MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL AND COMPUTATIONAL LEARNING
DEPARTMENT OF BRAIN AND COGNITIVE SCIENCES

A.I. Memo No. 1687 May, 2000
C.B.C.L Paper No. 187

Face Detection in Still Gray Images

Bernd Heisele, Tomaso Poggio, Massimiliano Pontil

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu. The pathname for this publication is: ai-publications/1500-1999/AIM-1687.ps.Z

Abstract

We present a trainable system for detecting frontal and near-frontal views of faces in still gray images using Support Vector Machines (SVMs). We first consider the problem of detecting the whole face pattern by a single SVM classifier. In this context we compare different types of image features, present and evaluate a new method for reducing the number features and discuss practical issues concerning the parameterization of SVMs and the selection of training data. The second part of the paper describes a component-based method for face detection consisting of a two-level hierarchy of SVM classifiers. On the first level, component classifiers independently detect components of a face, such as the eyes, the nose, and the mouth. On the second level, a single classifier checks if the geometrical configuration of the detected components in the image matches a geometrical model of a face.

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 is sponsored by a grant from Office of Naval Research under Contract No. N00014-93-1-3085 and Office of Naval Research under Contract No. N00014-95-1-0600. Additional support is provided by: AT&T, Central Research Institute of Electric Power Industry, Eastman Kodak Company, Daimler-Benz AG, Digital Equipment Corporation, Honda R&D Co., Ltd., NEC Fund, Nippon Telegraph & Telephone, and Siemens Corporate Research, Inc.

1 Introduction

Over the past ten years face detection has been thoroughly studied in computer vision research for mainly two reasons. First, face detection has a number of interesting applications: It can be part of a face recognition system, a surveillance system, or a video-based computer/machine interface. Second, faces form a class of visually similar objects which simplifies the generally difficult task of object detection. In this context, detecting chairs is often mentioned as an example where the high variation within the object class leads to a merely unsolvable detection problem. Besides the variability between individual objects of the same class, detection algorithms have to cope with variations in the appearance of a single object due to pose and illumination changes. Most in the past research work on face detection focussed on detecting frontal faces thus leaving out the problem of pose invariance. Although there is still some space for improvement on frontal face detection, the key issue of current and future research seems to be pose invariance.

In the following we give a brief overview on face detection techniques. One category of systems relies on detecting skin parts in color images [Wu et al. 99, Saber & Tekalp 96]. Common techniques for skin color detection estimate the distribution of skin color in the color space using labeled training data [Jebara & Pentland 97, Jones & Rehg 99]. A major problem of skin color detection is its sensitivity to changes in the spectral composition of the lighting and to changes in the characteristics of the camera. Therefore, most systems generate hypotheses by the skin color detector and verify them by a front-end pattern classification module. Depending on the application there are other efficient ways of generating object hypotheses. In case of a static video camera and a static background scenery, background subtraction [Ivanov et al. 98, Toyama et al. 99] is commonly used to detect objects.

Another category of algorithms performs face detection in still gray images. Since there are no color and motion cue available, face detection boils down to a pure pattern recognition task. One of the first systems for detecting faces in gray images combines clustering techniques with neural networks [Sung 96]. It generates face and non-face prototypes by clustering the training data consisting of 19×19 histogram normalized face images. The distances between an input pattern and the prototypes are classified by a Multi-Layer Perceptron. In [Osuna 98] frontal faces are detected by a SVM with polynomial kernel. A system able to deal with rotations in the image plane was proposed by [Rowley et al. 97]. It consists of two neural networks, one for estimating the orientation of the face, and another for detecting the derotated faces. The recognition step was improved [Rowley et al. 98] by arbitrating between independently trained networks of identical structure. The above described techniques have common classifiers which were trained on patterns of the whole face. A naïve Bayesian approach was taken in [Schneiderman & Kanade 98]. The method determines the empirical probabilities of the occurrence of 16×16 intensity patterns within 64×64 face images. Assuming statistical independence between the small patterns, the probability for the whole pattern being a face is calculated as the product of the probabilities for the small patterns. Another probabilistic approach which detects small parts of faces is proposed in [Leung et al. 95]. Local feature extractors are used to detect the eyes, corner of the mouth, and tip of the nose. Assuming that the position of the eyes is properly determined, the geometrical configuration of the detected parts in the image is matched with a model configuration by conditional search. A related method using statistical models is published in [Rikert et al. 99]. Local features are extracted by applying multi-scale and multi-orientation filters to the input image. The responses of the filters on the training set are modeled as Gaussian distributions. In contrast to [Leung et al. 95], the configuration of the local filter responses is not matched with a geometrical model. Instead, the global consistency of the pattern is verified by analyzing features at a coarse resolution. Detecting components has also been applied to face recognition. In [Wiskott 95] local features are computed on the nodes of an elastic grid. Separate templates for eyes, the nose and the mouth are matched in [Beymer 93, Brunelli & Poggio 93].

There are two interesting ideas behind part- or component-based detection of objects. First, some object classes can be described well by a few characteristic object parts[1] and their geometrical relation. Second, the patterns of some object parts might vary less under pose changes than the pattern belonging to the whole object. The two main problems of a component-based approach are how to choose the set of discriminatory object parts and how to model their geometrical configuration. The above mentioned approaches either manually define a set of components and model their geometrical configuration or uniformly partition the image into components and assume statistical independence between the components. In our system we started with a manually defined set of facial components and a simple geometrical model acquired from the training set. In a further step we developed a technique for automatically extracting discriminatory object parts using a database of 3-D head models.

The outline of the paper is as follows: In Chapter 2 we compare different types of image features for face detection. Chapter 3 is about feature reduction. Chapter 4 contains some experimental results on the parameterization of an SVM for face detection. Different techniques for generating training sets are discussed in Chapter 5. The first part of the paper about face detection using a single SVM classifier concludes in Chapter 6 with experimental results on standard test sets. Chapter 7 describes a component-based system and compares it to a whole face detector. Chapter 8 concludes the paper.

[1]In this paper we use the expression object part both for the 3-D part of an object and the 2D image of a 3-D object part.

2 Extracting image features

Regarding learning, the goal of image feature extraction is to process the raw pixel data such that variations between objects of the same class (within-class variations) are reduced while variations relevant for separating between objects of different classes (between-class variations) are kept. Sources of within-class variations are changes in the illumination, changes in the background, and different properties of the camera. In [Sung 96] three preprocessing steps were applied to the gray images to reduce within-class image variations. First, pixels close to the boundary of the 19×19 images were removed in order to eliminate parts belonging to the background. Then a best-fit intensity plane was subtracted from the gray values to compensate for cast shadows. Histogram equalization was finally applied to remove variations in the image brightness and contrast. The resulting pixel values were used as input features to the classifier. We compared these gray value features to gray value gradients and Haar wavelets. The gradients were computed from the histogram equalized 19×19 image using 3×3 x- and y-Sobel filters. Three orientation tuned masks (see Fig. 1) in two different scales were convoluted with the 19×19 image to compute the Haar wavelets. This lead to a 1,740 dimensional feature vector. Examples for the three types of features are shown in Fig. 2.

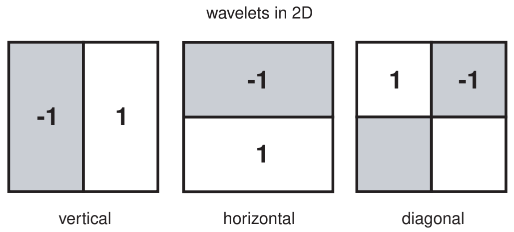

Figure 1: Convolution masks for calculating Haar wavelets.

