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

A.I. Memo No. 1697
C.B.C.L Paper No. 192

September, 2000

Feature Selection for Face Detection

Thomas Serre, Bernd Heisele, Sayan Mukherjee, Tomaso Poggio

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu. The pathname for this publication is: ai-publications/1500-1999/1697

Abstract

We present a new method to select features for a face detection system using Support Vector Machines (SVMs). In the first step we reduce the dimensionality of the input space by projecting the data into a subset of eigenvectors. The dimension of the subset is determined by a classification criterion based on minimizing a bound on the expected error probability of an SVM. In the second step we select features from the SVM feature space by removing those that have low contributions to the decision function of the SVM.

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 Contract No. N00014-93-1-3085, Office of Naval Research Contract No. N00014-95-1-0600, National Science Foundation Contract No. IIS-9800032, and National Science Foundation Contract No. DMS-9872936. Additional support is provided by: AT&T, Central Research Institute of Electric Power Industry, Eastman Kodak Company, DaimlerChrysler, Digital Equipment Corporation, Honda R&D Co., Ltd., NEC Fund, Nippon Telegraph & Telephone, and Siemens Corporate Research, Inc.

1 Introduction

The trainable system for detecting frontal and near-frontal views of faces in gray images presented in [Heisele et al. 2000] gave good results in terms of detection rates. The system used gray values of 19×19 images as inputs to a second-degree polynomial kernel SVM. This choice of kernel lead to more than 40,000 features in the feature space[1]. Searching an image for faces at different scales took several minutes on a PC. Many real-world applications require significantly faster algorithms. One way to speed-up the system is to reduce the number of features.

We present a new method to reduce the dimensions of both input and feature space without decreasing the classification rate. The problem of choosing the subset of input features which minimizes the expected error probability of the SVM is an integer programming problem, known to be NP-complete. To simplify the problem, we first rank the features and then select their number by minimizing a bound on the expected error probability of the classifier.

The outline of the paper is as follows: generating training and test data is described in Chapter 2. In Chapter 3 we give a brief overview of SVM theory. In Chapter 4 we rank features in the input space according to a classification criterion. We then determine the appropriate number of ranked features in Chapter 5. In Chapter 6 we remove features from the feature space that have small contributions to the decision function of the classifier. In Chapter 7 we applied feature selection to a real-world application.

2 Description of the Input Data

2.1 Input features

In this section we describe the pre-processing steps applied to the gray images in order to extract the input features to our classifier. To decrease the variations caused by changes of illumination we used three preprocessing steps proposed in [Sung 96]. A mask was first applied to eliminate pixels close to the boundary of the 19×19 images, reducing the number of pixels from 361 to 283. To account for cast shadows we subtracted a best-fit intensity plane from the images. Then we performed histogram equalization to remove variations in the image brightness and contrast. Finally the 283 gray values were re-scaled to a range between 0 and 1. We also computed the gray value gradients from the histogram equalized images using 3×3 x- and y-Sobel Filters. Again the results were re-scaled to be in a range between 0 and 1. These

[1] In the following, we use *input space* $\mathbb{R}^n$ for the representation space of the image data and *feature space* $\mathbb{R}^p$ $(p > n)$ for the non-linearly transformed input space.

gradient features were combined with the gray value features to form a second set of 572 features[2]. Additionally we applied Principal Component Analysis (PCA) to the whole training set and projected the data points into the eigenvector space.

To summarize we considered four different sets of input features:

- 283 gray features
- 572 gray/gradient features
- 283 PCA gray features
- 572 PCA gray/gradient features

2.2 Training and test sets

In our experiments we used one training and two test sets. The positive training set contained 2,429 19×19 faces. The negative training set contained 4,548 randomly selected non-faces patterns.

In the first part of this paper, we used a small test set in order to perform a large number of tests. The test set was extracted from the CMU test set 1[3]. We extracted all 479 faces and 23,570 non-face patterns. The non-face patterns were selected by a linear SVM classifier as the non-face patterns most similar to faces. The final evaluation of our system was performed on the entire CMU test set 1, containing 118 images. Processing all images at different scales resulted in about 57,000,000 analyzed 19×19 windows.

3 Support Vector Machine

Support Vector Machines [Vapnik 98] perform pattern recognition for two-class problems by finding the decision surface which minimizes the structural risk of the classifier. This is equivalent to determining the separating hyperplane that has maximum distance to the closest points of the training set. These closest points are called *Support Vectors* (SVs). Figure 1 (a) shows a 2-dimensional problem for linearly separable data. The gray area indicates all possible hyperplanes which separate the two classes. The optimal hyperplane in Figure 1 (b) maximizes the distance to the SVs.

[2] As reported in [Heisele et al. 2000], detection results with gradient alone were worse than those for gray values. That is why we combined gradient and gray features.

[3] 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.

