massachusetts institute of technology — artificial intelligence laboratory

Rotation Invariant Real-time Face Detection and Recognition System

Purdy Ho

AI Memo 2001-010 May 31, 2001
CBCL Memo 197

© 2001 massachusetts institute of technology, cambridge, ma 02139 usa — www.ai.mit.edu

Abstract

In this report, a face recognition system that is capable of detecting and recognizing frontal and rotated faces was developed. Two face recognition methods focusing on the aspect of pose invariance are presented and evaluated — the whole face approach and the component-based approach. The main challenge of this project is to develop a system that is able to identify faces under different viewing angles in realtime. The development of such a system will enhance the capability and robustness of current face recognition technology.

The whole-face approach recognizes faces by classifying a single feature vector consisting of the gray values of the whole face image. The component-based approach first locates the facial components and extracts them. These components are normalized and combined into a single feature vector for classification. The Support Vector Machine (SVM) is used as the classifier for both approaches.

Extensive tests with respect to the robustness against pose changes are performed on a database that includes faces rotated up to about 40° in depth. The component-based approach clearly outperforms the whole-face approach on all tests. Although this approach is proven to be more reliable, it is still too slow for real-time applications. That is the reason why a real-time face recognition system using the whole-face approach is implemented to recognize people in color video sequences. [1]

This report describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences and in the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology.

This research was sponsored by grants from: Office of Naval Research (DARPA) under contract No. N00014-00-1-0907, National Science Foundation (ITR) under contract No. IIS-0085836, National Science Foundation (KDI) under contract No. DMS-9872936, and National Science Foundation under contract No. IIS-9800032.

Additional support was provided by: Central Research Institute of Electric Power Industry, Center for e-Business (MIT), Eastman Kodak Company, DaimlerChrysler AG, Compaq, Honda R&D Co., Ltd., Komatsu Ltd., Merrill-Lynch, NEC Fund, Nippon Telegraph and Telephone, Siemens Corporate Research, Inc., Toyota Motor Corporation and The Whitaker Foundation.

[1] Part of this report is published in Face Recognition with Support Vector Machines: Global versus Component-based Approach, IEEE ICCV 2001.

1 Introduction

The development of biometrics identification systems is a very popular research topic in artificial intelligence. Biometrics security systems have a high potential of providing simple and powerful protection of the privacy of users and the information stored in the mobile electronic devices, such as cellular phones and laptops. Different kinds of biometrics security systems have been actively applied to commercial hand-held devices. For example, face recognition screensavers have been implemented in some laptop models, fingerprint and retinal pattern recognition technology has been applied to high-level security building access, and voice identification is a popular research topic in the cellular phone industry.

Among all the applications of biometrics identification, face recognition is most suitable for automatic visual surveillance systems. Face recognition can also be easily applied to hand-held devices with the availability of cheap and powerful hardware. That is the reason why a great deal of research is focusing on developing new algorithms and enhancing the capability and robustness of face recognition. However, most of these systems are only capable of recognizing frontal views of faces. The frontal face recognition approach is adequate in access control applications where the user is consistent from session to session, e.g. accessing a personal laptop or a cellular phone. However, in surveillance applications where the user is often not aware of the task, it is important for the system to handle faces rotated in depth.

Rotation invariant face recognition is an important issue to address because of its many real-world applications, especially in surveillance. It is clear that if a robust system is created, it will have a huge impact on many different areas of commercial and military technology.

1.1 Previous Work

A survey on face recognition is described in [5]. Most of the previous work on face recognition was primarily based on classifying frontal views of faces, assuming that the person was looking straight into the camera. The approaches adopted and developed in this report build on previous work in the areas of whole-face and component-based face detection [8]. In order to improve the robustness of the system, rotation of faces is taken into account in designing the system.

1.1.1 Whole-face Approach

In the whole-face approach, a single feature vector is used to represent the face image as an input to a classifier. Some common techniques include single-template matching, eigenfaces [13] [15], Fisher's discriminant analysis [2], and neural networks [7]. Eigenfaces, described in [13], represent face images in a low dimensional feature space using principle component analysis (PCA). In [7], back-propagation

neural networks were used to perform identification. These systems work well for classifying frontal views of faces. However, they are not robust against pose changes since the whole-face approach is highly sensitive to translations and rotations of the face. Figure 1 shows that a rotated face cannot be matched by a single whole-face pattern. To avoid this problem, an alignment stage can be added before classifying the face. Aligning an input face image with a reference face image requires computing correspondences between the two face images. The correspondences are usually determined for a small number of prominent points in the face, e.g. the centers of the eyes, the nostrils, or the corners of the mouth. Based on these correspondences, the input face image can be warped to a reference face image. In [4], face recognition is performed by independently matching templates of three facial regions (both eyes, nose and mouth). The configuration of the components during classification is unconstrained since the system does not include a geometrical model of the face. A similar approach with an additional alignment stage was proposed in [3]. Active shape models are used in [10] to align input faces with model faces.

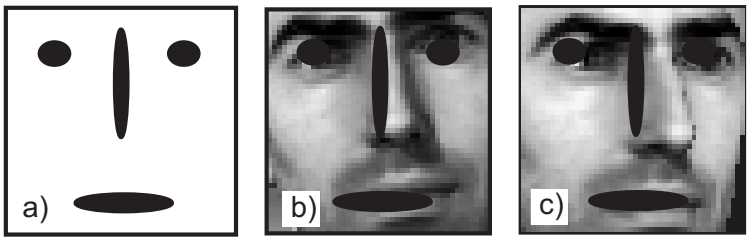


Figure 1: The problem caused by rotations.

1.1.2 Component-based Approach

Alternative to the whole-face approach, the component-based approach recognizes faces by first detecting the facial components. The advantage of using component-based recognition is that local facial components are less sensitive to translation and rotation than the whole face pattern. The component-based approach can compensate for pose changes by allowing a flexible geometrical relation between the components in the classification stage. Elastic grid matching, described in [18], uses Gabor wavelets to extract features at grid points and graph matching for the proper positioning of the grid. The recognition was based on wavelet coefficients that were computed on the nodes of the elastic graph. In [12], a window was shifted over the face image and the discrete cosine transform (DCT) coefficients computed within the window were fed into a 2-D Hidden Markov Model.

