SIViP (2007) 1:29–38
DOI 10.1007/s11760-007-0003-x

ORIGINAL PAPER

Mixtures of boosted classifiers for frontal face detection

**Julien Meynet · Vlad Popovici ·
Jean-Philippe Thiran**

Received: 12 October 2006 / Revised: 12 January 2007 / Accepted: 12 January 2007 / Published online: 3 March 2007

Abstract This paper describes a new approach to automatic frontal face detection which employs Gaussian filters as local image descriptors. We then show how the paradigm of classifier combination can be used for building a face detector that outperforms the current state-of-the-art systems, while remaining fast enough for being used in real–time systems. It is based on the combination of several parallel classifiers trained on subsets of the complete training set. We report a number of results on some reference datasets and we use an unbiased method for comparing the detectors.

Keywords Face detection · Adaboost ·
Combination of classifiers

1 Introduction

Given a still image, the goal of automatic face detection is to find all the human faces present in the image and to return their support regions (i.e. the bounding boxes). This is a key step in any system that relies on face processing (like face recognition or facial expression recognition) and its performance represents a limiting factor for the quality of the whole system [1]. There are a large number of factors that influence the detection, some of them being intrinsic (e.g. inter-personal variability, face expression, gender, age and so forth) others being imposed by the environment (e.g. illumination, shadowing, camera parameters). Their combined effect makes the task of automatic face detection very challenging and, despite the research effort of the last decades, still unsolved for general cases.

During the last decades, a full range of methods have been proposed and we give hereafter a brief overview of some of the most significant ones. At one end of the spectrum are the holistic methods, where the whole face is treated as a single object, while at the other end are the feature-based methods, where parts of the faces are identified independently and a final decision is taken by assembling the evidences. Between these two extrema lie other methods that combine global and local information for a better detection. The first category usually produces very fast detectors with better classification performances and turns out to be more robust to light changes. However, one of the main advantages of the feature-based methods is that they are more robust to head pose changes. In this work we only consider frontal faces and thus, in the remaining of the paper, only the holistic methods will be considered. For detailed surveys the reader is referred to [2] and [3].

The classical approach for face detection is to scan the input image with a sliding window and for each position, the window is classified as either face or non face. The method can be applied at different scales (and possibly different orientations) for detecting faces of various sizes (and orientations). Finally, after the whole search space has been explored, an arbitration technique may be employed for merging multiple detections. Of course the efficient exploration of the search space is a key ingredient for obtaining a fast face detector. There

J. Meynet (✉) · J.-P. Thiran
Ecole Polytechnique Fédérale de Lausanne (EPFL),
Signal Processing Institute, CH-1015 Lausanne, Switzerland
e-mail: julien.meynet@epfl.ch

V. Popovici
Bioinformatics Core Facility, Swiss Institute of Bioinformatics,
CH-1015 Lausanne, Switzerland

are various methods to speed up this search, like using additional information (e.g. skin color) or using a coarse-to-fine approach. Nevertheless the most important component of the system is the classifier deciding whether a given window contains a face or not. From this perspective, this paper focuses on both aspects, efficient search space and robust classifier.

The first reference algorithm has been proposed by Sung and Poggio [4]. They use clusters of face and non face models to decide whether a constant sized window contains a face or not. The principle is to use several Gaussian clusters to model both classes. Then the decision is taken according to the relative distance of the sample to the mean of both classes. In order to detect faces at any scale and position they use a sliding window which scans a pyramid of images at different scales. The detector proposed by Schneiderman and Kanade [5] also models the probability distribution of the face class, but they employ a naive Bayes classifier. A similar holistic approach proposed by Rowley et al. in [6] is one of the most representative for the class of neural network approaches. It comprises two modules: a classification module which hypothesizes the presence of a face and a module for arbitrating multiple detections. A fast algorithm is proposed by Viola and Jones in [7]. It is based on three main ideas. They first train a strong classifier by boosting the performance of simple rectangular Haar-like feature-based classifiers. They use the so-called integral image as image representation which allows to compute the base classifiers very efficiently. Finally they introduce a classification structure in cascade in order to improve both the detection speed and the classification results. This last method (in particular the cascade structure) leads to a very fast detection (about 30 frames per second on a 2.8 GHz PC for 320 × 240 images). As it will be explained later in the paper, we have used this method as a pre-processing step in order to reduce the search space. Some improvements to this method have been proposed in the last years. First, Lienhart and Maydt [8] used an extended set of rectangular shaped filters. They introduced filters rotated by ±45° still computable with the integral image. Then several variants of Adaboost have been studied: Realboost by Wu et al. [9] and Floatboost by Li and Zhang [10].

