A probabilistic visual learning method, based on density estimation in a high-dimensional space using an eigenspace decomposition was developed by Moghaddam and Pentland [103]. Principal component analysis (PCA) is used to define the subspace best representing a set of face patterns. These principal components preserve the major linear correlations in the data and discard the minor ones. This method decomposes the vector space into two mutually exclusive and complementary subspaces: the principal subspace (or feature space) and its orthogonal complement. Therefore, the target density is decomposed into two components: the density in the principal subspace (spanned by the principal components) and its orthogonal complement

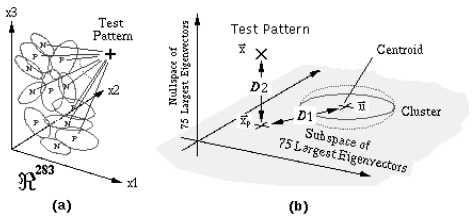


Figure 7: The distance measures used by Sung and Poggio [154]. Two distance metrics are computed between an input image pattern and the prototype clusters. (a) Given a test pattern, the distance between that image pattern and each cluster is computed. A set of 12 distances between the test pattern and the model's 12 cluster centroids. (b) Each distance measurement between the test pattern and a cluster centroid is a two-value distance metric. $\mathcal{D}_1$ is a Mahalanobis distance between the test pattern's projection and the cluster centroid in a subspace spanned by the cluster's 75 largest eigenvectors. $\mathcal{D}_2$ is the Euclidean distance between the test pattern and its projection in the subspace. Therefore, a distance vector of 24 values is formed for each test pattern and is used by a multilayer perceptron to determine whether the input pattern belong to the face class or not (Courtesy of K.-K. Sung and T. Poggio).

(which is discarded in standard PCA) (See Figure 8). A multivariate Gaussian and a mixture of Gaussians

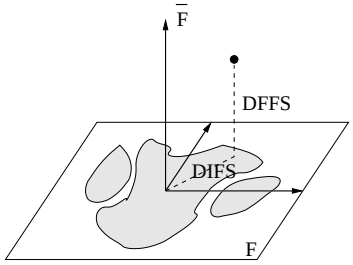


Figure 8: Decomposition of a face image space into the principal subspace F and its orthogonal complement $\overline{F}$ for an arbitrary density. Every data point $\mathbf{x}$ is decomposed into two components: distance in feature space (DIFS) and distance from feature space (DFFS) [103] (Courtesy of B. Moghaddam and A. Pentland).

are used to learn the statistics of the local features of a face. These probability densities are then used for object detection based on maximum likelihood estimation. The proposed method has been applied to face localization, coding and recognition. Compared with the classic eigenface approach [163], the proposed method shows better performance in face recognition. In terms of face detection, this technique has only been demonstrated on localization; see also [76].

In [175], a detection method based on a mixture of factor analysis was proposed. 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 principal component analysis (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 [36] [37] [9] [7] 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, Dayan, and Revow have applied FA to digit recognition, and they compare the performance of PCA and FA models [61]. A mixture model of factor analyzers has recently been extended [49] and applied to face recognition [46]. Both studies show that FA performs better than PCA in digit and face recognition. Since pose, orientation, expression, and lighting 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. They present a

Figure 9: Prototype of each face class using Kohonen's SOM by Yang, Ahuja and Kriegman [175]. Each prototype corresponds to the center of a cluster.

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 an EM algorithm.

A second method in [175] uses Fisher's Linear Discriminant (FLD) to project samples from the high dimensional image space to a lower dimensional feature space. Recently, the Fisherface method [7] and others [156] [181] based on linear discriminant analysis have been shown to outperform the widely used Eigenface method [163] 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 [97] for a discussion about training set size). In the second proposed method, they decompose the training face and nonface samples into several subclasses using Kohonen's Self Organizing Map (SOM) [80]. Figure 9 shows a prototype of each face class. From these re-labeled samples, the within-class and between-class scatter matrices are computed, thereby generating the optimal projection based on FLD. For each subclass, its density is modeled as a Gaussian whose parameters are estimated using maximum likelihood [36]. 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 or not. Both methods in [175] have been tested using the databases in [128] [154] which together consist of 225 images with 619 faces, and experimental results show that these two methods have detection rates of 92.3% for MFA and 93.6% for the FLD-based method.

2.4.3 Neural Networks

Neural networks have been applied successfully in many pattern recognition problems such as optical character recognition, object recognition and autonomous robot driving. Since face detection can be treated as a two class pattern recognition problem, various neural network architectures have been proposed. The advantage of using neural networks for face detection is the feasibility of training a system to capture the complex class conditional density of face patterns. However, one drawback is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance.

An early method using hierarchical neural networks was proposed by Agui et al. [1]. The first stage consists of two parallel subnetworks in which the inputs are intensity values from an original image and intensity values from filtered image using a 3×3 Sobel filter. The inputs to the second stage network consist of the outputs from the subnetworks and extracted feature values such as the standard deviation of the pixel values in the input pattern, a ratio of the number of white pixels to the total number of binarized pixels in a window, and geometric moments. An output value at the second stage indicates the presence of a face in the input region. Experimental results show that this method is able to detect faces if all faces in the test images have the same size. Propp and Samal developed one of the earliest neural networks for face detection

[117]. Their network consists of 4 layers with 1024 input units, 256 units in the first hidden layer, 8 units in the second hidden layer, and 2 output units. A similar hierarchical neural network is later proposed by [70]. The early method by Soulie, Vinnet, and Lamy [148] scans an input image with a time-delay neural network [166] (with a receptive field of 20×25 pixels) to detect faces. To cope with size variation, the input image is decomposed using wavelet transforms. They reported a false negative rate of 2.7% and false positive rate of 0.5% from a test of 120 images. In [164], Vaillant, Monrocq and Le Cun used convolutional neural networks to detect faces in images. Examples of face and nonface images of 20×20 pixels are first created. One neural network is trained to find approximate locations of faces at some scale. Another network is trained to determine the exact position of faces at some scale. Given an image, areas which may contain faces are selected as face candidates by the first network. These candidates are verified by the second network. Burel and Carel [12] proposed a neural network for face detection in which the large number of training examples of faces and nonfaces are compressed into fewer examples using a Kohonen's SOM algorithm [80]. A multi-layer perceptron is used to learn these examples for face/background classification. The detection phase consists of scanning each image at various resolution. For each location and size of the scanning window, the contents are normalized to a standard size, and the intensity mean and variance are scaled to reduce the effects of lighting conditions. Each normalized window is then classified by an MLP.

Feraud and Bernier presented a detection method using autoassociative neural networks [43] [42] [44]. The idea is based on [83] which shows an autoassociative network with five layers is able to perform a nonlinear principal component analysis. One autoassociative network is used to detect frontal-view faces and another one is used to detect turned faces up to 60 degrees to the left and right of the frontal view. A gating network is also utilized to assign weights to frontal and turned face detectors in an ensemble of autoassociative networks. On a small test set of 42 images, they report a detection rate similar to [126]. The method has also been employed in LISTEN [23] and MULTRAK [8].

Lin, Kung, and Lin presented a face detection system using probabilistic decision-based neural network (PDBNN) [91]. The architecture of PDBNN is similar to radial basis function (RBF) network with modified learning rules and probabilistic interpretation. Instead of converting a whole face image into a training vector of intensity values for the neural network, they first extract feature vectors based on intensity and edge information in the facial region that contains eyebrows, eyes and nose. The extracted two feature vectors are fed into two PDBNN's and the fusion of the outputs determine the classification result. Based on a set of 23 images provided by Sung and Poggio [154], their experimental results show comparable performance with the other leading neural network-based face detectors [154] [128].