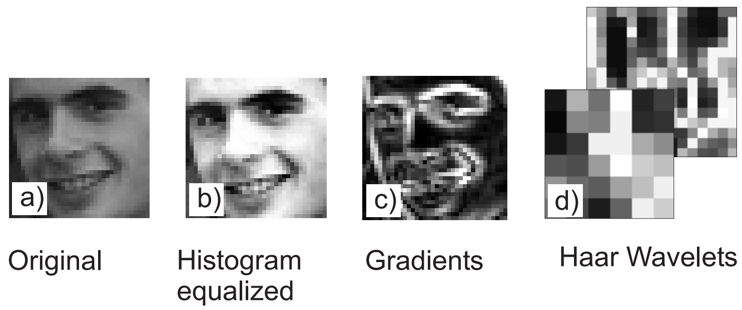

Figure 2: Examples of extracted features. The original gray image is shown in a), the histogram equalized image in b), the gray value gradients in c), and Haar wavelets generated by a single convolution mask in two different scales in d).

Gray, gray gradient and Haar wavelet features were rescaled to be in a range between 0 and 1 before they were used for training an SVM with 2nd-degree polynomial kernel. The training data consisted of 2,429 face and 19,932 non-face images. The classification performance was determined on a test set of 118 gray images with 479 frontal faces[2]. Each image was rescaled 14 times by factors between 0.1 and 1.2 to detect faces at different scales. A 19x19 window was shifted pixel-by-pixel over each image. Overall, about 57,000,000 windows were processed. The Receiver Operator Characteristic (ROC) curves are shown in Fig. 3, they were generated by stepwise variation of the classification threshold of the SVM. Histogram normalized gray values are the best choice. For a fixed FP rate the detection rate for gray values was about 10% higher than for Haar wavelets and about 20% higher than for gray gradients. We trained an SVM with linear kernel on the outputs of the gray/gradient and gray/wavelet classifiers to find out whether the combination of two feature sets improves the performance. For both combinations the results were about the same as for the single gray classifier.

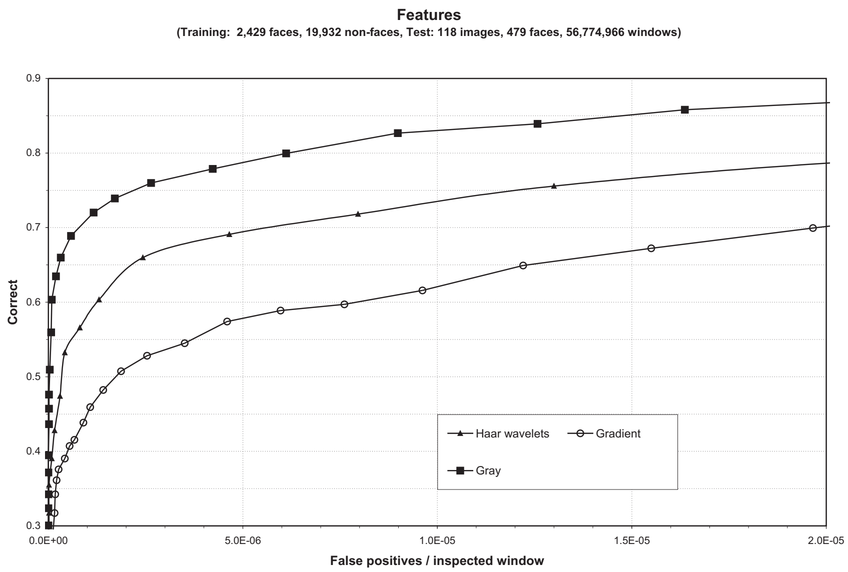

Figure 3: ROC curves for SVMs with 2nd-degree polynomial kernel trained on different types of image features.

[2]The test set is a subset of the CMU test set 1 [Rowley et al. 97] which consists of 130 images and 507 faces. We excluded 12 images containing line-drawn faces and non-frontal faces.

3 Feature Reduction

The goal of feature reduction is to improve the detection rate and to speed-up the classification process by removing class irrelevant features. We investigated two ways of feature reduction: a) Generating a new set of features by linearly combining the original features and b) selecting a subset of the original features.

3.1 Linear combination of features

We evaluated two techniques which generate new feature sets by linearly combining the original features:

- Principal Component Analysis (PCA) is a standard technique for generating a space of orthogonal, uncorrelated features.
- Iterative Linear Classification (ILC) determines the most class discriminant, orthogonal features by iteratively training a linear classifier on the labeled training samples. The algorithm consists of two steps:

  a) Determine the direction for separating the two classes by training a linear classifier on the current training samples.

  b) Generate a new sample set by projecting the samples into a subspace that is orthogonal to the direction calculated in a) and continue with step a).

  The new N-dimensional feature space is spanned by the N first directions calculated in step a). In the following experiments we used an SVM as linear classifier.

Both techniques were applied to the 283 gray value features described in Chapter 2. We downsized the previously used training and test sets in order to perform a large number of tests. The new negative training set included 4,550 samples randomly selected from the original negative training set. The positive training data remained unchanged. The new test set included all face patterns and 23,570 non-face patterns of the CMU test set 1. The non-face patterns were selected by the classifier described in Chapter 2 as the 23,570 non-face patterns which were most similar to faces. An SVM with a 2nd-degree polynomial kernel was trained on the reduced feature sets. The ROC curves are shown in Fig. 4 and 5 for PCA and ILC respectively. The first 3 ILC features were superior to the first 3 PCA features. However, increasing the number of ILC features up to 10 did not improve the performance. This is because ILC does not generate uncorrelated features. Indeed, the 10 ILC features were highly correlated with an average correlation of about 0.7. Increasing the number of PCA features up to 20, on the other hand, steadily improved the classification performance until it equaled the performance of the system trained on the original 283 features. Reducing the number of features to 20 sped-up the classification by a factor of $14^2 = 196$ for a 2nd-degree polynomial SVM.

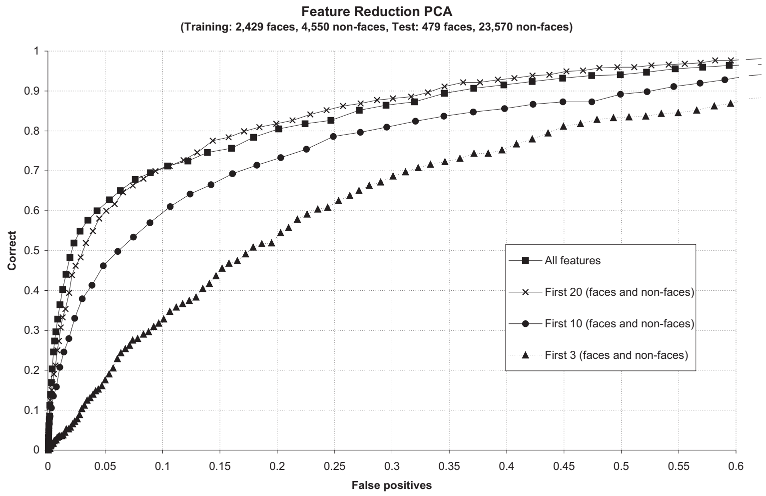

Figure 4: ROC curves for SVMs with 2nd-degree polynomial kernel trained on PCA features. The PCA has been calculated on the whole training set.