1.2 Our Approach

Both the whole-face approach and the component-based approach are implemented and evaluated in this report.

The whole-face approach consists of a face detector that extracts the face part from an image and propagates it to a set of SVM classifiers that perform face recognition. By using a face detector, the face part of the image is extracted from the background so the translation and scale invariance is achieved. Due to changes in the pose and viewpoints, there are many variations in face images, even of the same person, which make the recognition task difficult. For this reason, the database of each person is split into viewpoint-specific clusters. A linear SVM classifier is trained on each cluster so as to distinguish one person from all other people in the database. A real-time face recognition system based on the whole-face approach with clustering is built. Figure 2 shows a block diagram of the real-time system.

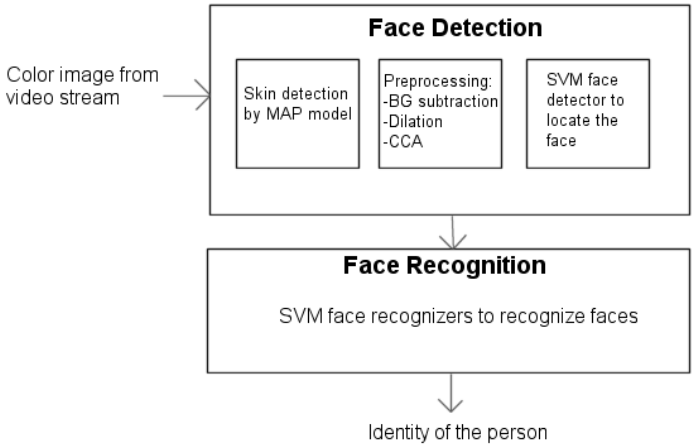


Figure 2: The system overview.

The component-based approach uses a face detector that detects and extracts local components of the face. The face detector consists of a set of SVM classifiers that locate different facial components and a single geometrical classifier that checks if the configuration of the components matches a learned geometrical face model. The detected components are extracted from the image, normalized in size, and fed into a set of SVM classifiers for face recognition.

The outline of this report is as follows: Section 2 gives an overview of the SVM classifier and its application on multi-class classification. Section 3 describes various real-time image processing techniques for preprocessing the images obtained from the video stream. Section 4 explains the whole-face and component-based face

recognition approaches. Section 5 contains experimental results and a comparison between the two face recognition approaches. Section 6 concludes the report and suggests future work.

2 Support Vector Machine Classifier

Support vector machines (SVMs) have been extensively used as classifiers in pattern recognition. The SVM performs binary pattern classification by finding a decision surface which separates the training data into two classes.

2.1 Binary Classification

Fig. 3a shows a 2-D problem for linearly separable data. In many two-class pattern classification problems, classes can be separated by more than one hyperplane. The dotted lines indicate all possible hyperplanes which separate the two classes. SVM determines the formulation of the hyperplane, which maximizes the distance between the two classes, and chooses it to be the decision plane. The decision plane is denoted by $f = 0$ in Fig. 3b. In [16], this hyperplane is described as the optimal hyperplane with respect to the structural risk minimization. Support vectors (SVs) are the closest points of each class to the decision plane. They are the circled data points in Fig. 3b. The distance from the decision plane to the SVs is denoted by M in Fig. 3b and is called the margin between the two classes.

SVM belongs to the class of margin maximizing classifiers because it chooses the hyperplane which gives the largest margin to be the decision surface. The SVM decision function has the following form:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x_i} \cdot \mathbf{x} + b \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, l$. Each point of $\mathbf{x}_i$ belongs to one of the two classes identified by the label $y_i \in \{-1, 1\}$. The coefficients α_i and b are the solutions of a quadratic programming problem [16]. α_i is non-zero for support vectors and is zero otherwise. Classification of a new data point $\mathbf{x}$ in the test set is performed by computing the sign of the right-hand side of Eq. (1). The distance from $\mathbf{x}$ to the hyperplane is computed as follows:

$$d(\mathbf{x}) = \frac{\sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x_i} \cdot \mathbf{x} + b}{|| \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x_i} ||} \tag{2}$$

The formulation in eq. (2) is the normalized output from eq. (1). It is the distance of a data point from the decision surface. The sign of d is the classification result

for the test data $\mathbf{x}$, and $|d|$ is the distance from $\mathbf{x}$ to the decision plane. The farther away a point is from the decision plane, the more reliable the classification result is.

When the data are not linearly separable, each point $\mathbf{x}$ in the input space is mapped to a point $\mathbf{z} = \Phi(\mathbf{x})$ of a higher dimensional feature space where the data can be separated by a hyperplane. The mapping $\Phi(\cdot)$ is represented in the SVM classifier by a kernel function $\mathrm{K}(\cdot, \cdot)$. The decision function of the SVM is thus:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i K(\mathbf{x_i}, \mathbf{x}) + b \tag{3}$$

An important family of kernel functions is the polynomial kernel:

$$K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^p \tag{4}$$

where p is the degree of the polynomial. In this case, the components of the mapping $\Phi(\mathbf{x})$ are all the possible monomials of input components up to the degree p. Most of the experiments in this project make use of the linear SVMs because the data are linearly separable, but in one of the experiments the polynomial second degree SVM is used for comparison.

2.2 Multi-class Classification

In order to classify q classes with SVM, the one-vs-all approach is used. In this approach, q SVMs are trained and each of the SVMs separates a single class from all the remaining classes [6] in the training set. The classification in our experiments is done by running a feature vector through all q SVMs. The identity of the person is established according to the SVM that produces the highest normalized output given by Eq. (2).

In the real-time system, the one-vs-all approach has a slightly different definition. Instead of separating a person from all other people in the database, two additional classes are used: the background class and the generic face class. The background class contains images of the empty office and the generic face class contains images of different people who are not the the positive database. These two classes are added to the negative class of all the SVMs for rejection.