In this work we present a new approach for detecting frontal faces in real time. First, binary classifiers are obtained by boosting weak classifiers based on anisotropic Gaussian filters which are more discriminative than the Haar filters introduced in [7]. Then the training of several of these classifiers in parallel reduces the complexity of the training process while improving the classification performances. This idea of splitting a complex problem into several lower complexity problems has been discused in [11] and [12], where the combination of several support vector machines (SVM) is done either by a linear SVM or using some other combination rules. A review of how classifiers can be combined together can be found in [13]. Figure 1 shows an overview of the complete face detection system presented in this work.

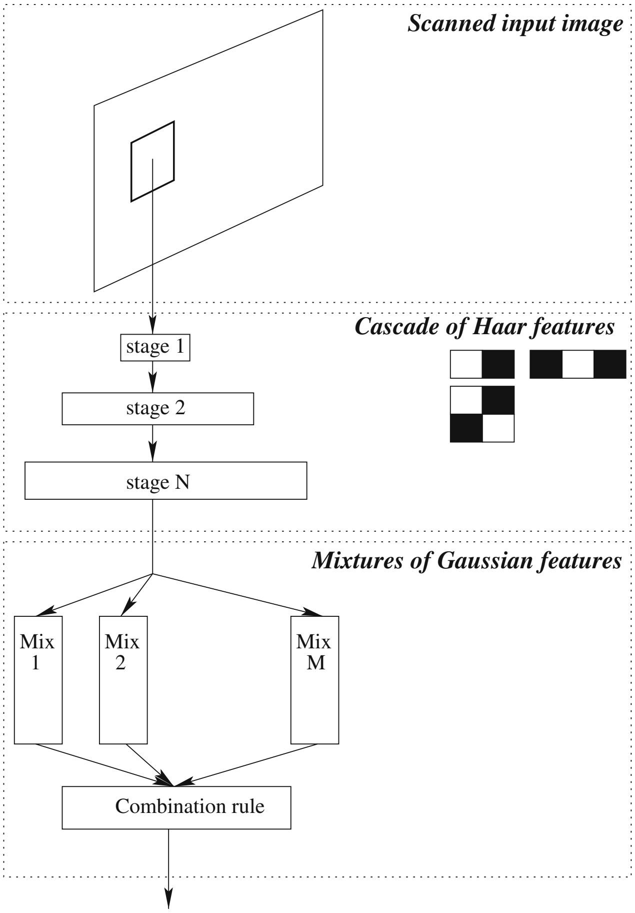


Fig. 1 Overview of the face detection system

The remaining of the paper is structured as follows. Section 2 gives an overview of the geometrical filters and discusses their ability to model the face patterns. It also describes the kind of classifiers used in this study. Section 3 presents the mixtures of boosted classifiers and how they are combined together to take the final decision. Section 4 reports some results as well as comparisons with relevant existing face detectors. Finally, we draw some conclusions in sect. 5.

2 Boosting anisotropic Gaussian features

We will start with a short overview of the Adaptive Boosting (AdaBoost) algorithm and we will show how it can be used for performing feature selection too. Then we will introduce the anisotropic Gaussian filters used for face modeling.

Let

$$\{(\mathbf{x}_i, y_i), i = 1, \dots, l\} \subset \mathbb{R}^n \times \{-1, +1\} \tag{1}$$

be a set of labeled examples generated according to an unknown (but fixed) probability distribution function $P(\mathbf{x}, y)$. The problem of learning may be expressed as an optimization problem in which one wants to find the function f_{α^*} (from a suitably chosen set of functions, indexed here by the parameter α) which minimizes the risk of misclassifying new vectors drawn from the same *pdf* P:

$$\alpha^* = \arg\min_{\alpha} R(\alpha) = \arg\min_{\alpha} \int \mathcal{L}(y, f_\alpha(\mathbf{x}))\mathrm{d}P(\mathbf{x}, y), \tag{2}$$

where R is called *risk* and $\mathcal{L}$ is called *loss functional*. The loss functional penalizes the differences between the true label y and the predicted one $f_\alpha(\mathbf{x})$, and it has specific forms in various learning algorithms.

2.1 AdaBoost

AdaBoost [14] is a learning algorithm which iteratively builds a linear combination of some basic functions (*weak classifiers*) by greedily minimizing the risk based on the exponential loss,

$$L(y, f(\mathbf{x})) = \exp(-yf(\mathbf{x})). \tag{3}$$

The final decision function has the form

$$f_T(\mathbf{x}) = \mathrm{sign}\left(\beta_0 + \sum_{k=1}^{T} \beta_k h_k(\mathbf{x})\right), \tag{4}$$

with $h_k : \mathbb{R}^n \to \{\pm 1\}$ being the weak classifiers. Training in the case of AdaBoost comes to finding the weak classifiers and their corresponding weights. For a detailed description of the algorithm the reader is referred to [14]. There are a number of theoretical and practical advantages in using AdaBoost, of importance here being the fact that by suitably choosing the weak classifiers, one may perform a feature selection implicitly when training the classifier. Another important feature of AdaBoost is that it converges towards a large margin classifier with positive impact on its generalization properties. However, in the presence of high levels of noise, AdaBoost, like the majority of classifiers, may overfit the training set.