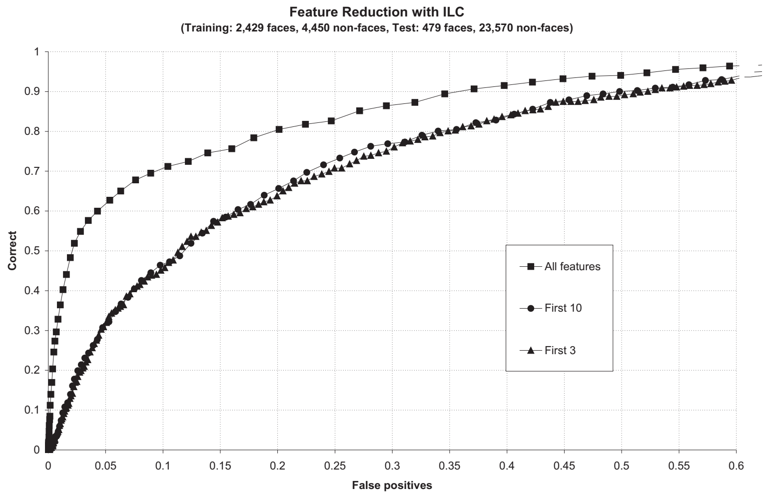

Figure 5: ROC curves for SVMs with 2nd-degree polynomial kernel trained on feature sets generated by ILC.

3.2 Selecting features

We developed a technique for selecting class relevant features based on the decision function $f(\mathbf{x})$ of an SVM:

$$f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x_i}) + b, \tag{1}$$

where $\mathbf{x}_i$ are the Support Vectors, α_i the Lagrange multipliers, y_i the labels of the Support Vectors (-1 or 1), $K(\cdot,\cdot)$ the kernel function, and b a constant. A point $\mathbf{x}$ is assigned to class 1 if $f(\mathbf{x}) > 0$, otherwise to class -1. The kernel function $K(\cdot,\cdot)$ defines the dot product in some feature space F^*. If we denote the transformation from the original feature space F to F^* by $\Phi(\mathbf{x})$, Eq. (1) can be rewritten as:

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b, \tag{2}$$

where $\mathbf{w} = \sum_i \alpha_i y_i \Phi(\mathbf{x_i})$. Note that the decision function in Eq. (2) is linear on the transformed features $\mathbf{x}^* = \Phi(\mathbf{x})$. For a 2nd-degree polynomial kernel with $K(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}\cdot\mathbf{y})^2$, the transformed feature space F^* with dimension $N^* = \frac{(N+3)N}{2}$ is given by $\mathbf{x}^* = (\sqrt{2}x_1, \sqrt{2}x_2, .., \sqrt{2}x_N, x_1^2, x_2^2, .., x_N^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, .., \sqrt{2}x_{N-1}x_N)$.

The contribution of a feature x_n^* to the decision function in Eq. (2) depends on w_n. A straightforward way to order the features is by decreasing $|w_n|$. Alternatively, we weighted $\mathbf{w}$ by the Support Vectors to account for different distributions of the features in the training data. The features were ordered by decreasing $|w_n \sum_i y_i x_{i,n}^*|$, where $x_{i,n}^*$ denotes the n-th component of Support Vector i in feature space F^*. Both ways of feature ranking were applied to an SVM with 2nd-degree polynomial kernel trained on 20 PCA features corresponding to 230 features in F^*. In a first evaluation of the rankings we calculated $\frac{1}{M}\sum_i |f(\mathbf{x_i}) - f_S(\mathbf{x_i})|$ for all M Support Vectors, where $f_S(\mathbf{x})$ is the decision function using the S first features according to the ranking. Note, that we did not retrain the SVM on the reduced feature set. The results in Fig. 6 show that ranking by the weighted components of $\mathbf{w}$ lead to a faster convergence of the error towards 0. The final evaluation was done on the test set. Fig. 7 shows the ROC curves for 50, 100, and 150 features for both ways of ranking. The results confirm that ranking by the weighted components of $\mathbf{w}$ is superior. The ROC curve for 100 features on the test set was about the same as for the complete feature set.

By combining PCA with the above described feature selection we could reduce the originally $\frac{(283+3)283}{2} = 40,469$ features in F^* to 100 features without loss in classification performance on the test set.

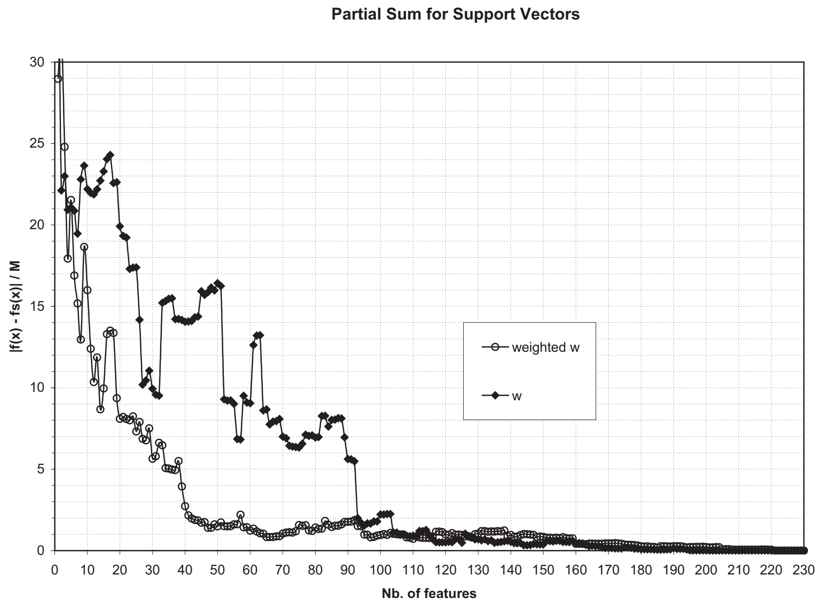

Figure 6: Classifying Support Vectors with a reduced number of features. The x-axis shows the number of features, the y-axis is the mean absolute difference between the output of the SVM using all features and the same SVM using the S first features only. The features were ranked according to the components and the weighted components of the normal vector of the separating hyperplane.

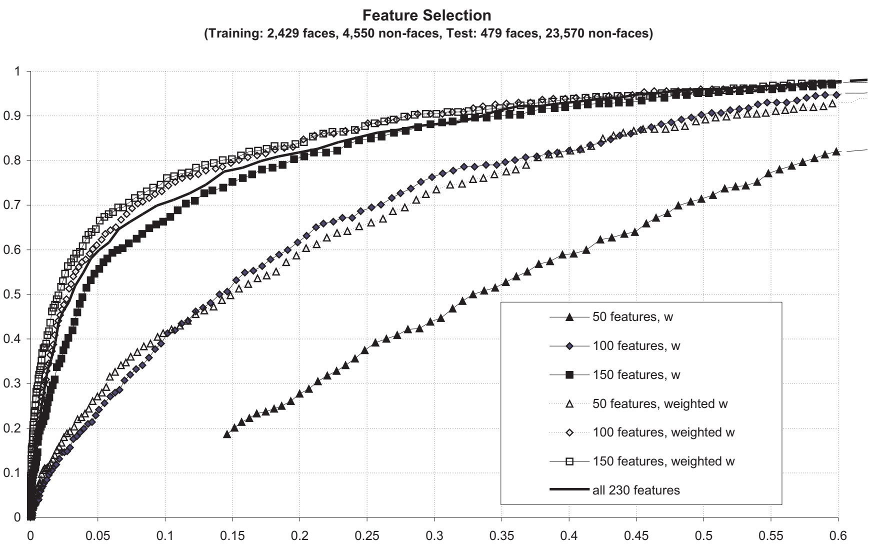

Figure 7: ROC curves for reduced feature sets.

4 Parameterization of SVMs

The choice of the classifier and its parameterization play an important role in the overall performance of a learning-based system. We chose the SVM as classifier since it is well founded in statistical learning theory [Vapnik 98] and has been successfully applied to various object detection tasks in computer vision [Oren et al. 97, Osuna 98]. An SVM is parameterized by its kernel function and the C value which determines the constraint violations during the training process. For more detailed information about SVMs refer to [Vapnik 98].