3 Preprocessing

Images from the video sequence are preprocessed in four steps. First, a skin detector based on a maximum *a posteriori* (MAP) probabilistic model is used to separate the skin pixels from the non-skin pixels in a scene. Second, background subtraction is used to remove the static background and the background pixels that are mistaken as skin pixels. Each of these steps generates a binary image. By combining these two

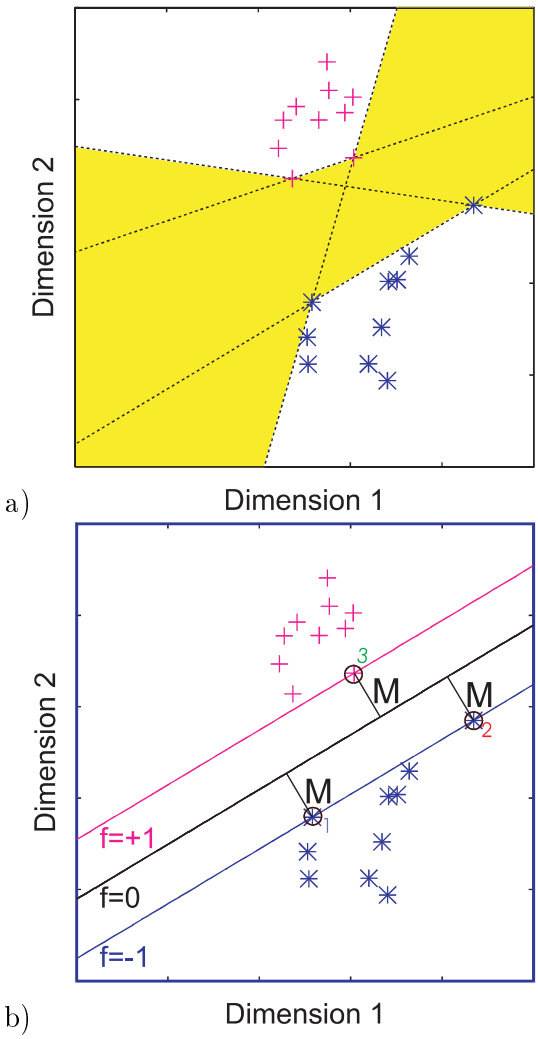


Figure 3: a) The gray area shows all possible hyperplanes which separate the two classes. b) The optimal hyperplane maximizes the distance between the SVs of the two different classes. The points (1, 2, and 3) are the SVs. The distance M is the margin.

binary images, a new binary image that shows the presence of skin pixels is produced. Third, a morphological operation is applied to dilate the combined binary image. Finally, the connected component analysis algorithm determines the largest region in the dilated image and claims that this is the face part of a person.

3.1 Skin Detection

The skin detector is trained and used to classify skin pixels from the non-skin pixels in video sequences [9]. Since the presence of skin pixels represents the presence of people, the skin detector is thus a person detector. Two separate sets of color images are used for training and testing the skin detector. The separation of the skin part from the non-skin part of color images is based on the distinct color properties of the human skin. Each pixel is represented by its normalized red and green colors. The classification of a skin pixel and a non-skin pixel is performed by a maximum *a posteriori* (MAP) probabilistic model adapted on a training set of skin pixels and non-skin pixels. The input to the MAP model is the normalized red and green color value of a pixel and the output is one of two classes: the skin class or the non-skin class. Fig. 4 is a block diagram that shows the training of the skin detector.

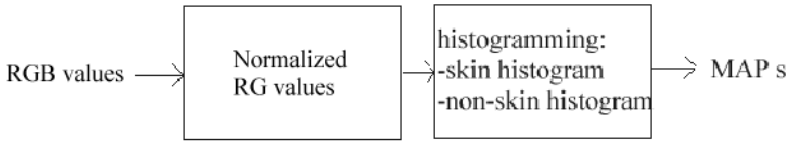


Figure 4: The block diagram of the skin detector.

The skin training set was obtained by taking pictures of the CBCL staff with a CCD color camera. Twenty pictures at a resolution of 640 x 480 pixels were taken of the faces of each person at 5 frames per second. The skin parts in these images were extracted and labeled manually. An independent set of skin images were used as test set. The non-skin training set was obtained by taking pictures of the empty office as well as some clothing samples with the same CCD camera.

The normalized red and green color space is often used for skin detection since it reduces sensitivity to changes in illumination and intensity. By normalizing the colors, luminance information is not taken into account. This makes the skin detector work for both light and dark skin colors. Normalized red and green color pixels from the skin and non-skin training sets are used to construct the skin and non-skin histograms.

Histogramming belongs to the non-parametric density estimation in which the probability density functions depend on the data itself and the form of the function

is not specified in advance. The 256 values of red and green are quantized into 32 discrete segments with 8 values in each segment. The 2-D histograms of the skin pixels and the non-skin pixels are obtained by dividing each of the red and green axes into 32 sections. These histograms approximate the probability density functions of the r-g color given the presence of a skin pixel and the presence of a non-skin pixel.

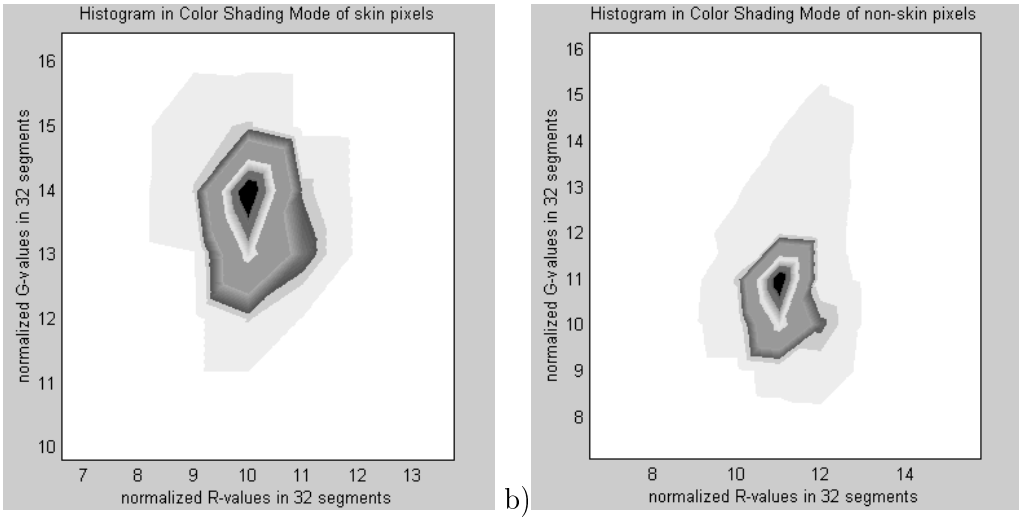


Figure 5: a) The 32 x 32 bin histogram of the skin pixels in the r-g plane. b) The 32 x 32 bin histogram of the non-skin pixels in the r-g plane. The darker color represents higher probability of occurrence.