Finally, note that depending on the application we might prefer to favor one of the classes. In AdaBoost, this can be easily implemented. We can first build an asymmetric version of AdaBoost that encourages the correct classification of the desired examples. We also tune the final threshold β_0 on an independent set in order to obtain the desired operating point on the receiver operating characteristic (ROC) curve.

Now let $\mathbf{x} \in \mathbb{R}^n$ be a vector whose components will be denoted by $x_j, j = 1, \dots, n$. If we let the weak classifiers be

$$h_j(\mathbf{x}) = \begin{cases} 1, & \text{if} \quad p_j x_j < p_j \theta_j \\ -1, & \text{otherwise} \end{cases}, \tag{5}$$

it turns out that AdaBoost will perform a feature selection too. Indeed, the final decision function will be a linear combination depending only on some of the features. This is the particular form of the weak classifiers that will be used for building the face detector and $\mathbf{x}$ will be the vector of all filter responses when applied to one image. For these weak learners (decision stumps), there are two parameters to be tuned: the threshold θ_j (chosen by maximum a posteriori rule) and the parity p_j.

2.2 Anisotropic Gaussian filters

In this section we propose a new set of local filters to be used for constructing the weak classifiers. The filters are made of a combination of a Gaussian in one direction and its first derivative in the orthogonal direction and have been introduced by Peotta et al. in [15] for image compression and signal approximation. The generating function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is given by:

$$\phi(u, v) = u \exp(-|u| - v^2). \tag{6}$$

It efficiently captures contour singularities with a smooth low resolution function in the direction of the contour and it approximates the edge transition in the orthogonal direction with the first derivative of the Gaussian.

In order to generate a collection of local filters, the following transformations can be applied to the generating function:

- Translation by (u_0, v_0): $\mathcal{T}_{u_0,v_0}\phi(u, v) = \phi(u - v_0, u - v_0)$.
- Rotation with θ: $\mathcal{R}_\theta\phi(u, v) = \phi(u\cos\theta - v\sin\theta, u \sin\theta + v\cos\theta)$.

- Bending by r:

$$\mathcal{B}_r\phi(u,v) = \begin{cases} \phi\left(r - \sqrt{(u-r)^2+v^2}, r\arctan\left(\frac{v}{r-u}\right)\right) & \text{if } u < r \\ \phi\left(r - |v|, u - r + r\frac{\pi}{2}\right) & \text{if } u \geq r \end{cases}$$

- Anisotropic scaling by (s_u, s_v): $\mathcal{S}_{s_u,s_v}\phi(u,v) = \phi(\frac{u}{s_u}, \frac{v}{s_v})$.

By combining these four basic transformations, we obtain a large collection of functions $\mathcal{D} = \{\psi_{s_u,s_v,\theta,r,u_0,v_0}(u,v)\} = \{\mathcal{T}_{u_0,v_0}\mathcal{R}_\theta\mathcal{B}_r\mathcal{S}_{s_u,s_v}\phi(u,v)\}$. Figure 2a shows some of these functions with various bending and rotating parameters. We define the example $\mathbf{x}_k = (x_{jk})$ as the local responses of an image I_k to the all of the filters from $\mathcal{D}$:

$$x_{jk} = \iint \psi_j(u,v) I_k(u,v)\,\mathrm{d}u\mathrm{d}v \quad \forall\psi_j \in \mathcal{D}, \tag{7}$$

where the integral is taken over a suitable domain.

Figure 3 shows some functions selected in the first iterations of AdaBoost. It turns out that they are particularly well adapted to capture local contours which are less sensitive to changes of the lighting conditions. In comparison, Haar filters model global contrasts that are more sensitive the direction of the light source.

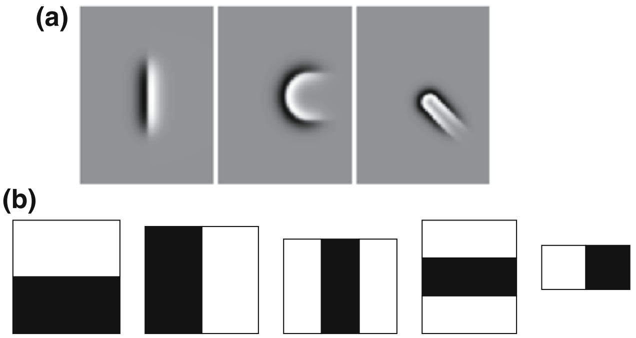


Fig. 2 Filters used for modeling the faces: **a** Anisotropic Gaussian and **b** Haar-like filter

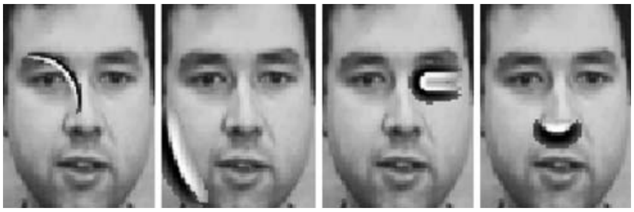


Fig. 3 Some of the first selected base functions

The proposed features are susceptible to be used by other classification methods too. Our previous experience taught us that SVM [16] and cascaded AdaBoost have similar performances, but usually the cascaded classifier is faster. It would be interesting to compare and analyze different classifiers in the context of Gaussian features, but this goes well beyond the scope of the present paper.