Among all the face detection methods that used neural networks, the most significant work is arguably done by Rowley, Baluja, and Kanade [127] [126] [128]. A multilayer neural network is used to learn the face and nonface patterns from face/nonface images (i.e., the intensities and spatial relationships of pixels) whereas Sung and Poggio [152] used a neural network to find a discriminant function to classify face and nonface patterns using distance measures. They also used multiple neural networks and several arbitration methods to improve performance while Burel and Carel [12] used a single network, and Vaillant, Monrocq, and Le Cun [164] used two networks for classification. 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). As shown in Figure 10, 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. Given a test pattern, the output of the trained neural network indicates the evidence for a nonface (close to -1) or face pattern (close to 1). To detect faces anywhere in an image, the neural network is applied at all image locations. To detect faces larger than 20×20 pixels, the input image is repeatedly subsampled, and the network is applied at each scale. Nearly 1,050 face samples of various sizes, orientations, positions and intensities are used to train the network. In each training image, the eyes, tip of the nose, corners, and center of the mouth are labeled manually and used to normalize the face to the same scale, orientation, and position. The second component of this method is to merge overlapping detection and arbitrate between the outputs of multiple networks. Simple arbitration schemes such as logic operators (AND/OR) and voting are used to improve performance. Rowley, Baluja, and Kanade [127] reported several systems with different arbitration schemes that are less computationally expensive than Sung and Poggio's system, and have higher detection rates based on a test set of 24 images containing 144 faces.

One limitation of the methods by Rowley [127] and by Sung [152] is that they can only detect upright,

frontal faces. Recently, Rowley et al. [129] extended this method to detect rotated faces using a router network which 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. However, the new system has a lower detection rate on upright faces than 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.

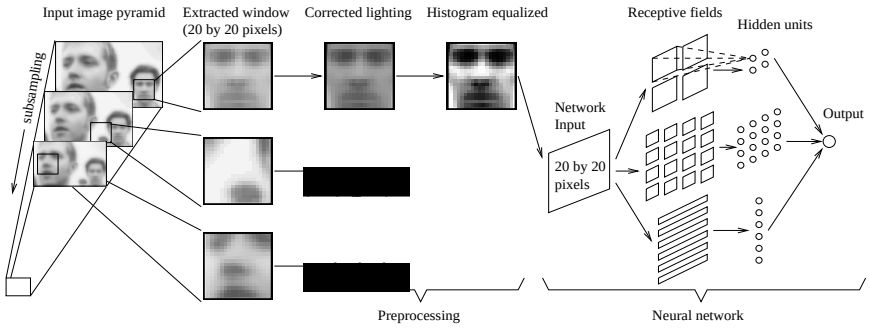


Figure 10: System diagram of Rowley's method [128]. Each face is pre-processed before feeding it to an ensemble of neural networks. Several arbitration methods are used to determine whether a face exists based on the output of these networks (Courtesy of H. Rowley, S. Baluja, and T. Kanade).

2.4.4 Support Vector Machines

Support Vector Machines (SVMs) were first applied to face detection by Osuna, Freud, and Girosi [107]. SVMs can be considered as a new paradigm to train polynomial function, neural networks, or radial basis function (RBF) classifiers. While most methods for training a classifier (e.g., Bayesian, neural networks, and RBF) are based on of minimizing the training error, i.e., *empirical risk*, SVMs operates on another induction principle, called *structural risk minimization*, which aims to minimize an upper bound on the expected generalization error. An SVM classifier is a linear classifier where the separating hyperplane is chosen to minimize the expected classification error of the unseen test patterns. This optimal hyperplane is defined by a weighted combination of a small subset of the training vectors, called support vectors. Estimating the optimal hyperplane is equivalent to solving a linearly constrained quadratic programming problem. However, the computation is both time and memory intensive. In [107] Osuna, Freund and Girosi developed an efficient method to train an SVM for large scale problems, and applied it to face detection. Based on two test sets of 10,000,000 test patterns of 19×19 pixels, their system has slightly lower error rates and runs approximately 30 times faster than the system by Sung and Poggio [153]. SVMs have also been used to detect faces and pedestrians in the wavelet domain [106] [108] [109].

2.4.5 Sparse Network of Winnows

Yang, Roth, and Ahuja proposed a method that uses SNoW learning architecture [125] [16] to detect faces with different features and expressions, in different poses, and under different lighting conditions [176]. They also studied the effect of learning with primitive as well as with multi-scale features. SNoW (Sparse Network of Winnows) is a sparse network of linear functions that utilizes the Winnow update rule [92]. It is specifically tailored for learning in domains in which the potential number of features taking part in decisions is very large, but may be unknown a priori. Some of the characteristics of this learning architecture are its sparsely connected units, the allocation of features and links in a data driven way, the decision mechanism and the utilization of an efficient update rule. In training the SNoW-based face detector, 1,681 face images from Olivetti [136], UMIST [56], Harvard [57], Yale [7] and FERET [115] databases are used to capture the variations in face patterns. To compare with other methods, they report results with two readily available data sets which contain 225 images with 619 faces [128]. With an error rate of 5.9%, this technique performs as well as other methods evaluated on the data set 1 in [128], including those using neural networks [128],

Kullback relative information [24], naive Bayes classifier [140] and support vector machines [107], while being computationally more efficient. See Table 4 for performance comparisons with other face detection methods.

2.4.6 Naive Bayes Classifier

In contrast to the methods in [107] [128] [154] which model the global appearance of a face, Schneiderman and Kanade described a naive Bayes classifier to estimate the joint probability of local appearance and position of face patterns (subregions of the face) at multiple resolutions [140]. They emphasize local appearance because some local patterns of an object are more unique than others; the intensity patterns around the eyes are much more distinctive than the pattern found around the cheeks. There are two reasons for using a naive Bayes classifier (i.e., no statistical dependency between the subregions). First, it provides better estimation of the conditional density functions of these subregions. Second, a naive Bayes classifier provides a functional form of the posterior probability to capture the joint statistics of local appearance and position on the object. At each scale, a face image is decomposed into four rectangular subregions. These subregions are then projected to a lower dimensional space using PCA and quantized into a finite set of patterns, and the statistics of each projected subregion are estimated from the projected samples to encode local appearance. Under this formulation, their method decides that a face is present when the likelihood ratio is larger than the ratio of prior probabilities. With an error rate of 93.0% on data set 1 in [128], the proposed Bayesian approach shows comparable performance to [128] and is able to detect some rotated and profile faces. Schneiderman and Kanade later extend this method with wavelet representations to detect profile faces and cars [141].

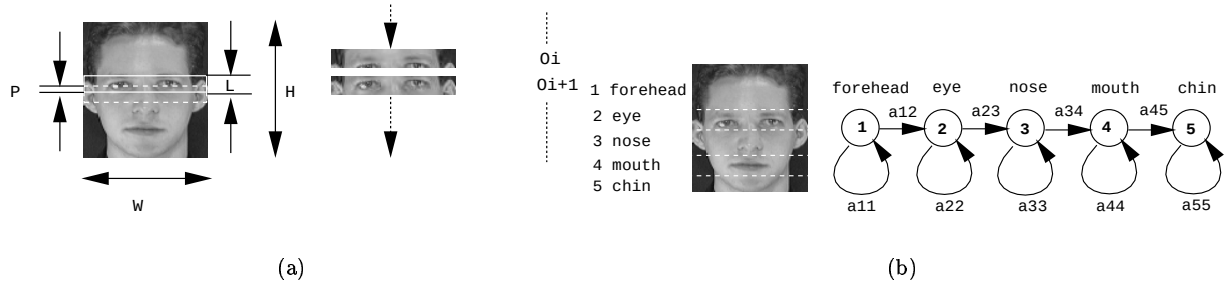
A related method using joint statistical models of local features was developed by Rickert, Jones, and Viola [124]. Local features are extracted by applying multiscale and multiresolution filters to the input image. The distribution of the features vectors (i.e., filter responses) is estimated by clustering the data and then forming a mixture of Gaussians. After the model is learned and further refined, test images are classified by computing the likelihood of their feature vectors with respect to the model. Their experimental results on face and car detection show interesting and good results.

2.4.7 Hidden Markov Model

The underlying assumption of the Hidden Markov Model (HMM) is that patterns can be characterized as a parametric random process, and that the parameters of this process can be estimated in a precise, well-defined manner. In developing an HMM for a pattern recognition problem, a number of hidden states need to be decided first to form a model. Then, one can train HMM to learn the transitional probability between states from the examples where each example is represented as a sequence of observations. The goal of training an HMM is to maximize the probability of observing the training data by adjusting the parameters in an HMM model with the standard Viterbi segmentation method and Baum-Welch algorithms [122]. After the HMM has been trained, the output probability of an observation determines the class to which it belongs.