A 32×32 bin skin histogram and a 32×32 bin non-skin histogram are constructed from the skin pixels and non-skin pixels of the training set. The conditional probabilities of the r-g color of a pixel given a skin pixel $\mathrm{P}(rg \mid skin)$ and the conditional probabilities of the r-g color of a pixel given a non-skin pixel $\mathrm{P}(rg \mid nonskin)$ are computed. Fig. 5a and 5b show the skin and non-skin histograms respectively. These two histograms are used to generate the MAP model of the skin detector. The equation of the MAP model is given in Eq. (5):

$$\frac{P(rg|skin)}{P(rg|nonskin)} > \frac{P(nonskin)}{P(skin)} \tag{5}$$

If the left-hand side of Eq. (5) is greater than the right-hand side, then the pixel is classified as a skin pixel. Otherwise, the pixel is classified as a non-skin pixel. The quantity on the right-hand side of the inequality is called the detection threshold, which is the ratio of the *a priori* probabilities. The *a priori* probabilities can be estimated from the training set. A reasonable choice of $\mathrm{P}(skin)$ can be obtained by dividing the total number of skin pixels by the total number of skin and non-skin

pixels. The decision boundary is determined by the border of the overlapping region of the two histograms. A receiver operating characteristic (ROC) curve shows the relationship between correct detection P(*"skin"* | *skin*) and false positive P(*"skin"* | *non-skin*) as a function of the detection threshold. The ROC of the skin detector test set is shown in Fig. 6. The performance of the classifier is determined by the area under the ROC curve and the amount of overlap between the skin and the non-skin histograms. Fig. 7b shows that a lot of background pixels were mistaken to be skin pixels. In order to solve this problem, the background subtraction algorithm is applied to eliminate these misclassifications.

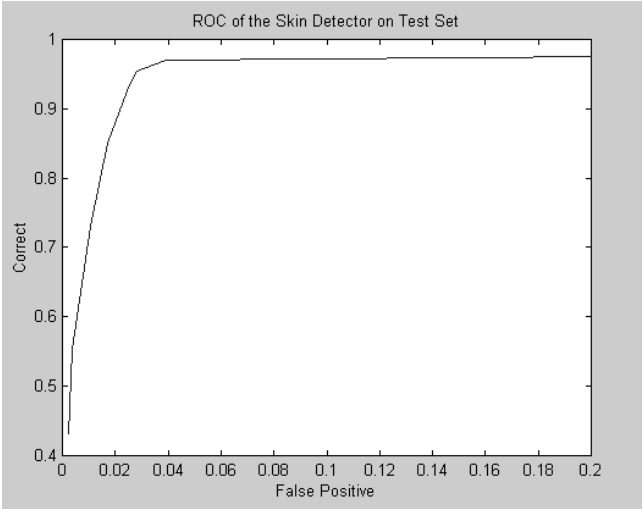


Figure 6: The ROC curve of the Skin Detector.

3.2 Background Subtraction

In background subtraction, the difference between an image from the video stream and the stored background image is computed and a binary image is produced in order to detect new objects in the scene. In this case, background subtraction is used to remove the background parts that are mistaken to be skin parts. This allows the use of a lower skin detection threshold and thus the skin detection rate can be increased. The background image updates itself at 0.5 frames per second. However, it will not update when the difference between the new image from the video stream and the stored background image is great. The updating algorithm is designed to avoid updating when there are no new objects entering the scene. For example, if the illumination condition in the room changes, the difference between the stored image and the new image will be big, but there is no new object entering

the scene. An example of the binary background subtraction image is shown in Fig. 7c. By combining the skin detection and the background subtraction binary images in Fig. 7b and 7c logically, a new binary image in Fig. 7d is produced which shows the presence of a person.

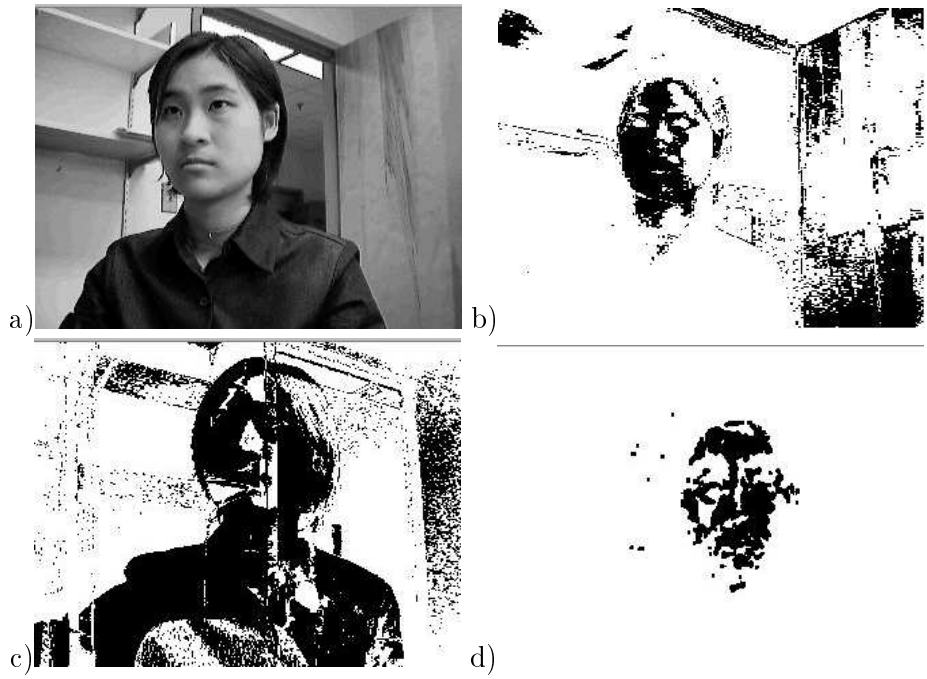


Figure 7: a) The original color image. b) The resulting binary image of skin detection. c) The background subtraction binary image. d) The combined skin detection and background subtraction binary image.