2.3 Gaussian vs. Haar-like features

We are interested, first of all, to compare the Gaussian features (GF) with the more commonly used Haar-like features (HF), introduced in [7] (see Fig. 2b for an example of such features). As we want to gain some insights about the intrinsic discrimination power of the two type of features, we trained two detectors using either the Gaussian filters or the Haar-like features, using the same training sets, and then we compared the two on an independent validation set. Figure 4 shows the classification performance of the two classifiers either in terms of error rates.

It is interesting to note from Fig. 4 that while for the first ~100 iterations the error rate decreases quickly as we add more features to the model of the both classifiers, it remains practically constant for the HF-based detector. However, the GF model keeps improving, as we add more and more features. This shows that the HFs are not discriminant enough for modeling the finer differences between the two classes. Figure 3 shows the GFs that were selected during the first iterations so those that were deemed the most discriminative. Note also how they adapt to model the most salient features of the faces.

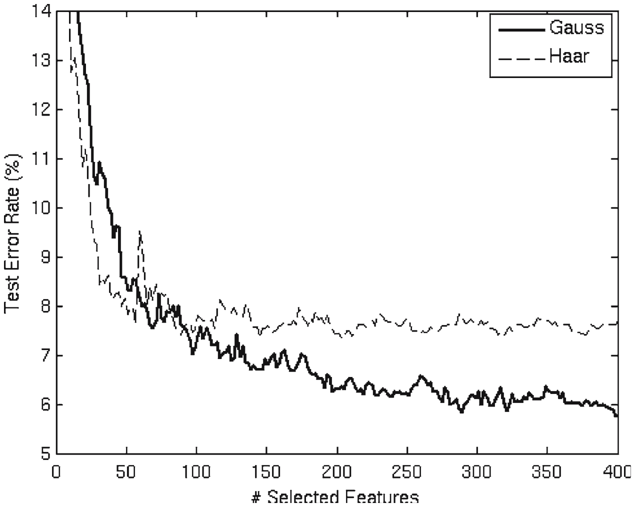


Fig. 4 Performance of GF and HF-based detectors

We also compared HF and GF in terms of computation time. We trained two similar classifiers with 200 HF on one hand and 200 GF on the other hand. By applying these two classifiers on several images, we compared the average computation time for applying a single HF (computed with the integral image trick as in [7]) and a single GF. We found that computing a GF takes roughly 2.86 more time than a HF (note that the Gaussian filters are precomputed in the model such that the expensive computation of the generative function is avoided).

3 Mixtures of boosted classifiers

3.1 Motivations

Boosting can construct a strong classifier by producing a linear combination of weak classifiers. This section introduces a structure that will improve the classification skills of the face detection system.

As already mentionned in the introduction, one of challenges of face detection resides in the fact that a very large set of face and non-face examples must be collected. Moreover a large number of features is needed to obtain a sufficiently low false positive rate. AdaBoost minimizes an exponential loss function (see Eq. 3) so that after several iterations, many features have to be added for slightly reducing the false positive rate.

On the other hand, some variation in the face training examples is needed in order to be able to detect faces with slight pose variations and with slight scale changes. A large training dataset is thus necessary to cover all this variability. Consequently, some weak classifiers potentially very efficient as experts on local subspaces of the training data might behave worse on the whole training set.

These motivations suggest to use a multi-classifier structure built in parallel. Instead of training a single boosted classifier on the complete training set, we built several classifiers on subsets of the original dataset. A similar technique was developed and discussed in [11] where SVM were used for the parallel classifiers. This is similar to the Bagging technique introduced in [17] except that the size of the bootstrap samples is smaller than the initial sample set.

There are many interesting points in such an approach. On the first hand, each classifier is trained on a subset of the training dataset so that it can be seen as an expert that focuses on its own domain. This will thus decrease the influence of potential outliers in the complete training set. More specifically, as the power of AdaBoost resides in the fact that it focuses on the hard to classify examples, the parallelization technique reduces the weight of the noisy examples or potential outliers. This last point also reduces the risk of overfitting as mentionned above. From a practical point of view it will decrease the false positive rate which is a important in the context of face detection.

On the second hand, this parallelization technique also allows to decrease the classifier complexity. The complexity of training AdaBoost varies linearly with the number of samples. Splitting the data and training several AdaBoosted classifiers on the subsets will thus not affect the training complexity as compared to a single AdaBoosted classifier. However, as it will be shown in Sect. 4, less features are needed for achieving equivalent classification rates compared to a single classifier trained on the complete training set.