Intuitively a face pattern can be divided into several regions such as the forehead, eyes, nose, mouth, and chin. A face pattern can then be recognized by a process in which these regions are observed in an appropriate order (from top to bottom and left to right). Instead of relying on accurate alignment as in template matching or appearance-based methods (where facial features such as eyes and noses need to be aligned well with respect to a reference point), this approach aims to associate facial regions with the states of a continuous density Hidden Markov Model. HMM-based methods usually treat a face pattern as a sequence of observation vectors where each vector is a strip of pixels as shown in Figure 11(a). During training and testing, an image is scanned in some order (usually from top to bottom) and an observation is taken as a block of pixels as shown in Figure 11(a). For face patterns, the boundaries between strips of pixels are represented by probabilistic transitions between states as shown in Figure 11(b), and the image data within a region is modeled by a multivariate Gaussian distribution. An observation sequence consists of all intensity values from each block. The output states correspond to the classes to which the observations belong. After the HMM has been trained, the output probability of an observation determines the class to which it belongs. HMMs have been applied to both face recognition and localization. Samaria [136] showed that the states of the HMM he trained corresponds to facial regions as shown in Figure 11(b). In other words, one state is responsible for characterizing the observation vectors of human foreheads, and another state is responsible for characterizing the observation vectors of human eyes. For face localization, an HMM

is trained for a generic model of human faces from a large collection of face images. If the face likelihood obtained for each rectangular pattern in the image is above a threshold, a face is located.



(a) (b)

Figure 11: Hidden Markov model for face localization. (a) Observation vectors: To train an HMM, each face sample is converted to a sequence of observation vectors. Observation vectors are constructed from a window of $W \times L$ pixels. By scanning the window vertically with P pixels of overlap, an observation sequence is constructed. (b) Hidden states: When an HMM with five states is trained with sequences of observation vectors, the boundaries between states are shown in (b) [136].

Samaria and Young applied 1-D and pseudo 2-D HMMs to facial feature extraction and face recognition [135] [136]. Their HMMs exploit the structure of a face to enforce constraints on the state transitions. Since significant facial regions such as hair, forehead, eyes, nose, and mouth occur in the natural order from top to bottom, each of these regions is assigned to a state in a one-dimensional continuous HMM. Figure 11(b) shows these five hidden states. For training, each image is uniformly segmented, from top to bottom into five states (i.e., each image is divided into five nonoverlapping regions of equal size). The uniform segmentation is then replaced by the Viterbi segmentation, and the parameters in the HMM are re-estimated using the Baum-Welch algorithm. As shown in Figure 11(a), each face image of width W and height H is divided into overlapping blocks of height L and width W. There are P rows of overlap between consecutive blocks in the vertical direction. These blocks form an observation sequence for the image, and the trained HMM is used to determine the output state. Similar to [135], Nefian and Hayes applied HMMs and Karhunen Loève Transform (KLT) to face localization and recognition [104]. Instead of using raw intensity values, the observation vectors consist of the (KLT) coefficients computed from the input vectors. Their experimental results on face recognition show a better recognition rate than [135]. On the MIT database which contains 432 images each with a single face, this pseudo 2-D HMM system has a success rate of 90%.

Rajagopalan et al. proposed two probabilistic methods for face detection [123]. In contrast to [154] which uses a set of multivariate Gaussians to model the distribution of face patterns, the first method in [123] uses higher order statistics (HOS) for density estimation. Similar to [154], both the unknown distributions of faces and nonfaces are clustered using six density functions based on higher order statistics of the patterns. As in [152], a multilayer perceptron is used for classification, and the input vector consists of twelve distance measures (i.e., log probability) between the image pattern and the twelve model clusters. The second method in [123] uses an HMM to learn the face to nonface and nonface to face transitions in an image. This approach is based on generating an observation sequence from the image and learning the HMM parameters corresponding to this sequence. The observation sequence to be learned is first generated by computing the distance of the subimage to the centers of the twelve face and nonface cluster centers estimated in the first method. After the learning completes, the optimal state sequence is further processed for binary classification. Experimental results show that both HOS and HMM methods have a higher detection rate than [128] [154], but with more false alarms.

2.4.8 Information-Theoretical Approach

The spatial property of face pattern can be modeled through different aspects. The contextual constraint, among others, is a powerful one and has often been applied to texture segmentation. The contextual

constraints in a face pattern are usually specified by a small neighborhood of pixels. Markov random field (MRF) theory provides a convenient and consistent way to model context-dependent entities such as image pixels and correlated features. This is achieved by characterizing mutual influences among such entities using conditional MRF distributions. According to the Hammersley-Clifford theorem, an MRF can be equivalently characterized by a Gibbs distribution and the parameters are usually maximum *a posteriori* (MAP) estimates [119]. Alternatively, the face and nonface distributions can be estimated using histograms. Using Kullback relative information, the Markov process that maximizes the information-based discrimination between the two classes can be found and applied to detection [89] [24].

Lew applied Kullback relative information [26] to face detection by associating a probability function $p(\mathbf{x})$ to the event that the template is a face and $q(\mathbf{x})$ to the event that the template is not a face [89]. A face training database consisting of 9 views of 100 individuals is used to estimate the face distribution. The nonface probability density function is estimated from a set of 143,000 nonface templates using histograms. From the training sets, the most informative pixels (MIP) are selected to maximize the Kullback relative information between $p(\mathbf{x})$ and $q(\mathbf{x})$ (i.e., to give the maximum class separation). It turns out the MIP distribution focuses on the eye and mouth regions and avoids the nose area. The MIP are then used to obtain linear features for classification and representation using the method of Fukunaga and Koontz [47]. To detect faces, a window is passed over the input image, and the distance from face space (DFFS) as defined in [114] is calculated. If the DFFS to the face subspace is lower than the distance to the nonface subspace, a face is assumed to exist within the window.

Kullback relative information is also employed by Colmenarez and Huang to maximize the information-based discrimination between positive and negative examples of faces [24]. Images from the training set of each class (i.e., face and nonface class) are analyzed as observations of a random process, and is 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.

Qian and Huang [119] presented a method that employs the strategies of both view-based and model-based methods. First, a visual attention algorithm, which uses high level domain knowledge, is applied to reduce the search space. This is achieved by selecting image areas in which targets may appear based on the region maps generated by a region detection algorithm (water-shed method). Within the selected regions, faces are detected with a combination of template matching and feature matching methods using a hierarchical Markov random field and maximum *a posteriori* estimation.

2.4.9 Inductive Learning

Inductive learning algorithms have also been applied to locate and detect faces. Huang, Gutta and Wechsler applied Quinlan's C4.5 algorithm [121] to learn a decision tree from positive and negative examples of face patterns [64]. Each training example is an 8×8 pixel window and is represented by a vector of 30 attributes which is composed of entropy, mean and standard deviation of the pixel intensity values. From these examples, C4.5 builds a classifier as a decision tree whose leaves indicate class identity and whose nodes specify tests to perform on a single attribute. The learned decision tree is then used to decide whether a face exists in the input example. The experiments show a localization accuracy rate of 96% on a set of 2,340 frontal face images in the FERET data set.

Duta and Jain [38] presented a method to learn the face concept using Mitchell's Find-S algorithm [101]. Similar to [154], they conjecture that the distribution of face patterns $p(\mathbf{x}|face)$ can be approximated by a set of Gaussian clusters, and that the distance from a face instance to one of the cluster centroids should be smaller than a fraction of the maximum distance from the points in that cluster to its centroid. The Find-S algorithm is then applied to learn the thresholding distance such that faces and nonfaces can be differentiated. This method has several distinct characteristics. First, it does not use negative (nonface) examples while [154] [128] use both positive and negative examples. Second, only the central portion of a face is used for training. Third, feature vectors consist of images with 32 intensity levels or textures while [154] uses full-scale intensity values as inputs This method achieves a detection rate of 90% on the first CMU data set.