3.3 Morphological Operation

Mathematical morphology is a field that involves the study of topological and structural properties of objects based on their images. The goal of using a morphology operation in binary images is to represent black pixels by regions in order to give a complete description of the image. A region in a binary image is a set of connected pixels with the same color. In order to group the skin-pixels into a region, the eight neighboring pixels of a particular pixel are considered. A pixel has two horizontal and two vertical neighbors that are each a unit distance from the pixel itself. There are also four diagonal neighbors. Together they form the eight neighbors of a pixel. A 3×3 all-white dilation filter is applied to the combined skin detection and background subtraction binary images. This 3×3 window is convolved with the

a) b)

Figure 8: a) The combined skin detection and background subtraction binary image. b) The dilated binary image.

combined binary image. If the number of black pixels within the eight neighboring pixels is greater than the predefined minimum number of black pixels, then all the nine pixels within the window will be set to black. Otherwise, all the nine pixels will be set to white. Fig. 8a shows the image before dilation and Fig. 8b shows the image after dilation.

3.4 Connected Component Analysis (CCA)

Connectivity between pixels is commonly used in establishing boundaries of objects and regions in binary images. Connected component analysis transforms a binary image into a graph, where each node of the graph represents a connected region and the boundaries of the region represent spatial relationships between regions [1]. CCA is used for finding the largest connected region in a binary image. The dilated image is the input to the connected component analysis. The black region in the dilated image represents the skin part. CCA finds the largest connected region and claims that to be the face part. A bounding box is drawn to surround this region of interest (ROI) in the original color video stream of the real-time system. Fig. 9 shows the ROI in the video stream. Face detection and recognition algorithms are applied to the bounding box extracted from the video stream.

4 Face Recognition

4.1 Whole-face Approach

The whole-face approach consists of two stages. In the face detection stage, a face is detected and extracted from the gray value image. In the recognition stage, the person's identity is established.

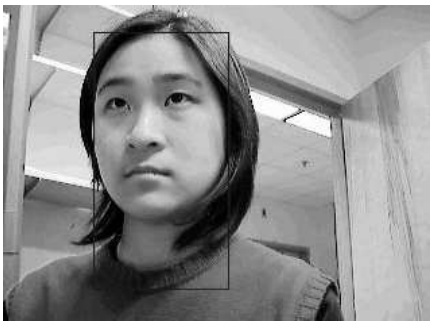


Figure 9: The ROI is indicated by the bounding box.

4.1.1 Face Detection

A linear SVM face detector similar to the one described in [8] is trained and used to extract the face part from the bounding box obtained from the video stream. The training data for the linear SVM face detector are generated by rendering seven textured 3-D head models [17]. The heads are rotated between -30° and +30° in depth and are illuminated by ambient light and a single directional light pointing towards the center of the face. 3,590 synthetic face images of size 58×58 pixels are generated to form the positive training data. The negative training data initially consists of 10,209 58×58 non-face patterns randomly extracted from 502 non-face images. The negative training data is further enlarged to 13,655 images by bootstrapping [14]. Bootstrapping is done by applying the linear SVM face detector, which trained on the initial negative set, to the 502 non-face images. The false positives (FPs) generated are added to the negative training data to build the final negative training set with 13,655 images. Then a new linear SVM face detector is retrained with the enlarged negative training set.

The face part extracted by the SVM face detector is converted into gray values and is re-scaled to 40×40 pixels. A best-fit intensity plane is subtracted from the gray values to compensate for cast shadows [14]. Histogram equalization is also applied to remove variations in image brightness and contrast. The 1,600 gray values of each face image are then normalized to the range between 0 and 1. Each image is represented by a single feature vector of length 1,600 because the face image has 40×40 pixels. These feature vectors are the inputs to the linear SVM face recognizers. Fig. 10 shows the training of the whole-face approach. Some face detection results are shown in Fig. 11.

4.1.2 Face Recognition

Changes in the head pose of a person lead to strong variations in the faces. These are considered in-class variations and they complicate the recognition task. The linear

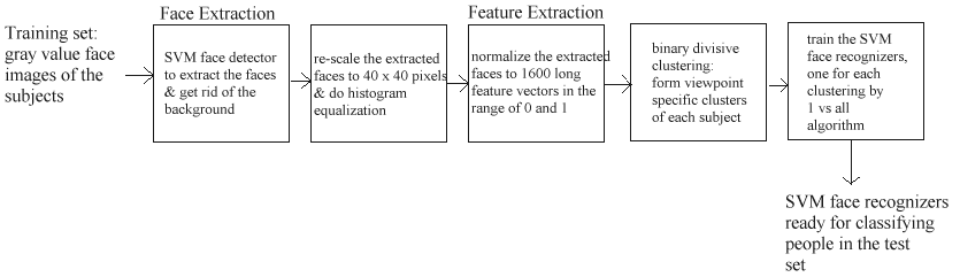


Figure 10: The training process of the whole-face approach.

Figure 11: The upper 2 rows are the original images before face extraction. The lower 2 rows show the face parts extracted by the SVM face detector. These face-extracted images are the training set of the face recognition system.

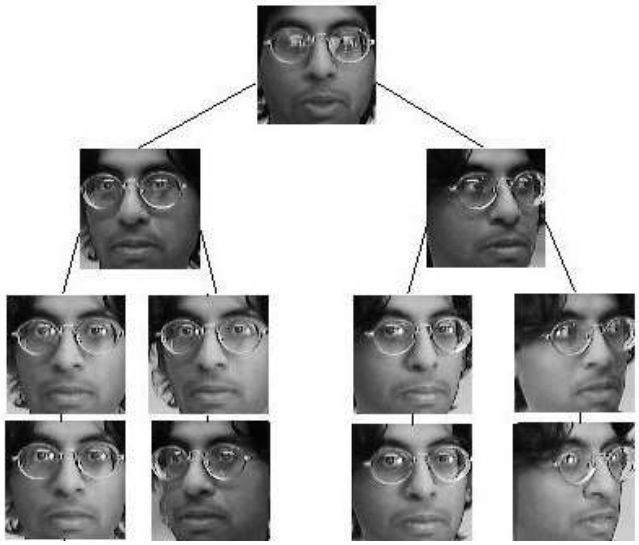


Figure 12: Binary tree of face images generated by divisive clustering.