4.1 Kernel function

Three common types of kernel functions were evaluated in our experiments:

- Linear kernel: $K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$
- Polynomial kernel: $K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^n$, n was set to 2 and 3.
- Gaussian kernel: $K(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2})$, σ^2 was set to 5 and 10.

All experiments were carried out on the training and test sets described in Chapter 3. The ROC curves are shown in Fig. 8. The 2nd-degree polynomial kernel seems a good compromise between computational complexity and classification performance. The SVM with Gaussian kernel ($\sigma^2 = 5$) was slightly better but required about 1.5 times more Support Vectors (738 versus 458) than the polynomial SVM.

4.2 C-parameter

We varied C between 0.1 and 100 for an SVM with 2nd-degree polynomial kernel. Some results are shown in Fig. 9. The detection performance slightly increases with C until $C = 1$. For $C \geq 1$ the error rate on the training data was 0 and the decision boundary did not change any more.

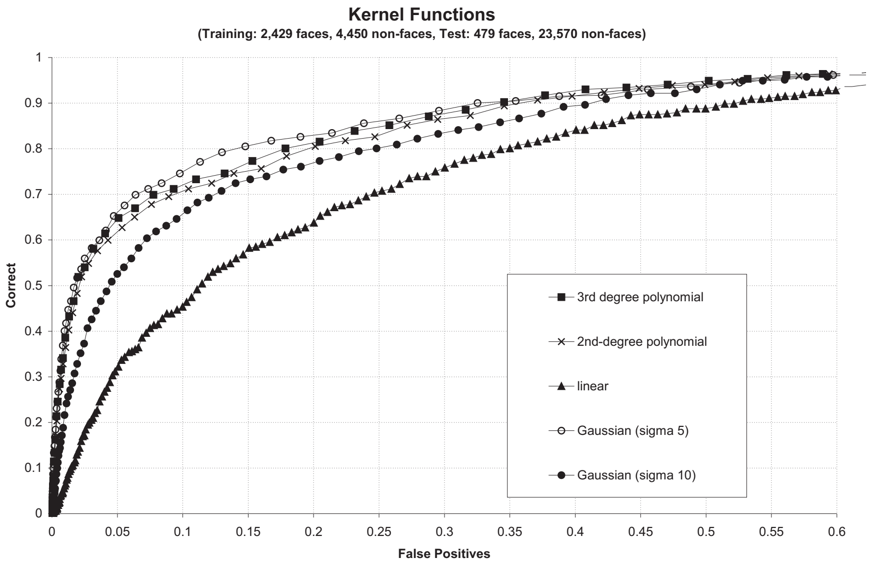

Figure 8: ROC curves for different kernel functions.

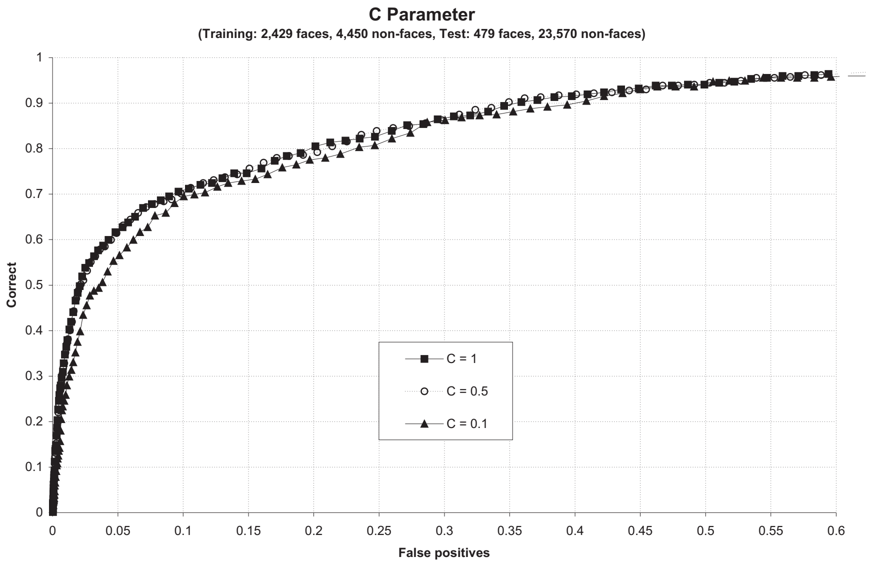

Figure 9: ROC curves for different values of C.

5 Training Data

Besides selecting the input features and the classifier, choosing the training data is the third important step in developing a classification system.

5.1 Positive training data

Extracting face patterns is usually a tedious and time-consuming work that has to be done manually. An interesting alternative is to generate artificial samples for training the classifier [Niyogi et al. 98]. In [Rowley et al. 97, Schneiderman & Kanade 98] the training set was enlarged by applying various image transformation to the original face images. We went a step further and generated a completely synthetic set of images by rendering 3-D head models [Vetter 98]. Using 3-D models for training has two interesting aspects: First, illumination and pose of the head are fully controllable and second, images can be generated automatically in large numbers by rendering the 3-D models. To create a large variety of synthetic face patterns we morphed between different head models and modified the pose and the illumination. Originally we had 7 textured head models acquired by a 3-D scanner. Additional head models were generated by 3-D morphing between all pairs of the original models. The heads were rotated between $-15°$ and $15°$ in azimuth and between $-8°$ and $8°$ in the image plane. The faces were illuminated by ambient light and a single directional light pointing towards the center of the face. The position of the light varied between $-30°$ and $30°$ in azimuth and between $30°$ and $60°$ in elevation. Overall, we generated about 5,000 face images. The negative training set was the same as in Chapter 3. Some examples of real and synthetic faces from our training sets are shown in Fig. 10. The ROC curves for SVMs trained on real and synthetic data are shown in Fig. 11. The significant difference in performance indicates that the image variations captured in the synthetic data do not cover the variations present in real face images. Most likely because our face models were too uniform: No people with beard, no differences in facial expression, no differences in skin color.

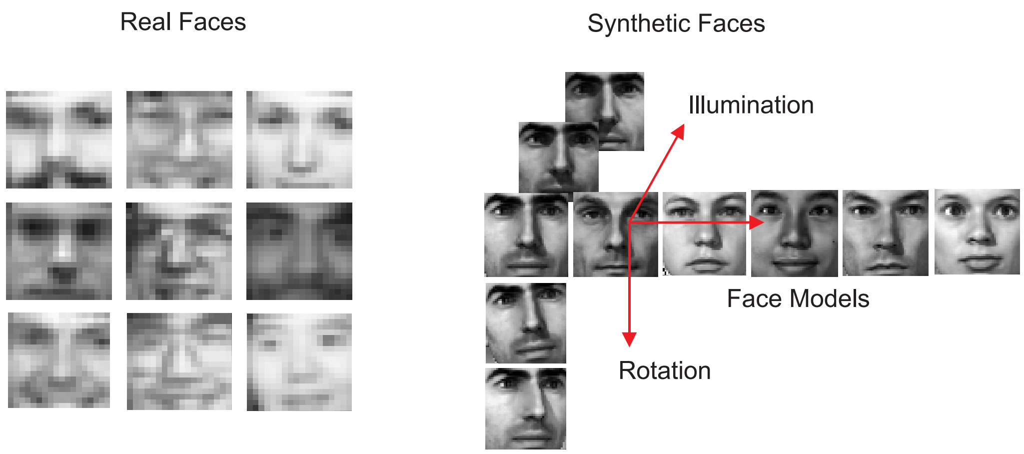

Figure 10: Examples of real and synthetic face images. The synthetic faces were generated by rendering 3-D head models under varying pose and illumination. The resolution of the synthetic faces was 50×50 pixels after rendering. For training the face detector we rescaled the images to 19×19 pixels.