2.5 Discussion

We have reviewed and classified face detection methods into four major categories. However, some methods can be classified into more than one category. For example, template matching methods usually use a face model and subtemplates to extract facial features [132] [27] [180] [143] [51], and then use these features to locate or detect faces. Furthermore, the boundary between knowledge-based methods and some template matching methods is blurry since the latter usually implicitly applies human knowledge to define the face templates [132] [28] [143]. On the other hand, face detection methods can also be categorized otherwise. For example, these methods can be classified based on whether they rely on local features [87] [140] [124] or treat a face pattern as whole (i.e., holistic) [154] [128]. Nevertheless, we think the four major classes categorize most methods sufficiently and appropriately.

3 Face Image Databases and Performance Evaluation

Most face detection methods require a training data set of face images, and the databases originally developed for face recognition experiments can be used as training sets for face detection. Since these databases were constructed to empirically evaluate recognition algorithms in certain domains, we first review the characteristics of these databases and their applicability to face detection. Although numerous face detection algorithms have been developed, most of them have not been tested on data sets with a large number of images. Furthermore, most experimental results are reported using different test sets. In order to fairly compare methods, a few benchmark data sets have recently been compiled. We review these benchmark data sets and discuss their characteristics. There are still a few issues that need to be carefully considered in performance evaluation even when the methods use the same test set. One issue is that researchers have different interpretations of what a "successful detection" is. Another issue is that different training sets are used, particularly for appearance based methods. We conclude this section with a discussion of these issues.

3.1 Face Image Database

Although many face detection methods have been proposed, less attention has been paid to the development of an image database for face detection research. The FERET database consists of monochrome images taken in different frontal views, and in left and right profiles [115]. Only the upper torso of an individual (mostly head and necks) appears in an image on a uniform and uncluttered background. The FERET database has been used to assess the strengthens and weaknesses of different face recognition approaches [115]. Since each image consists of an individual on a uniform and uncluttered background, it is not suitable for face detection benchmarking. This is similar to many databases that were created for the development and testing of face recognition algorithms. Turk and Pentland created a face database of 16 people [163] (available at ftp://whitechapel.media.mit.edu/pub/images/). The images are taken in frontal view with slight variability in head orientation (tilted upright, right and left) on a cluttered background. The face database from AT&T Cambridge Laboratories (formerly known as the Olivetti database) consists of ten different images for forty distinct subjects. (available at http://www.uk.research.att.com/facedatabase.html) [136] . The images were taken at different times, varying the lighting, facial expressions and facial details (glasses). The Harvard database consists of cropped, masked frontal face images taken from a wide variety of light sources [57]. It was used by Hallinan for a study on face recognition under the effect of varying illumination conditions. With sixteen individuals, the Yale face database (available at http://cvc.yale.edu/) contains ten frontal images per person, each with different facial expressions, with and without glasses, and under different lighting conditions [7]. The M2VTS multimodal database from the European ACTS projects was developed for access control experiments using multimodal inputs [116]. It contains sequences of face images of 37 people. The five sequences for each subject were taken over one week. Each image sequence contains images from right profile (-90 degree) to left profile (90 degree) while the subject counts from '0' to '9' in their native languages. The UMIST database consists of 564 images of 20 people with varying pose. The images of each subject cover a range of poses from right profile to frontal views [56]. The Purdue AR database contains over 3,276 color images of 126 people (70 males and 56 females) in frontal view [96]. This database is designed for face recognition experiments under several mixing factors such as facial expressions,

illumination conditions and occlusions. All the faces appear with different facial expression (neutral, smile, anger, and scream), illumination (left light source, right light source and sources from both sides), and occlusion (wearing sunglasses or scarf). The images were taken during two sessions separated by two weeks. All the images were taken by the same camera setup under tightly controlled conditions of illumination and pose. This face database has been applied to image and video indexing as well as retrieval [96]. Table 2 summarizes the characteristics of the abovementioned face image databases.

Table 2: Face image database

Data Set	Location	Description
MIT Database [163]	ftp://whitechapel.media.mit.edu/pub/images/	Faces of 16 people, 27 of each person under various illumination conditions, scale and head orientation.
FERET Database [115]	http://www.nist.gov/humanid/feret	A large collection of male and female faces. Each image contains a single person with certain expression.
UMIST Database [56]	http://images.ee.umist.ac.uk/danny/database.html	564 images of 20 subjects. Each subject covers a range of poses from profile to frontal views.
University of Bern Database	ftp://iamftp.unibe.ch/pub/Images/FaceImages/	300 frontal face images of 30 people (10 images per person) and 150 profile face images (5 images per person).
Yale Database [7]	http://cvc.yale.edu	Face images with expressions, glasses under different illumination conditions.
AT&T (Olivetti) Database [136]	http://www.uk.research.att.com	40 subjects, 10 images per subject.
Harvard Database [57]	ftp://ftp.hrl.harvard.edu/pub/faces/	Cropped, masked face images under a wide range of lighting conditions.
M2VTS Database [116]	http://poseidon.csd.auth.gr/M2VTS/index.html	A multimodal database containing various image sequences.
Purdue AR Database [96]	http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html	3,276 face images with different facial expressions and occlusions under different illuminations.

3.2 Benchmark Test Sets for Face Detection

The abovementioned databases are designed mainly to measure performance of face recognition methods, and thus each image contains only one individual. Therefore, such databases can be best utilized as training sets rather than test sets. The tacit reason for comparing classifiers on test sets is that these data sets represent problems that systems might face in the real world, and that superior performance on these benchmarks may translate to superior performance on other real-world tasks. Toward this end, researchers have compiled a wide collection of data sets from a wide variety of images. Sung and Poggio created two databases for face detection [152] [154]. The first set consists of 301 frontal and near-frontal mugshots of 71 different people. These images are high quality digitized images with a fair amount of lighting variation. The second set consists of 23 images with a total of 149 face patterns. Most of these images have complex background with faces taking up only a small amount of the total image area. The most widely-used face detection database has been created by Rowley, Baluja, and Kanade [127] [130] (available at http://www.cs.cmu.edu/~har/faces.html). It consists of 130 images with a total of 507 frontal faces. This data set includes 23 images of the second data set used by Sung and Poggio [154]. Most images contain more than one face on a cluttered background, and so this is a good test set to assess algorithms which detect upright frontal faces. Figure 12 shows some images in the data set collected by Sung and Poggio [154], and Figure 13 shows images from the data set collected by Rowley, Baluja, and Kanade [128].

Figure 12: Sample images in Sung's data set [154]. Some images are scanned from newspapers and thus have low resolution. Though most faces in the images are upright and frontal. Some faces in the images appear in different pose.

Figure 13: Sample images in Rowley's data set [128]. Some images contain hand-drawn cartoon faces. Most images contain more than one faces, and the face size varies significantly.

Rowley, Baluja, and Kanade also compiled another database of images for detecting 2-D faces with frontal pose and rotation in image plane [129]. It contains 50 images with a total of 223 faces, of which 210 are at angles of more than 10 degrees. Figure 14 shows some rotated images in this data set. To measure the performance of detection methods on faces with profile views, Schneiderman and Kanade gathered a set of 208 images where each image contains faces with facial expressions and in profile views [141]. Figure 15 shows some images in the test set.

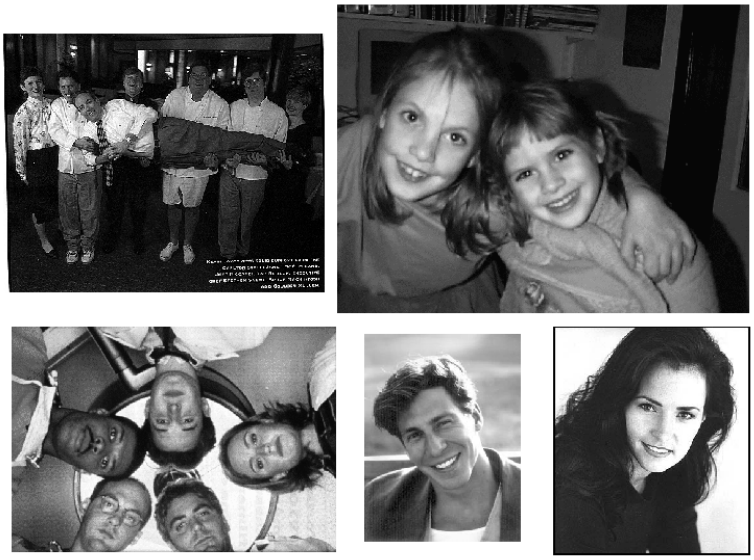


Figure 14: Sample images of Rowley's data set [129] which contains images with in-plane rotated faces against complex background.