SVM classifier cannot always separate faces of one person with different rotations from all other people without introducing training errors. In this case, the training set of each person is split into several smaller viewpoint-specific clusters by the divisive binary clustering algorithm [11]. This algorithm starts with an initial cluster that includes all the feature vectors of a person, denoted by $\mathbf{x}_n \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$ in Eq. (6), where N is the number of faces in the cluster. During each iteration, the algorithm creates a hierarchy by successively splitting the highest variance cluster into two new clusters. The variance of a cluster is calculated as:

$$\sigma^2 = min\{\frac{1}{N} \cdot \sum_{m=1}^{N} ||\mathbf{x}_n - \mathbf{x}_m||^2\}_{n=1}^{N} \tag{6}$$

where $\mathbf{x}_m$ is the average face of the cluster. The process repeats until the number of clusters reaches the predefined number. In these experiments, the predefined number of clusters is four. After clustering, the face with the minimum distance to all other faces in the same cluster is chosen to be the average face of the cluster. The clusters can be arranged in a binary tree. Fig. 12 shows the result of clustering applied to the training images of a person in the database. The nodes represent the average faces and the leaves represent faces in the final clusters. As expected, divisive clustering performs a viewpoint-specific grouping of faces.

The whole-face approach is a multi-class classification problem. The one-vs-all strategy described in section 2.2 is applied to the SVM training. A linear SVM is trained to distinguish between images in one cluster (label +1) and images of other people in the training set (label -1), so the total number of SVMs trained is equal

to the total number of clusters for all people. In this case, each SVM is associated to one cluster of each person. The class label y of a feature vector $\mathbf{x}$ is computed as follows:

$$y = n \text{ if } d_n(\mathbf{x}) + t > 0 \tag{7}$$
$$y = 0 \text{ if } d_n(\mathbf{x}) + t \leq 0$$
$$\text{with } d_n(\mathbf{x}) = max\{d_i(\mathbf{x})\}_{i=1}^{q}$$

where $d_i(\mathbf{x})$ is the distance of pattern $\mathbf{x}_i$ from the decision plane computed according to Eq. (2). The classification threshold is denoted as t. Classification is done according to the value of the class label y computed by Eq. (7) with q being the number of clusters of all people in the training set. A non-zero class label stands for recognition and the class label 0 stands for rejection. When $d_i(\mathbf{x})$ is too small, this pattern is too close to the decision plane. In this case, the system cannot tell which class this pattern belongs to, and thus the pattern is rejected.

4.2 Component-based Approach

The whole-face approach is highly sensitive to image variations caused by changes in the pose of the face as shown in Fig. 1. Since the changes in facial components due to rotations are relatively small compared to those in the whole face pattern, the component-based approach is implemented in order to avoid the problems caused by the whole-face approach. Fig. 13 shows the training process of the component-based approach.

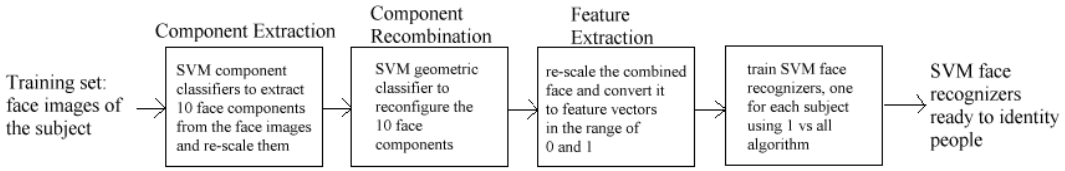


Figure 13: The training process of the component-based approach.

4.2.1 Face Detection

In order to detect the face, a two-level component-based face detector [8] is used. The principles of the system are illustrated in Fig. 14. On the first level, component classifiers independently detect 14 facial components. On the second level, a geometrical configuration classifier performs the final face detection by combining the

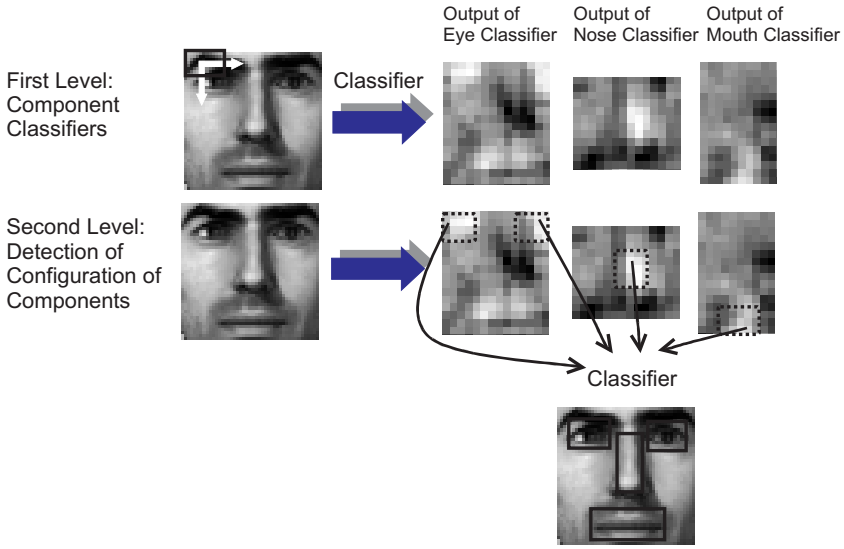


Figure 14: System overview of the component-based face detector using four components.

facial components resulting from the 14 component classifiers. The maximum continuous outputs of the component classifiers within the rectangular search regions around the expected positions of the components are used as inputs to the geometrical configuration classifier. The search regions have been calculated from the mean and the standard deviation of the components' locations in the training images. The geometrical classifier is used for arranging the components in the proper facial configuration. It is provided with the precise positions of the detected components relative to the upper left corner of the 58×58 window. The 14 facial components used in the detection system are shown in Fig. 15a. The shapes and positions of the components have been automatically determined from the training data in order to provide maximum discrimination between face and non-face images [8]. The face images in the training set are the same as that for the whole-face detector.