We could imagine two strategies for splitting the dataset into several subsets: either random sampling if we want to estimate several times the decision boundary or clustering if we want to build experts on subsets of the face class. In our case, no information is available about the distribution of the face class, we just want to simplify the problem while improving the classification skills, that is why simple random sampling has been chosen for creating the training subsets. Another reason why the clustering would not be appropriate comes from the variations introduced in the training set. This variability is obtained by slightly rotating, shifting and scaling the original face images. The clustering would eventually cluster examples resulting from similar transformations and thus the combination would probably fail.

3.2 Posterior probability estimation

The decision of the parallel classifiers are then combined using simple probability rules. For this, posterior probabilities of the boosted classifiers need to be estimated. This section discuss this probability estimation.

First recall that AdaBoost minimizes the exponential criterion:

$$J(f) = E(\mathrm{e}^{-yf(\mathbf{x})}). \tag{8}$$

Friedman et al. [18] shows that minimizing $J(f)$ in Eq. (8) is equivalent up to second order Taylor expansion to maximizing the expected binomial log-likelihood. The posterior probabilities $P(y=1|\mathbf{x})$ and $P(y=-1|\mathbf{x})$ are given by the following lemma:

Theorem 1 [18] $J(f) = E(\mathrm{e}^{-yf(\mathbf{x})})$ *is minimized at*

$$f(\mathbf{x}) = \frac{1}{2}\log\frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})}. \tag{9}$$

Hence

$$P(y = 1|\mathbf{x}) = \frac{e^{f(\mathbf{x})}}{e^{-f(\mathbf{x})} + e^{f(\mathbf{x})}}, \tag{10}$$

$$P(y = -1|\mathbf{x}) = \frac{e^{-f(\mathbf{x})}}{e^{-f(\mathbf{x})} + e^{f(\mathbf{x})}} \tag{11}$$

Then, several different strategies may be used for combining parallel classifiers. A complete overview can be found in [13] and [19]. Consider that we want to classify a pattern $\mathbf{x}$ in one of the classes $y = +1, y = -1$. We model both classes by the probability density functions $p(\mathbf{x}|y = i)$ and their prior probability by $P(y = i)$. Assume that we need to combine M classifiers and denote by $p_j(y_i|\mathbf{x})$ the posterior probability that $\mathbf{x}$ belongs to class y_i, estimated from the j-th classifier. The Bayes decision rule states that an example $\mathbf{x}$ is assigned to the class $y = 1$ if:

$$P(y = 1|\mathbf{x}) > P(y = -1|\mathbf{x}) \tag{12}$$

Equation (12) relies on the theoretical framework of the classification task but its computation is infeasible in practice. That is why we simplify the problem by using some basic combination rules easier to compute. In this work we focus on six well-known simple probabilistic rules. Although they seem simple, Kittler et al. [13] point out the probability assumptions that are needed for each rule. The rules are defined in the following:

– Product rule: Example $\mathbf{x}$ is assigned to the class $y = 1$ if:

$$P(y = 1) \prod_{j=1,\ldots,M} p_j(\mathbf{x}|y = 1) > P(y = -1) \prod_{j=1,\ldots,M} p_j(\mathbf{x}|y = -1) \tag{13}$$

  This rule derives directly from Bayes theorem by assuming that the measurements of the different classifiers are conditionally independent;

– Sum rule: Example $\mathbf{x}$ is assigned to the class $y = 1$ if:

$$(1 - M)P(y = 1) + \sum_{j=1,\ldots,M} P_j(y = 1|\mathbf{x}) > (1 - M)P(y = -1) + \sum_{j=1,\ldots,M} P_j(y = -1|\mathbf{x}). \tag{14}$$

Then from Eqs. (13) and (14) we derivate four other combination rules. In all the cases, Example $\mathbf{x}$ is assigned to the class $y = 1$ if:

– Max rule:

$$\max_{j=1,\ldots,M} P_j(y = 1|\mathbf{x}) > \max_{j=1,\ldots,M} P_j(y = -1|\mathbf{x}) \tag{15}$$

– Min rule:

$$\min_{j=1,\ldots,M} P_j(y = 1|\mathbf{x}) > \min_{j=1,\ldots,M} P_j(y = -1|\mathbf{x}) \tag{16}$$

– Median rule:

$$\text{median}_{j=1,\ldots,M} P_j(y = 1|\mathbf{x}) > \text{median}_{j=1,\ldots,M} P_j(y = -1|\mathbf{x}) \tag{17}$$

– Majority vote: The class with the largest number of votes is chosen.