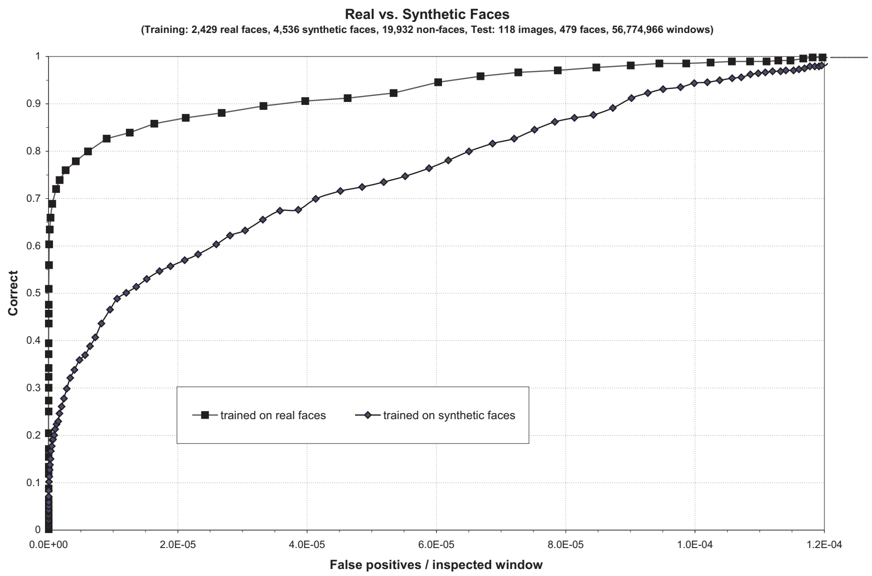

Figure 11: ROC curves for classifiers trained on real and synthetic faces.

5.2 Negative training data

Non-face patterns are abundant and can be automatically extracted from images that do not contain faces. However, it would require a huge number of randomly selected samples to fully cover the variety of non-face patterns. Iterative bootstrapping of the system with false positives (FPs) is a way to keep the training set reasonably small by specifically picking non-face patterns that are useful for learning. Fig. 12 shows the ROC curves for an SVM trained on the 19,932 randomly selected non-face patterns and an SVM trained on additional 7,065 non-face patterns determined in three bootstrapping iterations. At 80% detection rate the FP rate for the bootstrapped system was about $2 \cdot 10^{-6}$ per classified pattern which corresponds to 1 FP per image. Without bootstrapping, the FP rate was about 3 times higher.

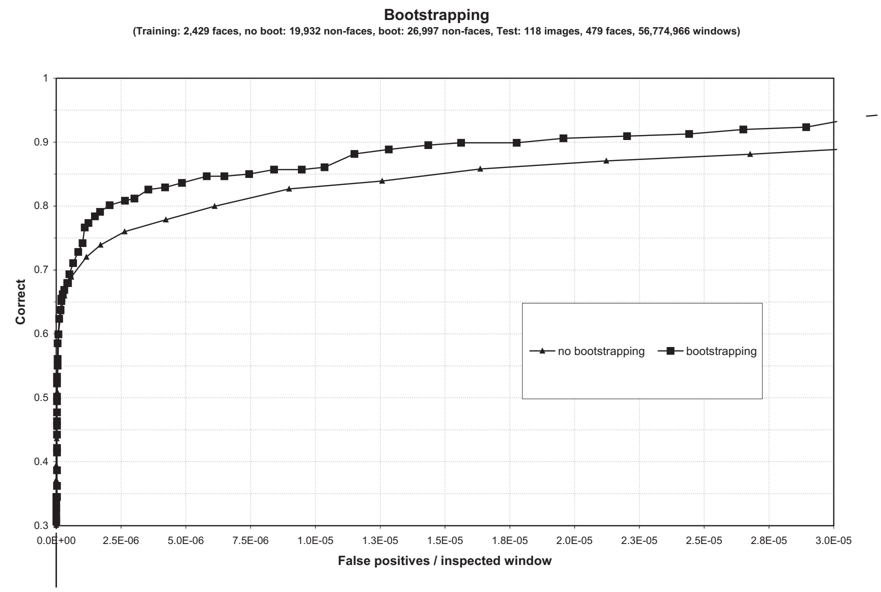

Figure 12: ROC curves for a classifier which was trained on 19,932 randomly selected non-face patterns and for a classifier which was bootstrapped with 7,065 additional non-face patterns.

6 Results and comparison to other face detection systems

There are two sets of gray images provided by the CMU [Rowley et al. 98] which are commonly used for evaluating face detection systems [Sung 96, Yang et al. 99, Osuna 98, Rowley et al. 98, Schneiderman & Kanade 98]. These test sets provide a good basis for comparisons between face detection systems. However, the use of different training data and different heuristics for suppressing false positives complicates comparisons.

To achieve competitive detection results we further enlarged the previously used positive and negative training sets and also implemented heuristics for suppressing multiple detections at nearby image locations. An SVM with 2nd-degree polynomial kernel was trained on histogram equalized 19×19 images of 10,038 faces and 36,220 non-faces. The positive training set consisted of 5,813 real faces and 4,225 synthetic faces. The synthetic faces were generated from a subset of the real faces by rotating them between -2° and 2° and changing their aspect ratio between 0.9 and 1.1. The negative training set was generated from an initial set of 19,932 randomly selected non-face patterns and additional 16,288 non-face patterns determined in six bootstrapping iterations. For testing, each test image was rescaled 14 times by factors between 0.1 and 1.2. A 19×19 window was shifted pixel-by-pixel over each image. We applied two heuristics to remove multiple detections at nearby image locations. First, a detection was suppressed if there was at least one detection with a higher SVM output value in its neighborhood. The neighborhood in the image plane was defined as a 19×19 box around the center of the detection. The neighborhood in the scale space was set to $[0.5, 2]$. The second heuristic counted the number of detections within the neighborhood. If there were less than three detections, the detection was suppressed. The results of our classifier are shown in Fig. 13 and compared to other results in Table 1. Our system outperforms a previous SVM-based face detector [Osuna 98] due to a larger training set and improvements in suppressing multiple detections. The results achieved by the naïve Bayes classifier [Schneiderman & Kanade 98] and the SNoW-based face detector [Yang et al. 99] are better than our results. However, it is not clear which heuristics were used in these systems to suppress multiple detections and how these heuristics affected the results.

System	Subset of test set 1 23 images, 155 faces		Test set 1 130 images, 507 faces	
	Det. Rate	FPs	Det. Rate	FPs
[Sung 96] Neural Network	84.6%	13	N/A	N/A
[Osuna 98] SVM	74.2%	20	N/A	N/A
[Rowley et al. 98] Single neural network	N/A	N/A	90.9%	738
[Rowley et al. 98] Multiple neural networks	84.5%	8	84.4%	79
[Schneiderman & Kanade 98][3] Naïve Bayes	91.1%	12	90.5%	33
[Yang et al. 99][4] SNoW, multi-scale	94.1%	3	94.8%	78
Our system[5]	84.7% 90.4%	11 26	85.6% 89.9%	9 75

Table 1: Comparison between face detection systems.