4.2.2 Face recognition

The component-based detector runs over each image in the training set and the components are extracted from each image. Only 10 out of the 14 original components are kept for face recognition because the remaining ones either contain few gray value structures or strongly overlap with other components. The 10 selected components are shown in Fig. 15b. The component-based face detector applied to face images in the original training set shown in the first 2 rows of Fig. 11 and the final training set of the component-based recognition system are shown in Fig. 16.

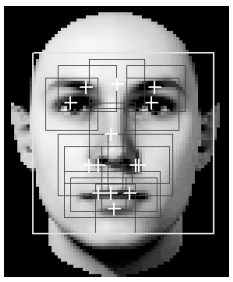
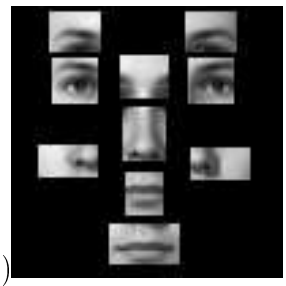
a)  b) 

Figure 15: (a) Shows the 14 components of the face detector. The centers of the components are marked by white crosses. The 10 components used for face recognition are shown in (b).

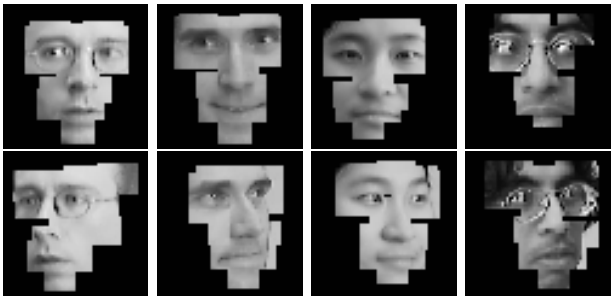


Figure 16: Examples of component-based face detection. Face parts covered by the 10 components are used as training data for face recognition.

By recombining the components, background pixels are successfully removed. In order to generate the input to the face recognition classifier, the components of each image are normalized in size. Their gray values are normalized to a range of 0 and 1 and are then combined into a single feature vector. Again, the one-vs-all strategy of multi-class classification is used. A linear SVM classifier is trained for each person in the database. The classification result is determined according to Eq. (7).

5 Results

5.1 Database

The training data for the face recognition system were recorded with a digital video camera at a frame rate of about 5Hz. The training set consists of 8,593 gray face images of five subjects; 1,383 of these images are frontal views. The resolutions of the face images range between 80×80 and 130×130 pixels with rotations in azimuth up to about $\pm$ 40°.

The test set was recorded with the same camera but on a separate day and with different illumination and background. The test set includes 974 images of all five subjects in the database. The rotation in depth is again up to about $\pm$ 40°. Fig. 17 and Fig. 18 show the experimental procedures when using the whole-face approach and the component-based approach.

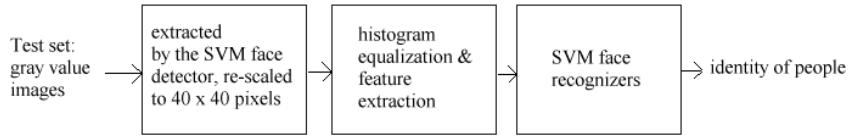


Figure 17: Overwiew of the whole-face approach experiment.

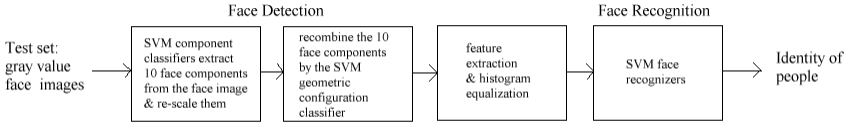


Figure 18: Overwiew of the component-based approach experiment.

5.2 Experiments

Two sets of experiments were carried out.

The first set of experiments was trained on all 8,593 rotated and frontal face images and tested on an independent test set with 974 frontal and rotated faces of all the subjects. This experiment contained four different tests:
1. Whole-face approach with one linear SVM for each person.
2. Whole-face approach with one linear SVM for each cluster.
3. Whole-face approach with one 2nd degree polynomial SVM for each person.
4. Component-based approach with one linear SVM for each person.

The second set of experiments was trained only on the 1,383 frontal face images but tested on the same test set used in the first set of experiments. This experiment contained three different tests:
1. Whole-face approach with one linear SVM for each person.
2. Whole-face approach with one linear SVM for each cluster.
3. Component-based approach with one linear SVM for each person.

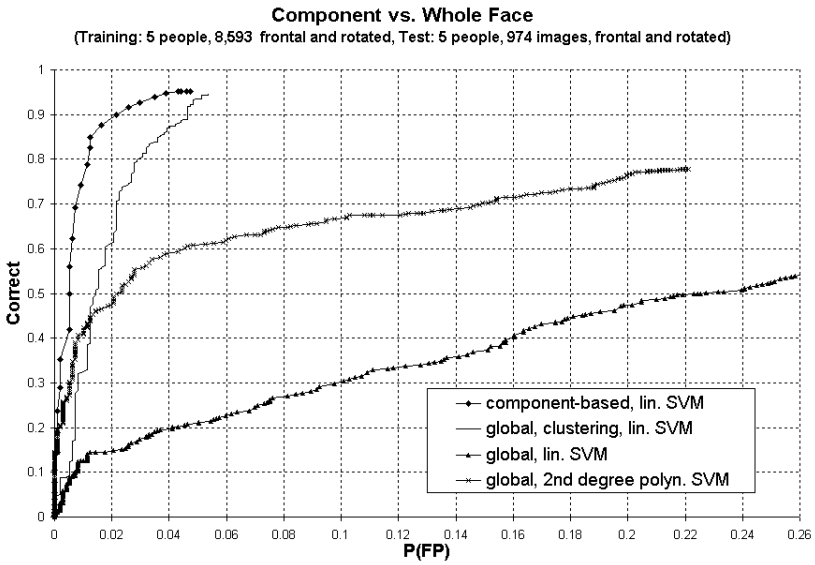
The ROC curves of these two set of experiments are shown in Fig. 19a and Fig. 19b. Each point on the ROC curve corresponds to a different value of the classification threshold t from Eq. (7). Some results of the component-based recognition system are shown in Fig. 20.