Previous studies [12,13,20] noticed that the choice of the rule has not a large influence on the overall performances. In this work we only consider the summation rule defined in Eq. (14) for combining the decisions of the multiple boosted classifiers. The choice of this rule is influenced by the splitting method that is used. The sum rule averages the decisions of the individual classifiers so that it is a good trade off for discarding false alarms while preserving the correct detection of faces. For example the product rule is known to be a severe rule which risks to strongly penalize the true positive rate. More comments about the choice of the decision criterion are given in [13].

A reason why simple probability rules are used for combining the expertise of each individual classifier is the stability of the parallel classifiers. The boosted classifiers are stable in the sense that small changes in the training set lead to small changes in the classifier output [19]. More sophisticated combination techniques like Boosting need unstable classifiers to improve the overall performance.

3.3 Discussion

This parallelization technique presents some advantages against the cascade structure. A cascade of classifiers is a sequential combination of classifiers such that an example is rejected if it is classified as negative at any stage of the cascade. It can be seen as a mixture of classifiers but considering a product probability rule for combining the decisions. In fact if we consider the parallel classifiers to be conditionally independent (which can be assumed in this study as we use random sampling for generating the subsets), if one of the classifiers considers an example as negative with probability close to 1, the probability that the final decision is negative will be high. The only difference would be from the complexity point of view as we would have to test all the classifiers whereas the cascade would directly stop the processing chain.

One advantage of our parallel approach over the cascade is that if a positive example is classified as negative by a given classifier, it can be reassigned to the positive class by the overall system where in the cascade case it would be rejected. This would especially happen in the last stages of the cascade as the examples becomes more and more complicated. It is clear that the mixture approach will not reduce the testing time as we roughly use the same features number as in a single layer classifier. However we do not need to optimize the testing time as we only need to test a few remaining critical windows.

4 Experiments and results

4.1 Structure of the system

In order to test the performances of this system and compare it with other relevant methods the following experiments have been performed. First, the size of the scanning window influences directly the quality of the detector [8]. According to previous empirical studies, we used 20 × 15 pixels window to scan the images. It is then dilated by powers of 1.2 in order to detect faces at any scale. Then a very simple arbitration method clusters the neighbor positive windows such that only one detection per face is returned. We simply return the mean window for each cluster (average position and scale).

To train the models, face images were collected from some classical face datasets: XM2VTS [21], BioID [22], FERET [23]. After adding some variations in scale, in-plane rotations and shifts, the complete face training set contained 9, 500 images. The non face dataset was bootstrapped from randomly selected images without human faces. A total of roughly 500, 000 non face images were finally used.

We also used a separate validation set made of roughly 10, 000 faces and 100, 000 in order to tune various hyperparameters. For example the number of filters selected in each classifier is determined according to the desired detection rates on this validation set. The learning process was stopped when the classifier achieved more that 99.9% of true positive rate and less than 0.1% of false positive rate. All these datasets are available upon request.

The set of Haar-like features (HF) that we used to train the cascade contained 37,520 filters (all possible combinations in a 20 × 15 pixels window). Training classifiers based on GF is more costly than only using HF. We first need to compute all the filter responses for each training pattern. Once the responses are computed, the complexity of the training varies linearly with the number of input filters. As the total number of GF is huge, we decided to randomly sample the collection of filters in order to keep a managable set for the training process. A total of 202,200 filters were finally generated. The final detector was trained in roughly one week on five parallel processors.

An ambiguous point in face detection algorithms is the way the performances are measured. Papers usually provide the detection rate and the false positive rate to show the quality of their system, however they often consider different criteria to measure those rates. It becomes very difficult to objectively compare different published results. In this work, the problem is addressed using the evaluation protocol proposed by Popovici et al. [24]. The evaluation is performed by taking into account several parameters between the detected location and the annotated positions. The scoring function measures the ratio of the between-eyes distances, the angle between the eyes axis and of course the distance between the annotated and detected eye positions. This method gives a more objective scoring of the detection performances. See [24] for details on how to use the scoring function.

4.2 BANCA database

The system has been tested on two distinct datasets. On one hand we considered the BANCA database [25] which was built for training and testing multi-modal identity verification systems. The face images were acquired using various cameras and under several scenarios (controlled, degraded and adverse). Some examples of detection results of the adverse scenario are shown in Fig. 5. In this work, we used 12,480 images from the so-called French and English datasets as we dispose of precise groundtruth annotations for these ones.