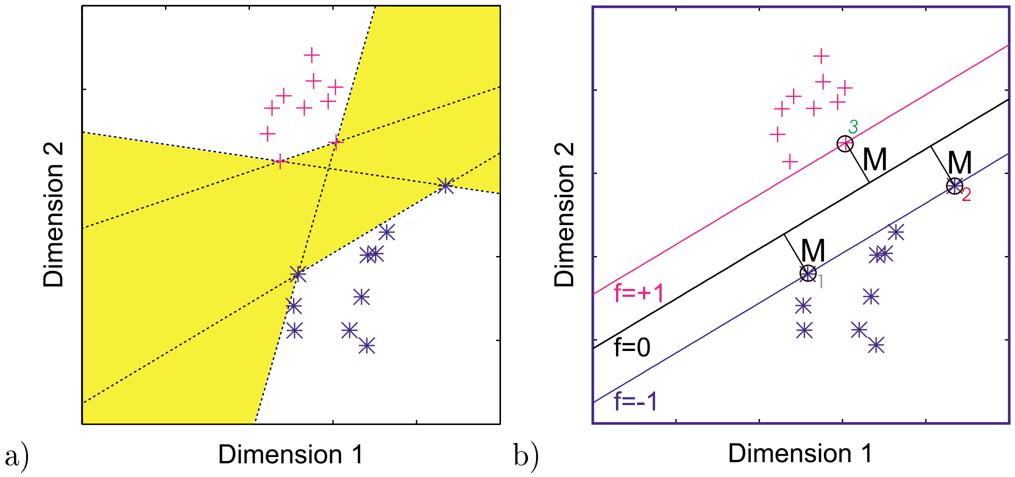

Figure 1: a) The gray area shows all possible hyperplanes which separate the two classes. b) The optimal hyperplane maximizes the distance to the closest points. These points (1, 2 and 3) are called Support Vectors (SVs). The distance M between the hyperplane and the SVs is called the margin.

If the data are not linearly separable in the input space: a non-linear transformation $\Phi(\cdot)$ maps the data points $\mathbf{x}$ of the *input space* $\mathrm{I\!R}^n$ into a high dimensional, called *feature space* $\mathrm{I\!R}^p$ $(p > n)$. The mapping $\Phi(\cdot)$ is represented in the SVM classifier by a kernel function $K(\cdot, \cdot)$ which defines an inner product in $\mathrm{I\!R}^p$. The decision function of the SVM is thus:

$$f(\mathbf{x}) = w \cdot \Phi(\mathbf{x}) + b = \sum_i \alpha_i^0 y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{1}$$

where y_i is the class label $\{-1, 1\}$ of the training samples. Again the optimal hyperplane is the one with the maximal distance (in feature space $\mathrm{I\!R}^p$) to the closest points $\Phi(\mathbf{x}_i)$ of the training data. Determining that hyperplane leads to maximizing the following functional with respect to α:

$$W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \tag{2}$$

under constraints $\sum_{i=1}^{\ell} \alpha_i y_i = 0$ and $C \geq \alpha_i \geq 0, \quad i = 1, ..., \ell$. The solution of this maximization problem is denoted $\alpha^0 = (\alpha_1^0, ..., \alpha_k^0, ..., \alpha_l^0)$.

An upper bound on the expected error probability EP_{err} of an SVM classifier is given by:

$$EP_{err} \leq \frac{1}{\ell}\, E\left(R^2\, W(\alpha^0)\right) \tag{3}$$

where R is the radius of the smallest sphere including all points $\Phi(\mathbf{x}_1), ..., \Phi(\mathbf{x}_\ell)$ of the training vectors $\mathbf{x}_1, ..., \mathbf{x}_\ell$. In the following, we will use this bound of the expectation of the leave-one-out-error to rank and select features.

4 Ranking Features in the Input Space

4.1 Description of the method

In [Weston et al. 2000] a gradient descent method is proposed to rank the input features by minimizing the bound of the expectation of the leave-one-out error of the classifier. We implemented an earlier approximation of this approach. The main idea is to re-scale the n-dimensional input space by a $n \times n$ diagonal matrix σ such that the margin M in Equation (3) is maximized. However, one can trivially increase the margin by simply multiplying all input vectors by a scalar. For this reason the following constraint is added $||\sigma||_F = N$, where N is some constant. This constraint approximately enforces the norm of radius R around the data to be constant while maximizing the margin. The new mapping function can be written as $\Phi_\sigma(\mathbf{x}) = \Phi(\sigma \cdot \mathbf{x})$ and the kernel function is $K_\sigma(\mathbf{x}, \mathbf{y}) = K(\sigma \cdot \mathbf{x}, \sigma \cdot \mathbf{y}) = (\Phi_\sigma(\mathbf{x}) \cdot \Phi_\sigma(\mathbf{x}))$. The decision function given in Equation (1) becomes:

$$f(\mathbf{x}, \sigma) = w \cdot \Phi_\sigma(\mathbf{x}) + b = \sum_i \alpha_i^0 y_i K_\sigma(\mathbf{x}_i, \mathbf{x}) + b \tag{4}$$