Recently, Kodak compiled an image database as a common test bed for direct benchmarking of face detection and recognition algorithms [94]. Their database has 300 digital photos that are captured in a variety of resolutions and face size ranges from as small as 13×13 pixels to as large as 300×300 pixels. Table 3 summarizes the characteristics of the abovementioned test sets for face detection.

Table 3: Test sets for face detection

Data Set	Location	Description
MIT Test Set [154]	http://www.cs.cmu.edu/~har	Two sets of high and low resolution gray scale images with multiple faces in complex background.
CMU Test Set [128]	http://www.cs.cmu.edu/~har	130 gray scale images with a total of 507 frontal 507 frontal faces.
CMU Profile Face Test Set [141]	ftp://eyes.ius.cs.cmu.edu/usr20/ ftp/testing_face_images.tar.gz	208 gray scale images with faces in profile views.
Kodak Data Set [94]	Eastman Kodak Corporation	Faces of multiple size, pose and under varying illumination in color images. Designed for face detection and recognition.

3.3 Performance Evaluation

In order to obtain a fair empirical evaluation of face detection methods, it is important to use a standard and representative test set for experiments. Although many face detection methods have been developed

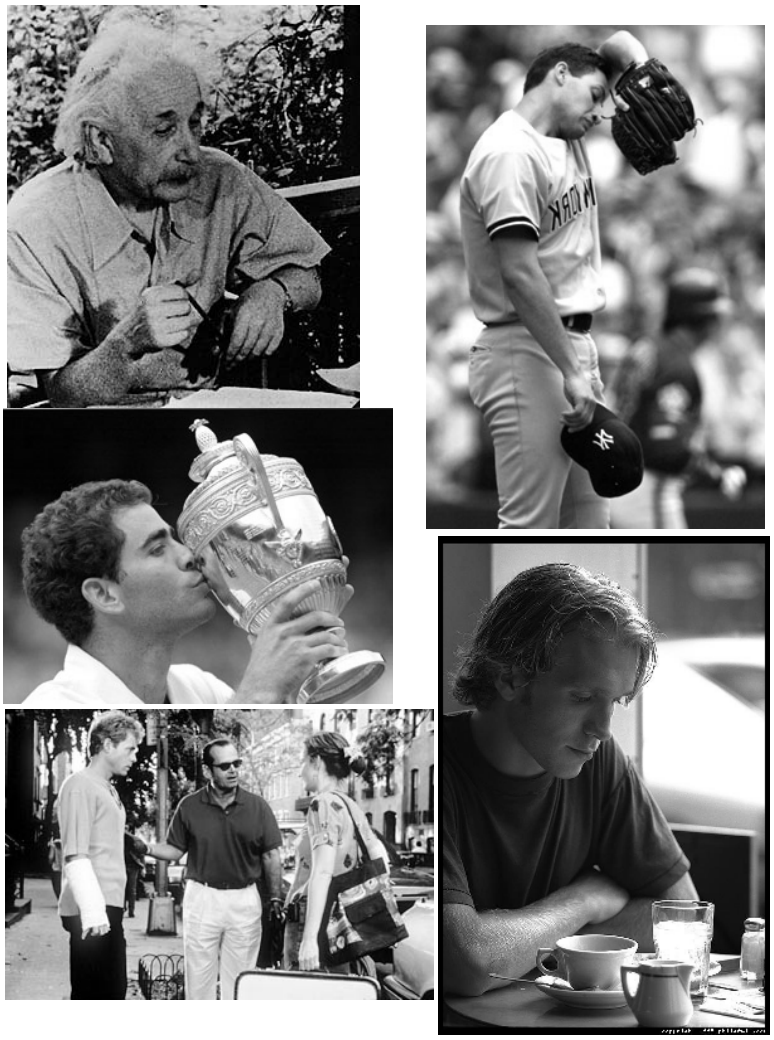


Figure 15: Sample images of profile faces from Schneiderman's data set [141]. This data set contains images with faces in profile views, and some with facial expressions.

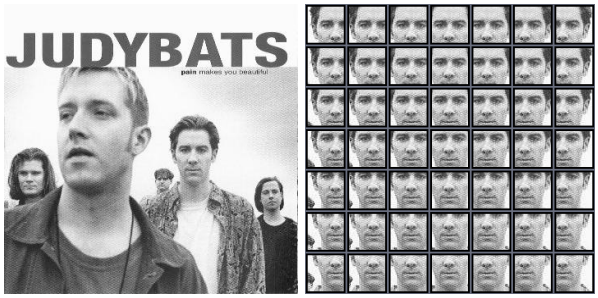
over the past decade, only a few of them have been tested on the same data set. Table 4 summarizes the reported performance among several appearance based face detection methods on two standard data sets described in the previous section.

Table 4: Experimental results on images from test set 1 (125 images with 483 faces) and test set 2 (23 images with 136 faces) (see text for details)

Method	Test Set 1		Test Set 2	
	Detection Rate	False Detections	Detection Rate	False Detections
Distribution based [154]	N/A	N/A	81.9%	13
Neural network [128]	92.5%	862	90.3%	42
Naive Bayes classifier [140]	93.0%	88	91.2%	12
Kullback relative information [24]	98.0%	12758	N/A	N/A
Support vector machine [107]	N/A	N/A	74.2%	20
Mixture of factor analyzers [175]	92.3%	82	89.4%	3
Fisher linear discriminant [175]	93.6%	74	91.5%	1
SNoW with primitive features [176]	94.2%	84	93.6%	3
SNoW with multi-scale features [176]	94.8%	78	94.1%	3
Inductive learning [38]	90%	N/A	N/A	N/A

Although Table 4 shows the performance of these methods on the same test set, such an evaluation may not characterize how well these methods will compare in the field. There are a few factors that complicate the assessment of these appearance based methods. First, the reported results are based on different training sets and different tuning parameters. The number and variety of training examples have a direct effect on the classification performance. However this factor is often ignored in performance evaluation, which is an appropriate criteria if the goal is to evaluate the systems rather than the learning methods. The second factor is the training time and execution time. Although the training time is usually ignored by most systems, it may be important for real-time applications that require on-line training on different data sets. Third, the number of scanning windows in these methods vary because they are designed to operate in different environments (i.e., to detect faces within a size range). For example, Colmenarez and Huang argued that their method scans more windows than others and thus the number of false detections is higher than others [24]. Furthermore, the criteria adopted in reporting the detection rates is usually not clearly described in most systems. Figure 16(a) shows a test image and Figure 16(b) shows some subimages to be classified as a face or non-face. Suppose that all the subimages in Figure 16(b) are classified as face patterns, some criteria may consider all of them as "successful" detections. However, a more strict criterion (e.g., each successful detection must contain all the visible eyes and mouths in an image) may classify most of them as false alarms. It is clear that a uniform criteria should be adopted to assess different classifiers. In [128] Rowley, Baluja, and Kanade adjust the criteria until the experimental results match their intuition of what a correct detection is, i.e., the square window should contain the eyes and also the mouth. The criteria they eventually use is that the center of the detected bounding box must be within 4 pixels and the scale must he within a factor of 1.2 (their scale step size) of ground truth (recorded manually).

Finally, the evaluation criteria may and should depend on the purpose of the detector. If the detector is going to be used to count people, then the sum of false positives and false negatives is appropriate. On the other hand, if the detector is to be used to verify that an individual is who he/she claims to be (validation), then it may be acceptable for the face detector to have additional false detections since it is unlikely that these false detections will be acceptable images of the individual, i.e., the validation process will reject the false detections. In other words, the penalty or cost of one type of error should be properly weighted such that one can build an optimal classifier using Bayes decision rule (See Sections 2.2-2.4 in [36]). This argument is supported by a recent study which points out accuracy of classifier (i.e., detection rate in face detection) is not an appropriate goal for many of the real world task [118]. One reason is that classification accuracy assumes equal misclassification costs. This assumption is problematic because for most real world problems one type of classification error is much more expensive than another. In some face detection applications,



(a) Test image (b) Detection results

Figure 16: Different criteria lead to different detection results. Suppose all the subimages in (b) are classified as face patterns by a classifier. A loose criterion may declare all the faces as "successful" detections while a more strict one would declare most of them as nonfaces.

it is important that all the existing faces are detected. Another reason is accuracy maximization assumes that the class distribution is known for the target environment. In other words, we assume the test data sets represent the "true" working environment for the face detectors. However, this assumption is rarely justified.