Table 1 gives a comparison of different classifiers tested on the 12,480 images. A first classifier is made of five stages of a cascade of Haar features. It discards a large majority of negative windows but is not sufficient for being used alone. It is therefore used as pre-processing to speed up the process. The thresholds are tuned in order to have a very low true negative rate. Then a classifier trained using GF using the complete training set is added. It significantly improves the classification rates. Finally the parallelization strategy has been tested for training the classifiers with GF, we improve the performances by roughly 6%. The single Gaussian-

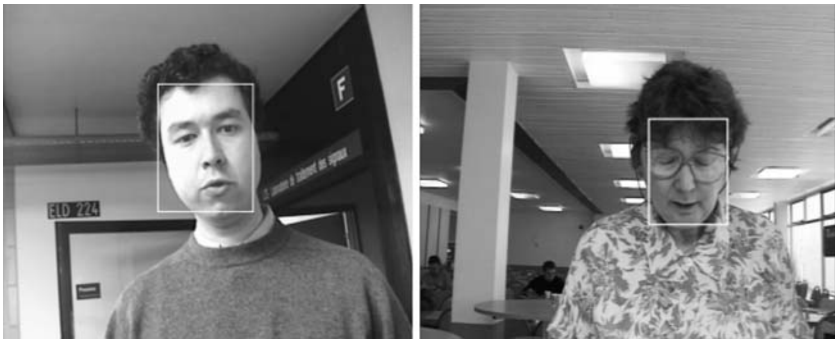


Fig. 5 Results on images of BANCA [25] in the complex adverse scenario

based classifier was trained using the same data than the complete mixture and roughly the same number of filters were selected for both cases, however the mixture performs better.

Table 1 Comparisons of various methods tested on the BANCA [25] database. Results are reported for the French and English parts following the evaluation protocol described in [24]. Detections with a global score larger than 0.95 are considered as correct

Classifier	% of detections with score > 0.95 (following [24])
5 stages Boosted HF	52.08
5 stages Boosted HF + 1 stage of 500 Boosted GF	90.74
5 stages Boosted HF + mix Boosted GF	96.37

4.3 CMU/MIT Test set

We now consider a more challenging database commonly used to evaluate performances of face detectors especially on very low resolution faces. The CMU/MIT Test set [26] was first introduced by Rowley et al. [6] for testing. The first version of this test set contained 23 images with a total of 155 very low resolution faces(it is referred as Dataset 1 in Table 2). The complete set contains 130 images with 507 faces (Dataset 3 in Table 2). However, some of these annotated faces are manually drawn and they are counted as false detections in some publications. To address this ambiguity, some papers only consider 123 images with 483 faces (Dataset 2 in Table 2). The three versions of the dataset are tested in this paper to avoid any confusion. Figure 7 shows some detection results on images of this database.

Table 2 gives comparisons with the state-of-the-art methods on these datasets. Performances of two different techniques are reported. On the first hand, a simple stage of roughly 500 Gaussian features has been tested and then a mixture of GF based classifiers. It comprises five classifiers each made of roughly 100 filters. The number of classifiers in the mixture was chosen in order to obtain a good trade-off between false positive rate and detection speed. Each tested window is pre-processed using histogram equalization and simple illumination correction. This study confirms the improvements due to the parallelization technique. The faces that are drawings or sketches detected in Dataset 3 are counted as false detections in Dataset 2. This explains why there are more false detections in the smallest version of the test set. Table 2 only gives a single operating regime (i.e. single point on the The receiver operating curve (ROC) curve). We thus also include complete ROC curves in Fig. 6. It shows that we gain several percents of detection rates compared to single GF-based classifier, and this at any operating point on the ROC.

The mixture of GF technique also compares favorably to state of the art. Nevertheless, the results in this table have to be taken cautiously as they are affected by many factors: the scanning parameters (scaling factor, window

Table 2 Performances on the CMU/MIT test set [26]. Three datasets configurations are considered: Dataset 1: 155 faces, Dataset 2: 483 faces, Dataset 3: 507 faces. It shows the detection rate (D.R.) and number of false alarms (F.A) for each method

	Dataset 1		Dataset 2		Dataset 3	
Methods	D.R. (%)	F.A.	D.R. (%)	F.A.	D.R. (%)	F.A.
Rowley et al. [6]	87.1	15	92.5	862	90.5	570
Sung and Poggio [4]	81.9	13	–	–	–	—
Shneiderman and Kanade [5]	–	–	93.0	88	94.4	65
Viola and Jones [7]	–	–	–	–	91.4	50
5 stages HF +1 stage GF	**89.2**	**17**	**91.7**	**63**	**91.8**	**55**
5 stages HF + Mix	**89.2**	**15**	**92.1**	**68**	**93.9**	**60**