The maximization problem of Equation (2) is now given by:

$$W(\alpha, \sigma) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K_\sigma(\mathbf{x_i}, \mathbf{x_j}) \tag{5}$$

subject to $\sum_{i=1}^{\ell} \alpha_i y_i = 0$, $C \geq \alpha_i \geq 0$, $||\sigma||_F = N$, and $\sigma_i \geq 0$. To solve this problem we stepped along the gradient of Equation (5) with respect to σ and α until we reached a local maximum. One iteration consisted of two steps: first we held σ constant and trained the SVM to calculate the solution α^0 of the maximization problem given in Equation (2). In a second step, we kept α constant and performed the gradient descent on $-W$ with respect to σ subject to the constraint on the norm of σ which is an approximation to minimizing the bound on EP_{err} according to Equation (3) for a fixed R. In our experiments we performed one iteration and then ranked the features by decreasing elements σ_i of σ.

4.2 Experiments on different input spaces

We first evaluated the ranking methods on the gray and PCA gray features. The tests were performed on the small test set for 60, 80 and 100 ranked features with a second-degree polynomial SVM. In Figure 2 we show the 100 best gray features, bright gray values indicate high ranking. The Receiver Operator Characteristic (ROC) curves of second-degree polynomial SVMs are shown in Figure 3. For 100 features there is no difference between gray and PCA gray features. However the PCA gray features gave clearly better results for 60 and 80 selected features. For this reason we focused in the following experiments on PCA features only. An interesting observation was that the ranking of the PCA features obtained by the above described gradient descent method was similar to the ranking by decreasing eigenvalues.

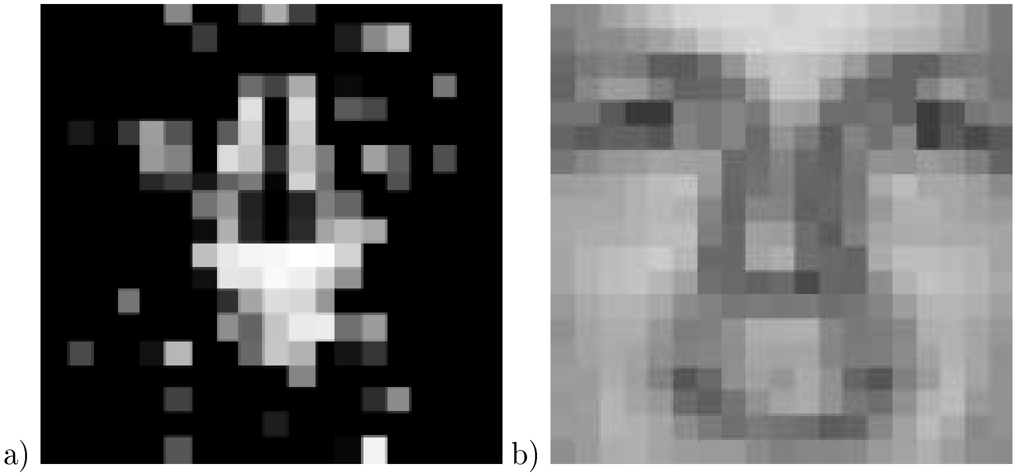

Figure 2: a) First 100 gray features according to ranking by gradient descent. Bright intensities indicate high ranking. b) Reference 19 × 19 face.

To compare PCA gray/gradients with PCA gray features, we performed tests with 50 features on the entire CMU test set 1. Surprisingly, the results for gray values alone were better than those for the combination of gray and gradient values. A possible explanation could be that the gradient value features are noisier than the gray ones.