When detection methods are used within real systems, it is important to consider what computational resources are required, particularly time and memory. Accuracy may need to be sacrificed for for speed.

The scope of the considered techniques in evaluation is also important. In this survey, we discuss at least four different forms of the face detection problem: (1) Localization in which there is a single face and the goal is provide a suitable estimate of position, scale to be used as input for face recognition. (2) In a cluttered monochrome scene, detect all faces. (3) In color images, detect (localize) all faces. (4) In a video sequence, detect and localize all faces. An evaluation protocol should carefully designed in assessing these different detection situations.

It should be noted that there is a potential risk of using a universal though modest sized standard test set. As researchers develop new methods or "tweak" existing ones to get better performance on the test set, they engage in a subtle form of the unacceptable practice of "testing on the training set." As a consequence, the latest methods may perform better against this hypothetical test set but not actually perform better in practice. This can be obviated by having a sufficiently large and representative universal test set. Alternatively, methods could be evaluated on a smaller test set if that test set is randomly chosen (generated) each time the method is evaluated.

In summary, fair and effective performance evaluation requires careful design of protocols, scope, and data sets. Such issues have attracted much attention in numerous vision problems [21] [60] [142] [115]. However, performing this evaluation or trying to declare a "winner" is beyond the scope of this survey. Instead, we hope that either a consortium of researchers engaged in face detection or a third party will take on this task. Until then, we hope that when applicable, researchers will report the result of their methods on the publicly available data sets described here. As a first step toward this goal, we have collected sample face detection codes and evaluation tools at http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html.

4 Discussion and Conclusion

This paper attempts to provide a comprehensive survey of research on face detection, and to provide some structural categories for the methods described in over 150 papers. When appropriate, we have reported on the relative performance of methods. But in so doing, we are cognizant that there is a lack of uniformity in how methods are evaluated, and so it is imprudent to explicitly declare which methods indeed have the lowest error rates. Instead, we urge members of the community to develop and share test sets and to

report results on already available test sets. We also feel the community needs to more seriously consider systematic performance evaluation: this would allow users of the face detection algorithms to know which ones are competitive in which domains. It will also spur researchers to produce truly more effective face detection algorithms.

Although significant progress has been made in the last two decades, there is still work to be done, and we believe that a robust face detection system should be effective under full variation in:

- Lighting conditions
- Orientation, pose, and partial occlusion
- Facial expression
- Presence of glasses, facial hair, variety of hair styles

Face detection is a challenging and interesting problem in and of itself. However, it can also be seen as a one of the few attempts at solving one of the grand challenges of computer vision, the recognition of object classes. The class of faces admits a great deal of shape, color, and albedo variability due to differences in individuals, non-rigidity, facial hair, glasses, and makeup. Images are formed under variable lighting and 3-D pose, and may have cluttered backgrounds. Hence, face detection research confronts the full range of challenges found in general purpose, object class recognition. However, the class of faces also has very apparent regularities that are exploited by many heuristic or model-based methods or are readily "learned" in data-driven methods. One expects some regularities when defining classes in general, but they may not be so apparent. Finally, though faces have tremendous within-class variability, face detection remains a two class recognition problem (face vs. nonface).

References

[1] T. Agui, Y. Kokubo, H. Nagashashi, and T. Nagao. Extraction of face recognition from monochromatic photographs using neural networks. In *Proceedings of the Second International Conference on Automation, Robotics and Computer Vision*, volume 1, pages CV–18.8.1–CV–18.8.5, 1992.

[2] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):1211–1235, 1996.

[3] Y. Amit, D. Geman, and B. Jedynak. Efficient focusing and face detection. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, editors, *Face Recognition: From Theory to Applications*, volume 163 of *NATO ASI Series F, Computer and Systems Sciences*, pages 124–156. Springer, 1998.

[4] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1300–1305, 1997.

[5] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis.* John Wiley, New York, 1984.

[6] M. F. Augusteijn and T. L. Skujca. Identification of human faces through texture-based feature recognition and neural network technology. In *Proceedings of IEEE Conference on Neural Networks*, pages 392–398, 1993.

[7] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[8] O. Bernier, M. Collobert, R. Féraud, V. Lemarie, J. E. Viallet, and D. Collobert. MULTRAK: A system for automatic multiperson localization and tracking in real-time. In *Proceedings of IEEE International Conference on Image Processing*, pages 136–140, 1998.

[9] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[10] C. Breazeal and B. Scassellati. A context-dependent attention system for a social robot. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, volume 2, pages 1146–1151, 1999.

[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees.* Wadsworth, 1984.

[12] G. Burel and D. Carel. Detection and localization of faces on digital images. *Pattern Recognition Letters*, 15(10):963–967, 1994.

[13] M. C. Burl, T. K. Leung, and P. Perona. Face localization via shape statistics. In *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, pages 154–159, 1995.

[14] J. Cai, A. Goshtasby, and C. Yu. Detecting human faces in color images. In *Proceedings of the 1998 International Workshop on Multi-Media Database Management Systems*, pages 124–131, 1998.

[15] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[16] A. Carleson, C. Cumby, J. Rosen, and D. Roth. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, 1999.

[17] D. Chai and K. N. Ngan. Locating facial region of a head-and-shoulders color image. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 124–129, 1998.

[18] R. Chellappa, C. L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–740, 1995.

[19] Q. Chen, H. Wu, and M. Yachida. Face detection by fuzzy matching. In *Proceedings of the Fifth IEEE International Conference on Computer Vision*, pages 591–596, 1995.

[20] D. Chetverikov and A. Lerch. Multiresolution face detection. In *Theoretical Foundations of Computer Vision*, volume 69 of *Mathematical Research*, pages 131–140. Akademie Verlg, 1993.

[21] K. Cho, P. Meer, and J. Cabrera. Performance assessment through bootstrap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1185–1198, 1997.

[22] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proceedings of the Third IEEE International Conference on Computer Vision*, pages 616–623, 1990.

[23] M. Collobert, R. Féraud, G. L. Tourneur, O. Bernier, J. E. Viallet, Y. Mahieux, and D. Collobert. LISTEN: A system for locating and tracking individual speakers. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 283–288, 1996.

[24] 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*, pages 782–787, 1997.

[25] T. F. Cootes and C. J. Taylor. Locating faces using statistical feature detectors. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 204–209, 1996.

[26] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley Interscience, 1991.

[27] I. Craw, H. Ellis, and J. Lishman. Automatic extraction of face features. *Pattern Recognition Letters*, 5:183–187, 1987.

[28] I. Craw, D. Tock, and A. Bennett. Finding face features. In *Proceedings of the Second European Conference on Computer Vision*, pages 92–96, 1992.

[29] J. L. Crowley and J. M. Bedrune. Integration and control of reactive visual processes. In *Proceedings of the Third European Conference on Computer Vision*, volume 2, pages 47–58, 1994.

[30] J. L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 640–645, 1997.

[31] Y. Dai and Y. Nakano. Extraction for facial images from complex background using color information and SGLD matrices. In *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, pages 238–242, 1995.

[32] Y. Dai and Y. Nakano. Face-Texture model based on SGLD and its application in face detection in a color scene. *Pattern Recognition*, 29(6):1007–1017, 1996.

[33] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, 2000.

[34] A. Dempster. A generalization of Bayesian theory. *Journal of the Royal Statistical Society*, 30:205–247, 19978.

[35] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 2000.

[36] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley, New York, 1973.

[37] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Intersciance, New York, 2001.