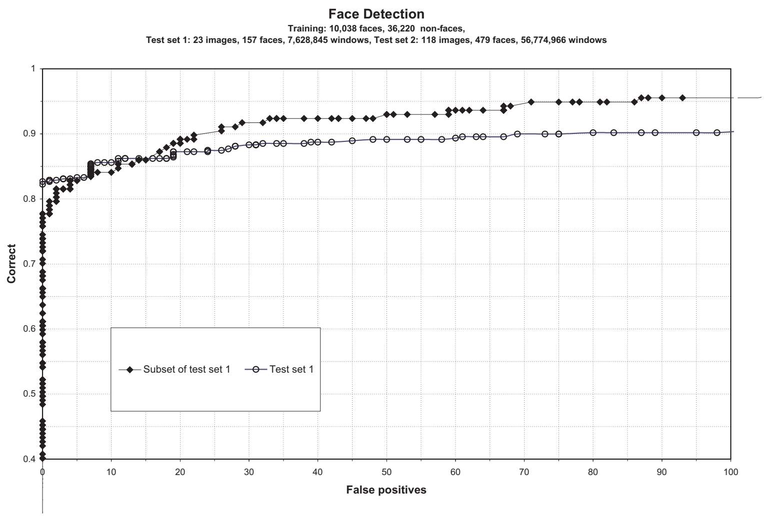

Figure 13: ROC curves for bootstrapped classifier with heuristics for suppressing multiple detections.

[3]Five images of hand-drawn faces were excluded from test set 1.

[4]Images of hand-drawn faces and cartoon faces were excluded from test set 1.

[5]Twelve images containing line-drawn faces, cartoon faces and non-frontal faces were excluded from test set 1.

7 Component-based face detection

7.1 Motivation

Until now we considered systems where the whole face pattern was classified by a single SVM. Such a global approach is highly sensitive to changes in the pose of an object. Fig. 14 illustrates the problem for the simple case of linear classification. The result of training a linear classifier on frontal faces can be represented as a single face template, schematically drawn in Fig. 14 a). Even for small rotations the template clearly deviates from the rotated faces as shown in Fig. 14 b) and c). The component-based approach tries to avoid this problem by independently detecting parts of the face. In Fig. 15 the eyes, nose, and the mouth are represented as single templates. For small rotations the changes in the components are small compared to the changes in whole face pattern. Slightly shifting the components is sufficient to achieve a reasonable match with the rotated faces.

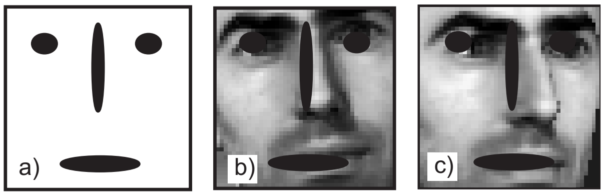

Figure 14: Matching with a single template. The schematic template of a frontal face is shown in a). Slight rotations of the face in the image plane b) and in depth c) lead to considerable discrepancies between template and face.

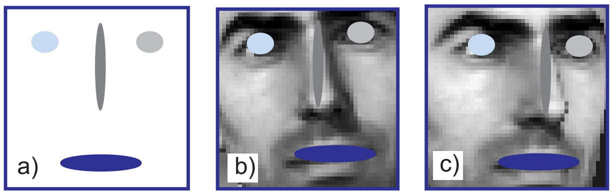

Figure 15: Matching with a set of component templates. The schematic component templates for a frontal face are shown in a). Shifting the component templates can compensate for slight rotations of the face in the image plane b) and in depth c).

7.2 Component-based classifier

An overview of our two-level component-based classifier is shown in Fig. 16. A similar architecture was used for people detection [Mohan 99]. On the first level, component classifiers independently detect the eyes (9×7 pixels), the nose (9×11 pixels) and the mouth (13×7 pixels). Each component classifier was trained on a set of manually extracted facial components and a set of randomly selected non-face patterns. The components were extracted from the same set of 2,429 real face images as used in previous experiments.

On the second level the geometrical configuration classifier performs the final face detection by combining the results of the component classifiers. Given a 19×19 image, the maximum outputs of the eyes, nose, and mouth classifiers within rectangular search regions[6] 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 standard deviation of the components' locations in the training images.

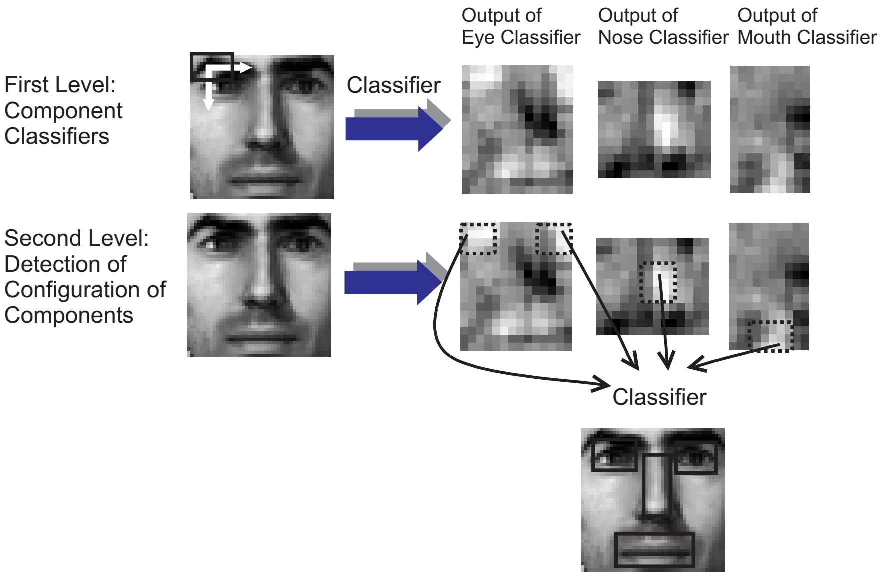

Figure 16: System overview of the component-based classifier. On the first level, windows of the size of the components (solid lined boxes) are shifted over the face image and classified by the component classifiers. On the second level, the maximum outputs of the component classifiers within predefined search regions (dotted lined boxes) are fed into the geometrical configuration classifier.

[6]To account for changes in the size of the components, the outputs were determined over multiple scales of the input image. In our tests, we set the range of scales to $[0.75, 1.2]$.

The ROC curves for CMU test set 1 are shown in Fig. 17. The component classifiers were SVMs with 2nd-degree polynomial kernels and the geometrical configuration classifier was a linear SVM[7]. Up to about 90% recognition rate, the four component system performs worse than the whole face classifier. Probably due to class-relevant parts of the face that were not covered by the four components. Therefore, we added the whole face as a fifth component similar to the template-based face recognition system proposed in [Brunelli & Poggio 93]. As shown in Fig. 17 the five component classifier performs similar to the whole face classifier. This indicates that the whole face is the most dominant of the five components.

To check the robustness of the classifiers against object rotations we performed tests on synthetic faces generated from 3-D head models. The synthetic test set consisted of two groups of 19×19 face images: 4,574 faces rotated in the image plane, and 15,865 faces rotated in depth. At each rotation angle we determined the FP rate for 90% detection rate based on the ROC curves in Fig. 17. The results in Fig. 18 and 19 show that the best performance was achieved by the five component system. However, it deteriorated much faster with increasing rotation than the four component system. This is not surprising since the whole face pattern changes more under rotation than the patterns of the other components.