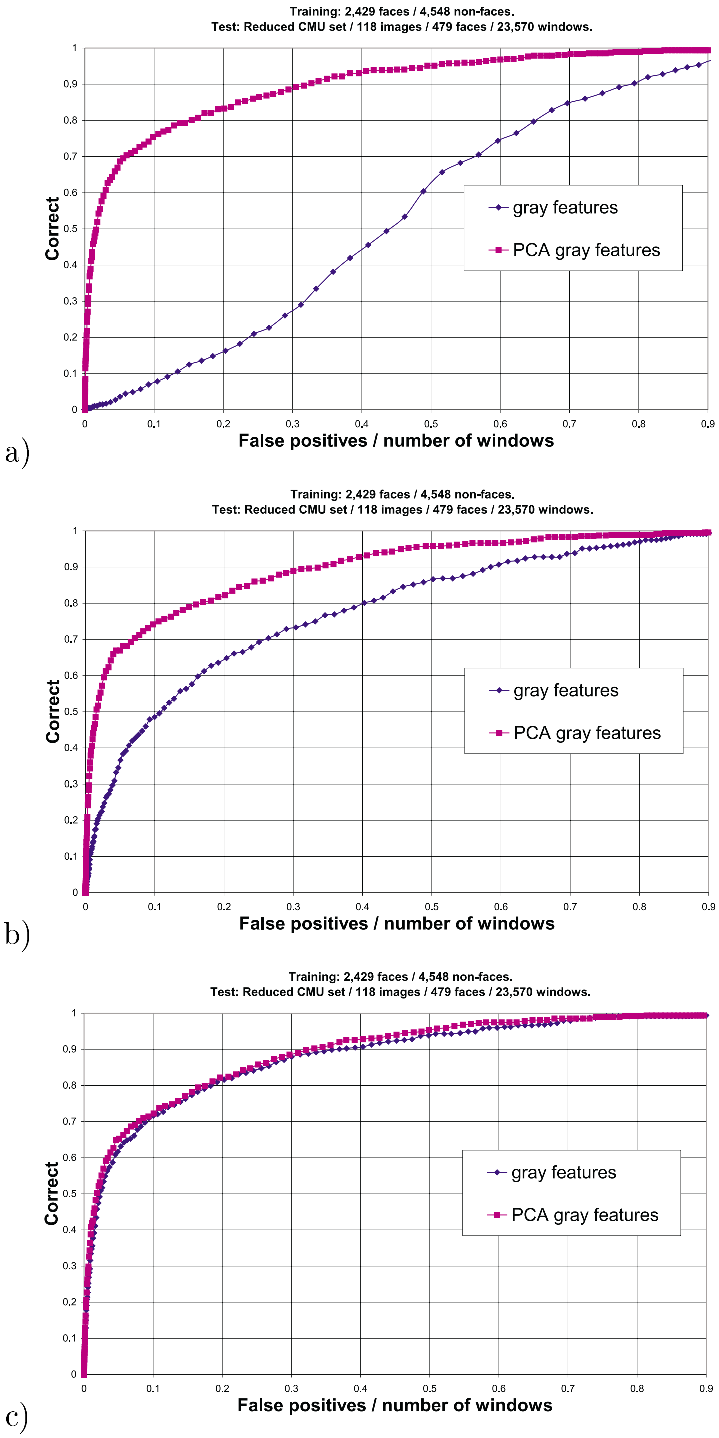

Figure 3: Comparison of the two input spaces for a) 60 features b) 80 features and c) 100 features.

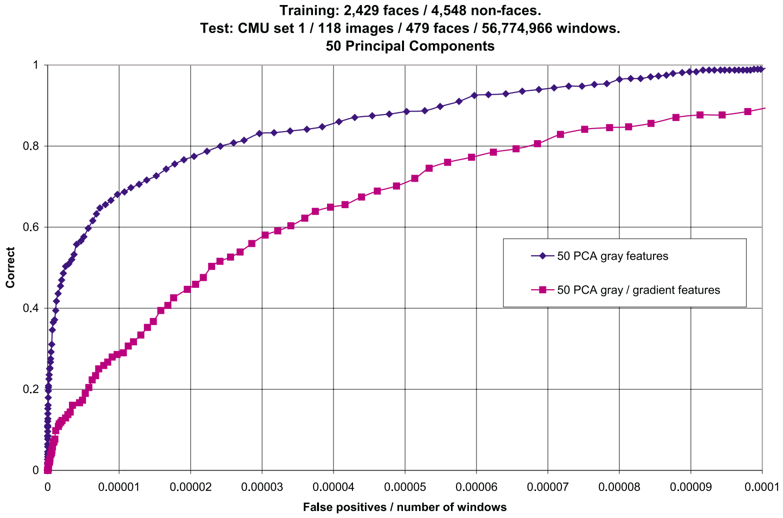

Figure 4: Comparison of the ROC curves for PCA gray features and PCA gray / gradient features.

5 Selecting Features in the Input Space

5.1 Description of the method

In Chapter 4 we ranked the features according to their scaling factors σ_i. Now the problem is to determine a subset of the ranked features $(x_1, x_2, ..., x_n) \in \mathbb{R}^n$. This problem can be formulated as finding the optimal subset of ranked features $(x_1, x_2, ..., x_{n^*})$ among the n possible subsets where $n^* < n$ is the number of selected features. As a measure of the classification performance of an SVM for a given subset of ranked features we used again the bound on the expected error probability.

$$EP_{err} \leq \frac{1}{\ell} \, E\left(R^2 \, W(\alpha^0)\right) \qquad (6)$$

To simplify the computation of our algorithm and to avoid solving a quadratic optimization problem in order to compute the radius R, we approximated[4] R^2 by $2p$ where p is the dimension of the feature space $\mathbb{R}^p$. For a second-degree polynomial