[38] N. Duta and A. K. Jain. Learning the human face concept from black and white pictures. In *Proceedings of International Conference on Pattern Recognition*, pages 1365–1367, 1998.

[39] G. J. Edwards, C.J.Taylor, and T. Cootes. Learning to identify and track faces in image sequences. In *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pages 317–322, 1998.

[40] I. A. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *Proceedings of the Fifth IEEE International Conference on Computer Vision*, pages 360–367, 1995.

[41] S. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D. S. Touretsky, editor, *Advances in Neural Information Processing Systems 2*, pages 524–532, 1990.

[42] R. Féraud. PCA, neural networks and estimation for face detection. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, editors, *Face Recognition: From Theory to Applications*, volume 163 of *NATO ASI Series F, Computer and Systems Sciences*, pages 424–432. Springer, 1998.

[43] R. Féraud and O. Bernier. Ensemble and modular approaches for face detection: A comparison. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 472–478. MIT Press, 1998.

[44] R. Féraud, O. J. Bernier, J.-E. Villet, and M. Collobert. A fast and accuract face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):42–53, 2001.

[45] D. Forsyth. A novel approach to color constancy. *International Journal of Computer Vision*, 5(1):5–36, 1990.

[46] 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*, pages 32–37, 1998.

[47] F. Fukunaga and W. Koontz. Applications of the Karhunen-Loève expansion to feature selection and ordering. *IEEE Transactions on Computers*, 19(5):311–318, 1970.

[48] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic, New York, 1972.

[49] 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.

[50] R. C. Gonzalez and P. A. Wintz. *Digital Image Processing.* Addison Wesley, Reading, 1987.

[51] V. Govindaraju. Locating human faces in photographs. *International Journal of Computer Vision*, 19(2):129–146, 1996.

[52] V. Govindaraju, D. B. Sher, R. K. Srihari, and S. N. Srihari. Locating human faces in newspaper photographs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 549–554, 1989.

[53] V. Govindaraju, S. N. Srihari, and D. B. Sher. A computational model for face location. In *Proceedings of the Third IEEE International Conference on Computer Vision*, pages 718–721, 1990.

[54] H. P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. In *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, pages 41–46, 1995.

[55] H. P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan. Multimodal system for locating heads and faces. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 88–93, 1996.

[56] D. B. Graham and N. M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, editors, *Face Recognition: From Theory to Applications*, volume 163 of *NATO ASI Series F, Computer and Systems Sciences*, pages 446–456. Springer, 1998.

[57] P. Hallinan. *A Deformable Model for Face Recognition Under Arbitrary Lighting Conditions.* PhD thesis, Harvard University, 1995.

[58] C.-C. Han, H.-Y. M. Liao, K.-C. Yu, and L.-H. Chen. Fast face detection via morphology-based pre-processing. In *Proceedings of the Ninth International Conference on Image Analysis and Processing*, pages 469–476, 1998.

[59] R. M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

[60] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. A robust visual method for assessing the relative performance of edge detection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1338–1359, 1997.

[61] G. E. Hinton, P. Dayan, and M. Revow. Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8(1):65–74, 1997.

[62] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 498–520, 1933.

[63] K. Hotta, T. Kurita, and T. Mishima. Scale invariant face detection method using higher-order local autocorrelation features extracted from log-polar image. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 70–75, 1998.

[64] J. Huang, S. Gutta, and H. Wechsler. Detection of human faces using decision trees. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 248–252, 1996.

[65] D. Hutenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.

[66] T. S. Jebara and A. Pentland. Parameterized structure from motion fro 3D adaptive feedback tracking of faces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 144–150, 1997.

[67] T. S. Jebara, K. Russell, and A. Pentland. Mixtures of eigenfeatures for real-time structure from texture. In *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pages 128–135, 1998.

[68] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[69] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 274–280, 1999.

[70] P. Juell and R. Marsh. A hierarchical neural network for human face detection. *Pattern Recognition*, 29(5):781–787, 1996.

[71] T. Kanade. *Picture processing by computer complex and recognition of human faces*. PhD thesis, Kyoto University, 1973.

[72] K. Karhunen. Uber lineare methoden in der wahrscheinlichkeitsrechnung. *Annales Academiae Sciientiarum Fennicae, Series AI: Mathematica-Physica*, 37:3–79, 1946. (Translated: RAND Corp., Santa Monica, CA, Report T-131, August, 1960).

[73] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First IEEE International Conference on Computer Vision*, pages 259–269, 1987.

[74] R. Kauth, A. Pentland, and G. Thomas. Blob: An unsupervised clustering approach to spatial preprocessing of MSS imagery. In *Proceedings of the Eleventh International Symposium on Remote Sensing of the Environment*, pages 1309–1317, 1977.

[75] D. G. Kendall. Shape manifolds, procrustean metrics, and complex projective shapes. *Bulletins of the London Mathematical Society*, 16:81–121, 1984.

[76] C. Kervrann, F. Davoine, P. Perez, H. Li, R. Forchheimer, and C. Labit. Generalized likelihood ratio-based face detection and extraction of mouth features. In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication*, pages 27–34, 1997.

[77] S.-H. Kim, N.-K. Kim, S. C. Ahn, and H.-G. Kim. Object oriented face detection using range and color information. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 76–81, 1998.

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

[79] R. Kjeldsen and J. Kender. Finding skin in color images. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 312–317, 1996.

[80] T. Kohonen. *Self-Organization and Associative Memory*. Springer, 1989.

[81] C. Kotropoulos and I. Pitas. Rule-based face detection in frontal views. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2537–2540, 1997.

[82] C. Kotropoulos, A. Tefas, and I. Pitas. Frontal face authentication using variants of dynamic link matching based on mathematical morphology. In *Proceedings of IEEE International Conference on Image Processing*, pages 122–126, 1998.

[83] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[84] Y. H. Kwon and N. da Vitoria Lobo. Face detection using templates. In *Proceedings of International Conference on Pattern Recognition*, pages 764–767, 1994.

[85] K. Lam and H. Yan. Fast algorithm for locating head boundaries. *Journal of Electronic Imaging*, 3(4):351–359, 1994.

[86] A. Lanitis, C. J. Taylor, and T. F. Cootes. An automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401, 1995.

[87] T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proceedings of the Fifth IEEE International Conference on Computer Vision*, pages 637–644, 1995.

[88] T. K. Leung, M. C. Burl, and P. Perona. Probabilistic affine invariants for recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–684, 1998.

[89] M. S. Lew. Information theoretic view-based and modular face detection. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 198–203, 1996.

[90] F. Leymarie and M. D. Levine. Tracking deformable objects in the plan using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634, 1993.

[91] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Transactions on Neural Networks*, 8(1):114–132, 1997.

[92] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[93] M. M. Loève. *Probability Theory*. Van Nostrand, Princeton, 1955.

[94] A. C. Loui, C. N. Judice, and S. Liu. An image database for benchmarking of automatic face detection and recognition algorithms. In *Proceedings of IEEE International Conference on Image Processing*, pages 146–150, 1998.

[95] K. V. Mardia and I. L. Dryden. Shape distributions for landmark data. *Advanced Applied Probability*, 21:742–755, 1989.

[96] A. Martinez and R. Benavente. The AR face database. Technical Report CVC 24, Purdue University, 1998.

[97] A. Martinez and A. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[98] S. McKenna, S. Gong, and Y. Raja. Modelling facial colour and identity with Gaussian mixtures. *Pattern Recognition*, 31(12):1883–1892, 1998.

[99] S. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3/4):223–229, 1998.

[100] J. Miao, B. Yin, K. Wang, L. Shen, and X. Chen. A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template. *Pattern Recognition*, 32(7):1237–1248, 1999.

[101] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[102] Y. Miyake, H. Saitoh, H. Yaguchi, and N. Tsukada. Facial pattern detection and color correction from television picture for newspaper printing. *Journal of Imaging Technology*, 16(5):165–169, 1990.

[103] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[104] A. V. Nefian and M. H. H. III. Face detection and recognition using Hidden Markov Models. In *Proceedings of IEEE International Conference on Image Processing*, volume 1, pages 141–145, 1998.

[105] N. Oliver, A. Pentland, and F. Berard. LAFER: Lips and face real time tracker. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 123–129, 1997.