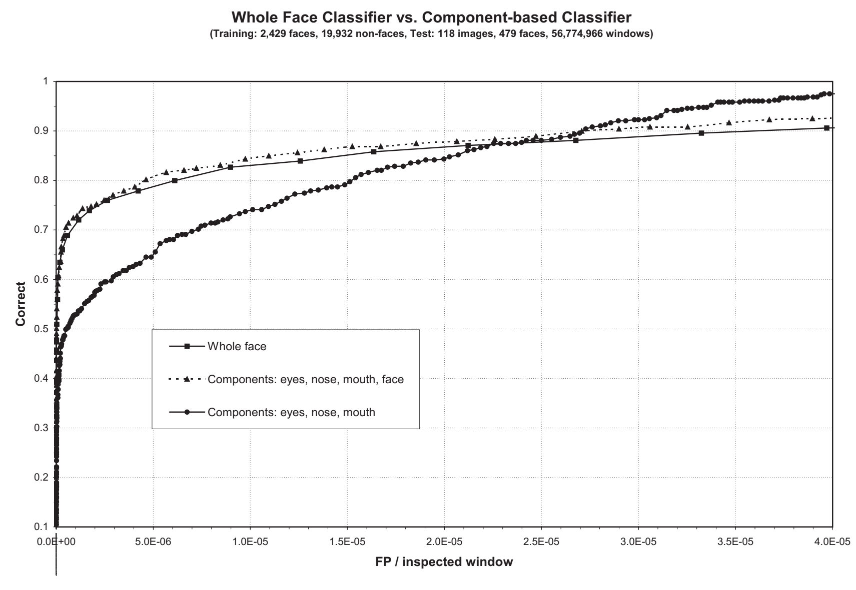

Figure 17: ROC curves for frontal faces.

[7]Alternatively we tried linear classifiers for the components and a polynomial kernel for the geometrical classifier but the results were clearly worse.

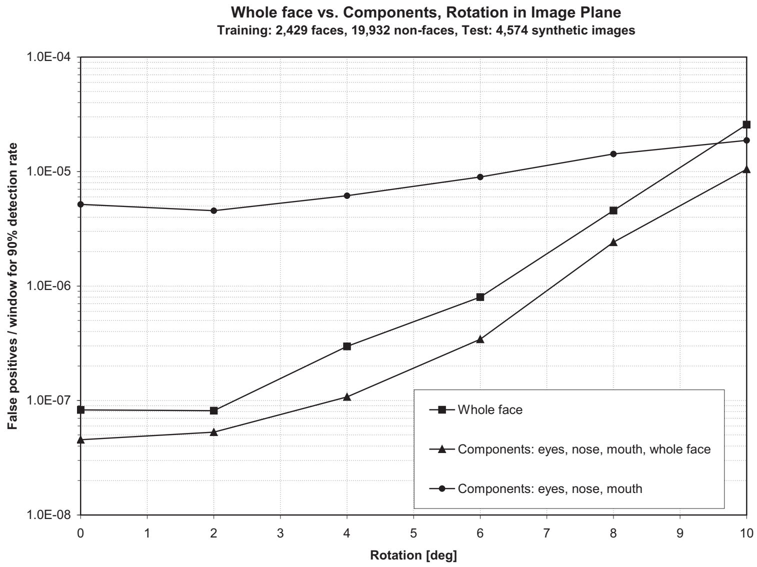

Figure 18: Classification results for synthetic faces rotated in the image plane.

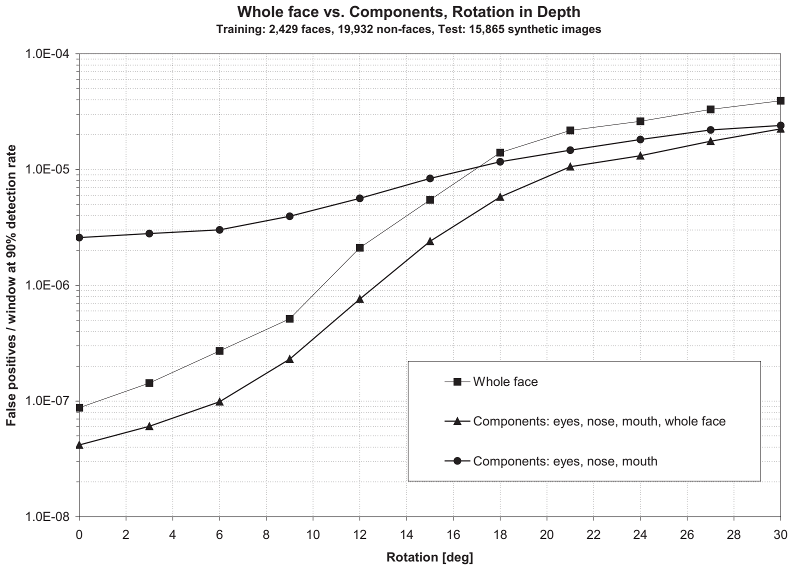

Figure 19: Classification results for synthetic faces rotated in depth.

7.3 Determining the components: preliminary results

In our previous experiments we manually selected the eyes, the nose and the mouth as characteristic components of a face. Although this choice is somehow obvious, it would be more sensible to choose the components automatically based on their discriminative power and their robustness against pose changes. Moreover, for objects other than faces, it might be difficult to manually define a set of meaningful components. In the following we present two methods for learning components from examples.

The first method arbitrarily defines components and lets the geometrical configuration classifier learn to weight the components according to their relevancy. We carried out an experiment with 16 non-overlapping components of size 5×5 evenly distributed on the 19×19 face pattern (see Fig. 20). As in previous experiments the component classifiers were SVMs with 2nd-degree polynomial kernels and the geometrical configuration classifier was a linear SVM. The training errors of the component classifiers give information about the discriminative power of each component (see Fig. 21). The components 5, 8, 9, and 12 are located on the cheeks of the face. They contain only few gray value structures which is reflected in the comparatively high error rates. Surprisingly, the components 14 and 15 around the mouth also show high error rates. This might be due to variations in the facial expression and slight misalignments of the faces in the training set.

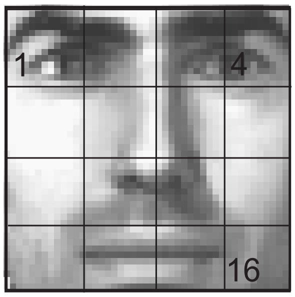

Figure 20: Partitioning the face pattern into 16 non-overlapping components.

An alternative to using a large set of arbitrary components is to specifically generate discriminative components. Following this idea, we developed a second method that automatically determines rectangular components in a set of synthetic face images. The algorithm starts with a small rectangular component located around a pre-selected point in the face (e.g. center of the left eye)[8]. The component is extracted from all synthetic face images to build a training set of positive examples.

[8]We could locate the same facial point in all face images since we knew the point-by-point correspondences between the 3-D head models.

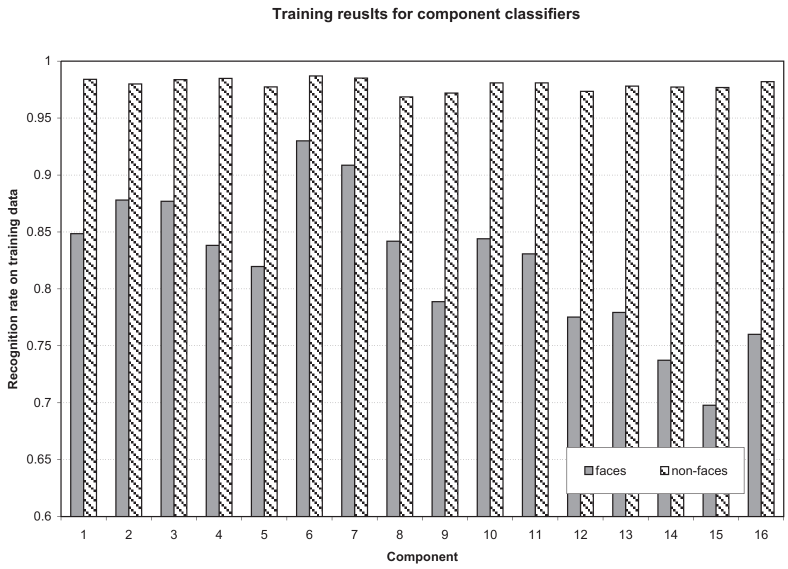

Figure 21: Training results for the 16 component classifiers.