[4] We previously normalized all the data in $\mathbb{R}^n$ to be in a range between 0 and 1. As a result the points lay within a p-dimensional cube of length $\sqrt{2}$ in $\mathbb{R}^p$ and the smallest sphere including all the data points is upper bound by $\sqrt{2p}$.

kernel of type $(1 + \mathbf{x} \cdot \mathbf{y})^2$ we get:

$$EP_{err} \leq \frac{1}{\ell}\ 2p\ E\Big(W(\alpha^0)\Big) \leq \frac{1}{\ell}\ n^*(n^*+3)\ E\Big(W(\alpha^0)\Big) \tag{7}$$

where n^* is the number of selected features[5]. The bound of the expectation of the leave-one-out error is shown in Figure 5. We had no training error for more than 22 selected features. The margin continuously increases with increasing numbers of features. The bound on the expected error shows a plateau between 30 to 60 features, then it significantly increases.

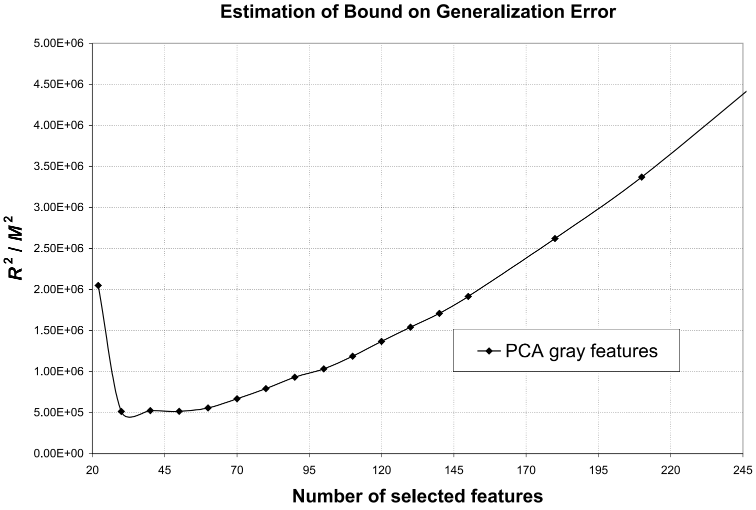

Figure 5: Bound on the expected error number of selected features[6].

5.2 Experiments

To evaluate our method, we tested the system on the large CMU test set 1 consisting of 479 faces and about 57,000,000 non-face patterns. In Figure 6, we compare the ROC curves obtained for different numbers of selected features. The results show that using more than 60 features did not improve the performance of the system.

[5] As we used a second-degree polynomial SVM the dimension of the feature space $p = n^*(n^*{+}3)/2$.

[6] Note that we did not normalize the by the number of training samples l.

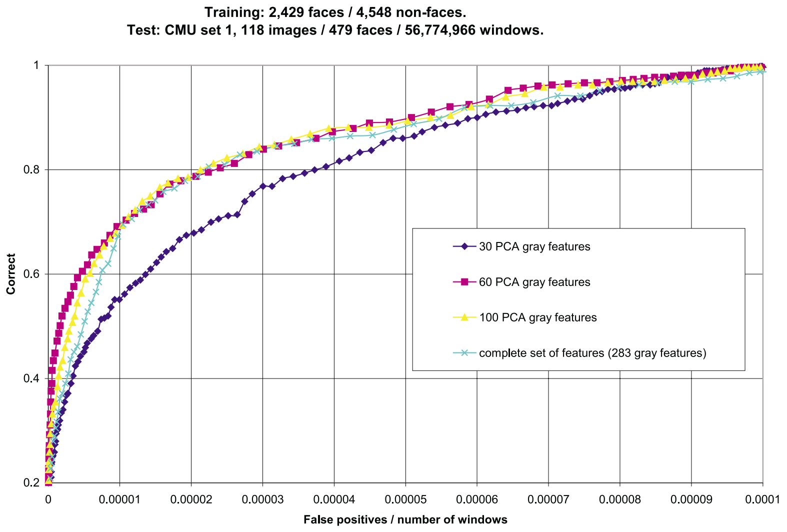

Figure 6: ROC curves for different number of features.

6 Feature Reduction in the Feature Space

In the previous Chapter we described how to reduce the number of features in the input space. Now we consider the problem of reducing the number of features from the feature space. We used the method proposed in [Heisele et al. 2000] based on the contribution of the features to the decision function $f(\mathbf{x})$ of the SVM.

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b = \sum_i \alpha_i^0 y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{8}$$

where $\mathbf{w} = (w_1, ..., w_p)$. For a second-degree polynomial kernel with $K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^2$, the feature space $\mathbb{R}^p$ with dimension $p = \frac{n(n+3)}{2}$ 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_k^* to the decision function in Equation (8) depends on w_k. A straightforward way to order the features is by decreasing $|w_k|$. Alternatively, one can weight $\mathbf{w}$ by the Support Vectors to account for different distributions of the features in the training data. The features were ordered by decreasing $|w_k \sum_i y_i x_{i,k}^*|$, where $x_{i,k}^*$ denotes the k-th component of Support Vector i in feature space $\mathbb{R}^p$.