[106] 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*, pages 193–199, 1997.

[107] 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*, pages 130–136, 1997.

[108] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pages 555–562, 1998.

[109] C. Papageorgiou and T. Poggio. A trainable system for object recognition. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[110] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[111] A. Pentland. Looking at people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119, 2000.

[112] A. Pentland. Perceptual intelligence. *Communications of the ACM*, 43(3):35–44, 2000.

[113] A. Pentland and T. Choudhury. Face recognition for smart environments. *IEEE Computer*, pages 50–55, 2000.

[114] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proceedings of the Fourth IEEE International Conference on Computer Vision*, pages 84–91, 1994.

[115] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1034, 2000.

[116] S. Pigeon and L. Vandendrope. The M2VTS multimodal face database. In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication*, 1997.

[117] M. Propp and A. Samal. Artificial neural network architectures for human face detection. *Intelligent Engineering Systems Through Artificial Neural Networks*, 2, 1992.

[118] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[119] 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*, pages 186–192, 1997.

[120] R. J. Qian, M. I. Sezan, and K. E. Matthews. A robust real-time face tracking algorithm. In *Proceedings of IEEE International Conference on Image Processing*, pages 131–135, 1998.

[121] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Kluwer Academic, 1993.

[122] L. R. Rabiner and B.-H. Jung. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[123] 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*, pages 640–645, 1998.

[124] T. Rikert, M. Jones, and P. Viola. A cluster-based statistical model for object detection. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1046–1053, 1999.

[125] D. Roth. Learning to resolve natural language ambiguities: A unified approach. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 806–813, 1998.

[126] H. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 875–881, 1996.

[127] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 203–208, 1996.

[128] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[129] 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*, pages 38–44, 1998.

[130] H. A. Rowley. *Neural Network-Based Face Detection.* PhD thesis, Carnegie Mellon University, 1999.

[131] E. Saber and A. M. Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters*, 17(8):669–680, 1998.

[132] T. Sakai, M. Nagao, and S. Fujibayashi. Line extraction and pattern detection in a photograph. *Pattern Recognition*, 1:233–248, 1969.

[133] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, 1992.

[134] A. Samal and P. A. Iyengar. Human face detection using silhouettes. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(6):845–867, 1995.

[135] F. Samaria and S. Young. HMM based architecture for face identification. *Image and Vision Computing*, 12:537–583, 1994.

[136] F. S. Samaria. *Face Recognition Using Hidden Markov Models.* PhD thesis, University of Cambridge, 1994.

[137] S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and detecting faces in news videos. *IEEE Multimedia*, 6(1):22–35, 1999.

[138] D. Saxe and R. Foulds. Toward robust skin identification in video images. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 379–384, 1996.

[139] B. Scassellati. Eye finding via face detection for a foevated, active vision system. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, 1998.

[140] 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*, pages 45–51, 1998.

[141] 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*, volume 1, pages 746–751, 2000.

[142] J. A. Shufelt. Performance evaluation and analysis of monocular building extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):311–326, 1997.

[143] P. Sinha. Object recognition via image invariants: A case study. *Investigative Ophthalmology and Visual Science*, 35(4):1735–1740, 1994.

[144] P. Sinha. *Processing and recognizing 3D forms.* PhD thesis, Massachusetts Institute of Technology, 1995.

[145] S. A. Sirohey. Human face segmentation and identification. Technical Report CS-TR-3176, University of Maryland, 1993.

[146] J. Sobottka and I. Pitas. Segmentation and tracking of faces in color images. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 236–241, 1996.

[147] K. Sobottka and I. Pitas. Face localization and feature extraction based on shape and color information. In *Proceedings of IEEE International Conference on Image Processing*, pages 483–486, 1996.

[148] F. Soulie, E. Viennet, and B. Lamy. Multi-modular neural network architectures: Pattern recognition applications in optical character recognition and human face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):721–755, 1993.

[149] T. Starner and A. Pentland. Real-time ASL recognition from video using HMM's. Technical Report 375, Media Lab, MIT, 1996.

[150] Y. Sumi and Y. Ohta. Detection of face orientation and facial components using distributed appearance modeling. In *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, pages 254–259, 1995.

[151] Q. B. Sun, W. M. Huang, and J. K. Wu. Face detection based on color and local symmetry information. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 130–135, 1998.

[152] K.-K. Sung. *Learning and Example Selection for Object and Pattern Detection*. PhD thesis, Massachusetts Institute of Technology, 1996.

[153] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. Technical Report AI Memo 1521, MIT AI Lab, 1994.

[154] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[155] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[156] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):891–896, 1996.

[157] B. Takacs and H. Wechsler. Face location using a dynamic model of retinal feature extraction. In *Proceedings of the First International Workshop on Automatic Face and Gesture Recognition*, pages 243–247, 1995.

[158] A. Tefas, C. Kotropoulos, and I. Pitas. Variants of dynamic link architecture based on mathematical morphology for frontal face authentication. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 814–819, 1998.

[159] J. C. Terrillon, M. David, and S. Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and invariant moments. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 112–117, 1998.

[160] J. C. Terrillon, M. David, and S. Akamatsu. Detection of human faces in complex scene images by use of a skin color model and invariant Fourier-Mellin moments. In *Proceedings of International Conference on Pattern Recognition*, pages 1350–1355, 1998.

[161] A. Tsukamoto, C.-W. Lee, and S. Tsuji. Detection and tracking of human face with synthesized templates. In *Proceedings of the First Asian Conference on Computer Vision*, pages 183–186, 1993.

[162] A. Tsukamoto, C.-W. Lee, and S. Tsuji. Detection and pose estimation of human face with synthesized image models. In *Proceedings of International Conference on Pattern Recognition*, pages 754–757, 1994.

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

[164] R. Vaillant, C. Monrocq, and Y. Le Cun. An original approach for the localisation of objects in images. In *IEE Proceedings: Vision, Image and Signal Processing*, volume 141, pages 245–250, 1994.

[165] M. Venkatraman and V. Govindaraju. Zero crossings of a non-orthogonal wavelet transform for object location. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 57–60, 1995.

[166] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):328–339, 1989.

[167] H. Wang and S.-F. Chang. A highly efficient system for automatic face region detection in MPEG video. *IEEE Transaction on Circuits and Systems for Video Technology*, 7(4):615–628, 1997.

[168] H. Wu, Q. Chen, and M. Yachida. Face detection from color images using a fuzzy pattern matching method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):557–563, 1999.

[169] H. Wu, T. Yokoyama, D. Pramadihanto, and M. Yachida. Face and facial feature extraction from color image. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 345–350, 1996.

[170] G. Yang and T. S. Huang. Human face detection in complex background. *Pattern Recognition*, 27(1):53–63, 1994.

[171] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel. Visual tracking for multimodal human computer interaction. In *Proceedings of ACM Human Factors in Computing Systems Conference(CHI 98)*, pages 140–147, 1998.

[172] J. Yang and A. Waibel. A real-time face tracker. In *Proceedings of the Third Workshop on Applications of Computer Vision*, pages 142–147, 1996.

[173] M.-H. Yang and N. Ahuja. Detecting human faces in color images. In *Proceedings of IEEE International Conference on Image Processing*, volume 1, pages 127–130, 1998.

[174] M.-H. Yang and N. Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. In *Proceedings of the SPIE: Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 458–466, 1999.

[175] M.-H. Yang, N. Ahuja, and D. Kriegman. Mixtures of linear subspaces for face detection. In *Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition*, pages 70–76, 2000.

[176] M.-H. Yang, D. Roth, and N. Ahuja. A SNoW-based face detector. In S. A. solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 855–861. MIT Press, 2000.

[177] K. C. Yow and R. Cipolla. A probabilistic framework for perceptual grouping of features for human face detection. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 16–21, 1996.

[178] K. C. Yow and R. Cipolla. Feature-based human face detection. *Image and Vision Computing*, 15(9):713–735, 1997.

[179] K. C. Yow and R. Cipolla. Enhancing human face detection using motion and active contours. In *Proceedings of the Third Asian Conference on Computer Vision*, pages 515–522, 1998.

[180] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[181] 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*, pages 336–341, 1998.