In both sets of experiments, the component-based approach clearly outperformed the whole-face approach, even though the classifiers used in the component-based approach (linear SVMs) are less powerful than those used in the whole-face approach (polynomial second degree SVMs and SVMs with clustering).

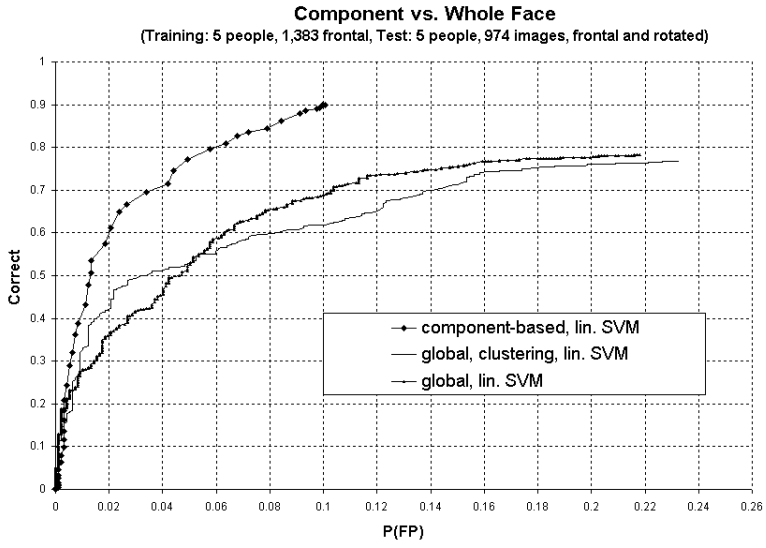
Clustering also leads to a significant improvement of the whole-face approach with the training set including the rotated faces. Clustering generates viewpoint-specific clusters that have smaller in-class variations than the whole set of images of a person, so the whole-face approach with clustering and linear SVMs is superior to the whole-face approach without clustering and with a non-linear SVM. This shows that weaker classifiers trained on properly chosen subsets of the data can outperform a single and more powerful classifier trained on the whole data set.

6 Conclusion and Future Work

A whole-face approach and a component-based approach of face recognition were implemented and their performances with respect to robustness against pose changes were compared. The component-based approach detected and extracted a set of 10 facial components and arranged them in a single feature vector that was classified by linear SVMs. The whole-face approach detected the whole face, extracted it from



a)



b)

Figure 19: (a) ROC curves trained and tested on both frontal and rotated faces. (b) ROC curves trained on frontal faces and tested on frontal and rotated faces.

Figure 20: Examples of component-based face recognition. The first 3 rows of images and the first image in the last row are correct identification. The last two images in the bottom row are misclassifications due to too much rotation and unexpected facial expression.

the image, and used it as an input to a set of viewpoint-specific SVM classifiers.

Tests were performed on both systems with a test database that included faces rotated in depth up to about $\pm$ 40°. In both sets of experiments, the component-based approach outperformed the whole-face approach. This shows that using facial components instead of the whole face pattern as input features significantly simplifies the task of face recognition. However, the speed of the component-based approach is much slower than that of the whole-face approach, since a lot more SVM classifiers are used in the component-based approach for extracting the facial components. This approach is not suitable for applications involving real-time systems for the time being. Fig. 19a shows that the performance of the whole-face approach with clustering is just slightly worse than the performance of the component-based approach. However, the recognition speed of the whole-face approach is a lot faster. This is the reason why the real-time system is implemented based on the whole-face approach.

A potential future research topic would be to reduce the number and dimensions of face components. The dimensions of the components and the combined face images can be reduced using techniques such as the principal component analysis (PCA). Fewer facial components could be selected and used in order to reduce the number of classifiers. These improvements could speed up the classification rate of the component-based approach and make it more desirable for use in real-time applications. Powerful computers with multi-processors could also be used to parallel-process the component classifications in order to reduce the computation time when implementing real-time systems. More experiments should be done on larger and standardized test sets so as to compare my system with the existing ones.

References

[1] N. Bartneck. *Ein Verfahren zur Umwandlung der ikonischen Bildinformation digitalisierter Bilder in Datenstrukturen zur Bildauswertung.* PhD thesis, Technische Universitat Carolo–Wilhelmina, Braunschweig, 1987.

[2] P. Belhumeur, P. 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.

[3] D. J. Beymer. Face recognition under varying pose. A.I. Memo 1461, Center for Biological and Computational Learning, M.I.T., Cambridge, MA, 1993.

[4] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[5] R. Chellapa, C. Wilson, and S. Sirohey. Human and machine recognition of faces: a survey. *Proceedings of the IEEE*, 83(5):705–741, 1995.

[6] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[7] M. Fleming and G. Cottrell. Categorization of faces using unsupervised feature extraction. In *Proc. IEEE IJCNN International Joint Conference on Neural Networks*, pages 65–70, 90.

[8] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. In *AI Memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA*, 2000.

[9] M. Jones and J. Rehg. Statistical color models with application to skin detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 1(280), 1999.

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

[11] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[12] A.V Nefian and M.H Hayes. An embedded hmm-based approach for face detection and recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3553–3556, 1999.

[13] Soirovich and Kerby. Low-dimensional procedure for the characterization of human faces. In *Opt. Soc. Am. A*, 1987.

[14] K.-K. Sung. *Learning and Example Selection for Object and Pattern Recognition*. PhD thesis, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA, 1996.

[15] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[16] V. Vapnik. *The nature of statistical learning*. Springer Verlag, 1995.

[17] T. Vetter. Synthesis of novel views from a single face. *International Journal of Computer Vision*, 28(2):103–116, 1998.

[18] J. Zhang, Y. Yan, and M. Lades. Face recognition: eigenface, elastic matching, and neural nets. *Proceedings of the IEEE*, 85(9):1423–1435, 1997.