For the two methods we first trained an SVM with a second-degree polynomial kernel with an input space of 60 features which corresponds to 1891 features in the feature space. We then calculated $\sum_i |f(\mathbf{x_i}) - f_S(\mathbf{x_i})|$ for all Support Vectors, where $f_S(\mathbf{x})$ is the decision function using the S first features according to their ranking. The results in Figure 7 show that ranking by the weighted features of $\mathbf{w}$ lead to faster convergence of the error.

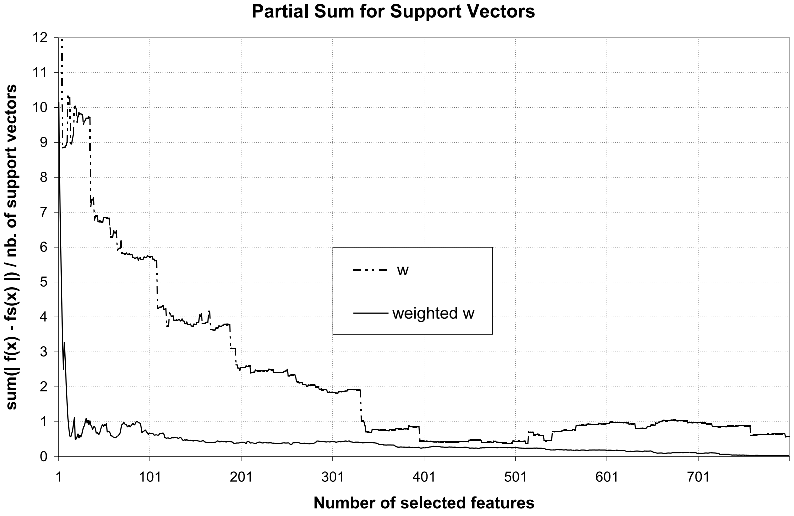

Figure 7: Classifying Support Vectors with a reduced number of features. The x-axis shows the number of features, the y-axis is the mean absolute difference between the output of the SVM using all features and the same SVM using the S first features only. The features were ranked according to the features and the weighted features of the normal vector of the separating hyperplane.

Figure 8 shows the ROC curves for 500 and 1000 features. As a reference we added the ROC curve for a second-degree SVM trained on the original 283 gray features. This corresponds to a feature space of dimensionality $\frac{(283+3)283}{2} = 40,469$. By combining both methods of feature reduction we could reduce the dimensionality by a factor of about 40 without loss in classification performance.

7 Application

7.1 Architecture of the system

We applied feature selection to a real-world application where the goal was to determine the orientation (right side up or up side down) of face images in real-time. To solve this problem we applied frontal face detection to the original and the rotated images (180°). The images in which at least one face was detected with high confidence were considered to be right side up.

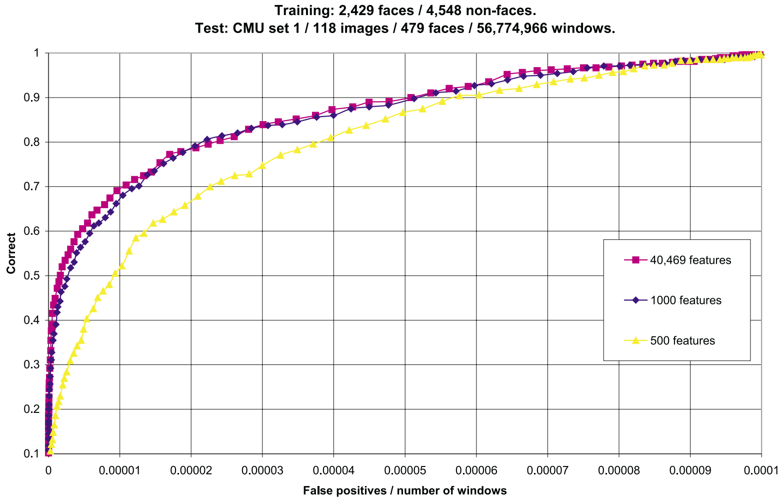

Figure 8: ROC curves for different dimension of the feature space.

We used a subset of the Kodak Database consisting of 283 images of size 512 × 768. The resolution of the faces varied approximately between 20 × 20 and 200 × 200. The average number of faces per image was 2. Even after applying the two feature selection methods described in this paper, the computational complexity of a polynomial second-degree SVM classifier was still too high for a real-time system. That is why we implemented a two-layer system where the first layer consists of a fast linear SVM that removes large parts of the background. The second layer consists of a more accurate polynomial SVM performs the final face detection. Our system is illustrated in Figure 9. (B) and (C) show the responses of the linear classifier for the original and the rotated images. Bright values indicate the presence of faces. Thresholding these images leads to binary images (A) and (D) where the locations of potential faces are drawn in black. At these locations we search for faces using the polynomial second-degree SVM of the second layer.