We also generate a training set of non-face patterns that have the same rectangular shape as the component. After training an SVM on the component data we estimate the performance of the SVM according to its leave-one-out error [Vapnik 98]:

$$\rho = R^2 \mathbf{w}^2, \tag{3}$$

where R is the radius of the smallest sphere[9] in the feature space F^* containing the Support Vectors, and $\mathbf{w}^2$ is the square norm of the coefficients of the SVM (see Eq. (2)). After determining ρ we enlarge the component by expanding the rectangle by one pixel into one of four directions (up, down, left, right). Again, we generate training data, train an SVM and determine ρ. We keep the expansion if it lead to a decrease in ρ else it is rejected and an expansion into one of the remaining directions was tried. This process is continued until the expansions into all four directions lead to an increase of ρ. In a preliminary experiment we applied the algorithm to three 3×3 regions located at the center of the eye, tip of the nose and center of the mouth. The final components are shown in Fig. 22, they were determined on about 4,500 synthetic faces (65×85 pixels, rotation in depth between $-45°$ and $45°$). The

[9] In our experiments we replaced R^2 in Eq. (3) by the dimensionality N of space F^*. This because our data points lay within an N-dimensional cube of length 1, so the smallest sphere containing the data had radius equal to $\sqrt{N}/2$. This approximation was mainly for computational reasons as in order to compute R we need to solve an optimization problem [Osuna 98].

eyes (24×8 pixels) and mouth (30×12 pixels) are similar to the manually selected components. The component located at the tip of the nose (6×4 pixels), however, is small. This indicates that the pattern around the tip of the nose strongly varies under rotation.

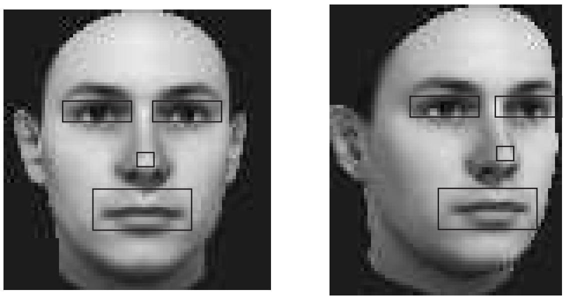

Figure 22: Automatically generated components.

8 Conclusion and future work

We presented and compared two systems for frontal and near-frontal face detection: a whole face detection system and a component-based detection system. Both systems are trained from examples and use SVMs as classifiers. The first system detects the whole face pattern with a single SVM. In contrast, the component-based system performs the detection by means of a two level hierarchy of classifiers. On the first level, the component classifiers independently detect parts of the face, such as eyes, nose, and mouth. On the second level, the geometrical configuration classifier combines the results of the component classifiers and performs the final detection step. In addition to the whole face and component-based face detection approaches we presented a number of experiments on image feature selection, feature reduction and selection of training data. The main points of the paper are as follows:

- Gray values are better input features for a face detector than are Haar wavelets and gradient values.
- By combining PCA- with SVM-based feature selection we sped-up the detection system by two orders of magnitude without loss in classification performance.
- Bootstrapping the classifier with non-face patterns increased the detection rate by more than 5%.
- We developed a component-based face detector which is more robust against face rotations than a comparable whole face detector.

- We proposed a technique for learning characteristic components from examples.

We have shown that a component-based classifier trained on frontal faces can deal with slight rotations in depth. The next logical step is to cover a larger range of pose changes by training the component classifiers on rotated faces. Another promising topic for further research is learning a geometrical model of the face by adding the image locations of the detected components to the input features of the geometrical configuration classifier.

References

[Beymer 93] D. J. Beymer. *Face recognition under varying pose.* A.I. Memo 1461, Center for Biological and Computational Learning, M.I.T., Cambridge, MA, 1993.

[Brunelli & Poggio 93] R. Brunelli, T. Poggio. *Face Recognition: Features versus Templates.* IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1993) 1042–1052.

[Ivanov et al. 98] Y. Ivanov, A. Bobick, J. Liu. *Fast lighting independent background subtraction.* Proc. IEEE Workshop on Visual Surveillance, 1998, 49–55.

[Jebara & Pentland 97] T. Jebara, A. Pentland. *Parametrized structure from motion for 3D adaptive feedback tracking of faces.* Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Juan, 1997, 144–150.

[Jones & Rehg 99] M. J. Jones, J. M. Rehg. *Statistical color models with application to skin detection.* Proc. IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, 1999, 274–280.

[Leung et al. 95] T. K. Leung, M. C. Burl, P. Perona. *Finding faces in cluttered scenes using random labeled graph matching.* Proc. International Conference on Computer Vision, 1995, 637–644.

[Mohan 99] A. Mohan. *Object detection in images by components.* A.I. Memo 1664, Center for Biological and Computational Learning, M.I.T., Cambridge, MA, 1999.

[Niyogi et al. 98] P. Niyogi, F. Girosi, T. Poggio. *Incorporating prior information in machine learning by creating virtual examples.* Proceedings of the IEEE 86 (1998) 2196–2209.

[Oren et al. 97] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, T. Poggio. *Pedestrian detection using wavelet templates.* IEEE Conference on Computer Vision and Pattern Recognition, San Juan, 1997, 193–199.

[Osuna 98] E. Osuna. *Support Vector Machines: Training and Applications.* Ph.D. thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, MA, 1998.

[Rikert et al. 99] T. D. Rikert, M. J. Jones, P. Viola. *A cluster-based statistical model for object detection.* Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1999, 1046–1053.

[Rowley et al. 97] H. A. Rowley, S. Baluja, T. Kanade. *Rotation Invariant Neural Network-Based Face Detection.* Computer Scienct Technical Report CMU-CS-97-201, CMU, Pittsburgh, 1997.

[Rowley et al. 98] H. A. Rowley, S. Baluja, T. Kanade. *Neural Network-Based Face Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 23–38.

[Saber & Tekalp 96] E. Saber, A. Tekalp. *Face detection and facial feature extraction using color, shape and symmetry based cost functions.* Proc. International Conference on Pattern Recognition, Vol. 1, Vienna, 1996, 654–658.

[Schneiderman & Kanade 98] H. Schneiderman, T. Kanade. *Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition.* Proc. IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, 1998, 45–51.

[Sung 96] K.-K. Sung. *Learning and Example Selection for Object and Pattern Recognition.* Ph.D. thesis, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA, 1996.

[Toyama et al. 99] K. Toyama, J. Krumm, B. Brumitt, B. Meyers. *Wallflower: principles and practice of background maintenance.* Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1999, 255–261.

[Vapnik 98] V. Vapnik. *Statistical learning theory.* New York: John Wiley and Sons, 1998.

[Vetter 98] T. Vetter. *Synthesis of novel views from a single face.* International Journal of Computer Vision 28 (1998) 103–116.

[Wiskott 95] L. Wiskott. *Labeled Graphs and Dynamic Link Matching for Face Recognition and Scene Analysis.* Ph.D. thesis, Ruhr-Universität Bochum, Bochum, Germany, 1995.

[Wu et al. 99] H. Wu, Q. Chen, M. Yachida. *Face detection from color images using a fuzzy pattern matching method.* IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 557–563.

[Yang et al. 99] M.-H. Yang, D. Roth, N. Ahuja. *A SNoW-based face detector.* Advances in Neural Information Processing Systems 12, 1999.