7.2 Experiments

In the first experiment we applied a second-degree SVM classifier trained on 60 PCA features to the Kodak database. All 283 images were right side up. The results are shown in Figure 10 and compared to the ROC curve for the CMU test set. The fact that the ROC curve for the Kodak database is worse than the ROC curve for the CMU test set 1 can be explained by the large number of rotated faces, faces of babies, and children with masked faces (see Figure 11).

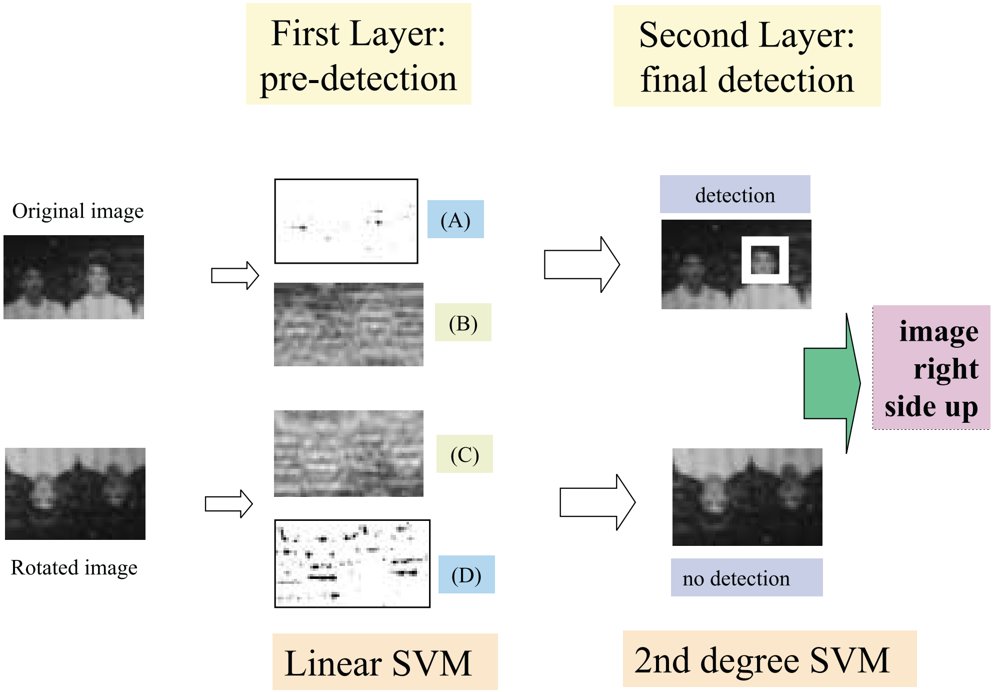

Figure 9: Architecture of the real-time system determining the orientation of a face.

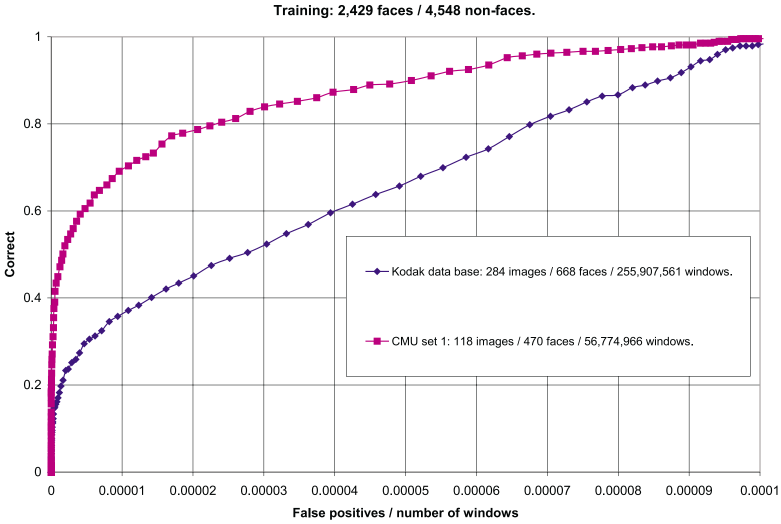

Figure 10: ROC curve for the Kodak database.

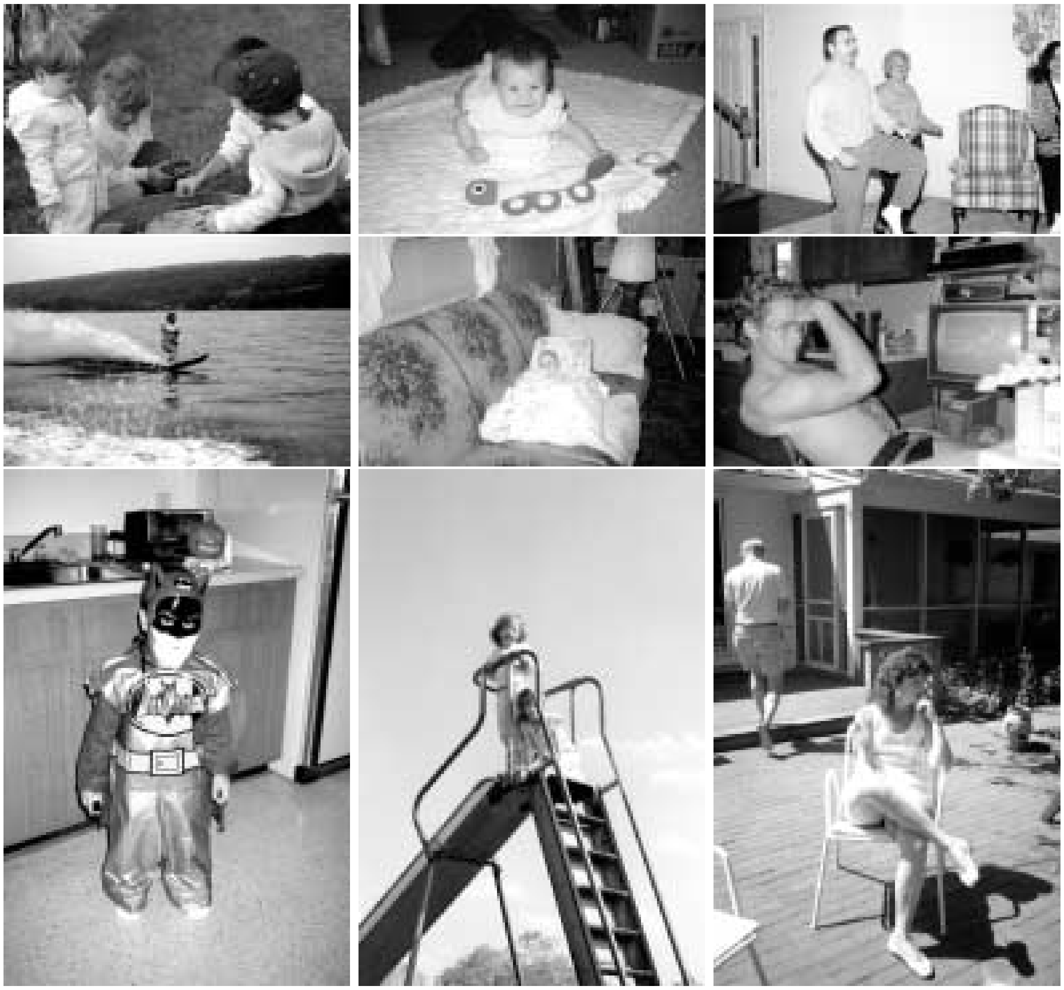

Figure 11: Images from the Kodak database.

In a second experiment we considered the two-layer system. We chose the threshold for the linear SVM from previous results on the CMU test set. For this threshold we classified correctly 99.8% of faces and 99.9% of non-face patterns.
In the worst case, the average number of multiplications for the whole system is about 300 per pixel and per scale [7]. Searching for a face directly with a second-degree polynomial SVM using gray values would have lead to $81,000$ operations. As a result, we sped up the system by a factor of 270.

8 Conclusion

We presented a method to select features for a face detection system using Support Vector Machines (SVMs). By ranking and then selecting PCA gray features according to a SVM classification criterion we could remove about 80% of the input features. In a second step we further reduced the dimensionality by removing features with low contributions to the decision function of the SVM. Overall we kept less than 2% of the original features without loss in classification performance. We demonstrated the efficiency of our method by developing a real-time system that is able to determine the orientation of faces.

References

[Heisele et al. 2000] B. Heisele, T. Poggio, M. Pontil. *Face Detection in Still gray Images.* A.I. memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000.

[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.

[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.

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

[7] The number of operations for the first level is equal to 283 (dimension of the space). For the second level we assume that the percentage of pixels that pass the first level is equal to 0.001. For projecting the data into the eigenvector space we have to perform 60 × 283 multiplications. Finally we have to project the input features into the feature space and calculate the dot product of the 1000 selected features with the normal vector of the separating hyperplane. Overall this results in $0.001 \cdot (60 \cdot 283 + 2 \cdot 1000) = 19$ multiplications per shifted window.

[Weston et al. 2000] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V. Vapnik. *Feature Selection for SVM's.* Submitted to Advances in Neural Information Processing Systems 13, 2000.