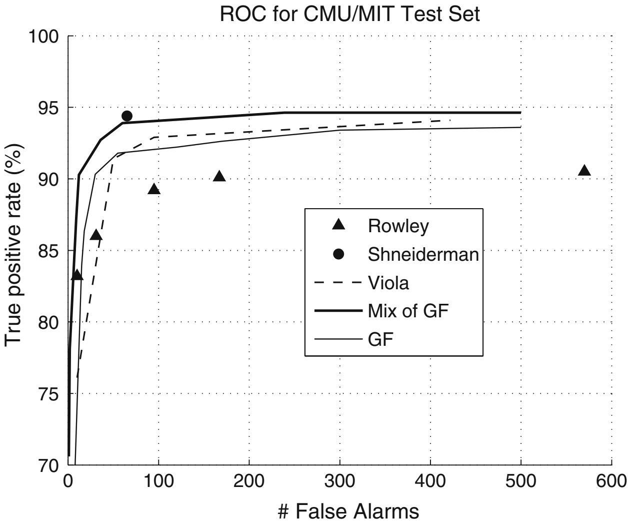
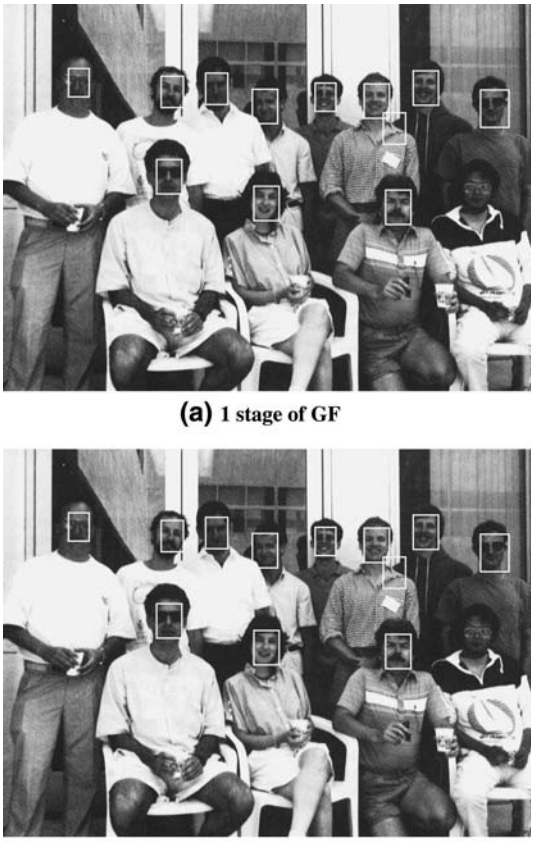


Fig. 6 ROC analysis for comparing the algorithms on the MIT/CMU testset [26]



(a) 1 stage of GF

(b) Mixture GF

Fig. 7 Comparison between 1 stage of GF (**a**) and a mixture of GF (**b**), both pre-processed by 5 stages of HF. The image is taken from the CMU/MIT test set [26]

shifting step, etc, ...), the technique chosen for merging overlapping windows, the number of training patterns and so forth (Fig. 7). In particular, the way the non–face examples are generated has a major impact on the decision functions. Objective comparison of detectors, while a desirable goal, is almost impossible in practice without access to the programs used by authors to build the models and perform the detection. However, one can have a rough idea about the relative time performance of various methods by looking at the complexity of the detection task, which in case of Shneiderman and Kanade [5] comprises several intensity correction steps followed by a complex wavelets-based network. That is why we consider that the proposed method being faster and better suited for a low-latency system. We give in next section numerical comparisons of processing speed.

4.4 Processing speed

As noted in Sect. 2.3, computing the response of a Gaussian filter was roughly three times more expensive than applying a Haar filter. Moreover the parallelization technique also increases the processing time as all the candidate windows are tested by each classifier in the mixture. However we show in this section that the overall detection speed in not significantly reduced by these two contributions.

Let us consider four detectors, all pre-processed by a cascade made of 5 stages of HF: a cascade of 7 other stages of HF, a cascade of 12 stages of GF, 1 stage of 500 GF, a mixture of GF. We apply these four detectors on a sequence of 1,500 images with 320 × 240 pixels, each frame containing one or several faces. We then report in Table 3 the average detection speed (number of frames per seconds).

Detectors with GF are only slightly slower than the one only based on HF. Moreover the speed of the GF-based detectors does not depend on the structure of the system after the pre-processing step. Using a cascade of GF, a single stage of GF or a mixture of GF lead to roughly equivalent detectors in terms of computation complexity. In fact the 5 stages of HF discard a large majority of non-face windows so that the computation of the GF does not affect much the overall detection speed.

Table 3 Detection speed in frames per seconds (fps) of four detectors. The measure is an average over the 1,500 frames of a sequence of 320 × 240 pixels images

Detector	fps
12 stages HF	28.63
5 stages HF + 12 stages GF	27.32
5 stages HF + 1 stage GF	27.18
5 stages HF + mix of GF	27.22

5 Conclusions

This paper presents a new face detection system using a combination of several boosted classifiers which leads to high detection performances and can be applied in real-time. Each classifier is trained using local discriminant features based on anisotropic Gaussian filters. The complete training set is randomly subsampled and separate classifiers are trained on each subsets. They are then combined probabilistically. It has been shown that the mixture of boosted classifiers decreases significantly the false positive rate without affecting the true positive rate. The complete system has been tested on reference datasets and compared favorably to state-of-the-art. In a future work, this method will be extended to multi-pose face detection and pose estimation as the Gaussian features seem to be also very discriminant for poses others than frontal.

Acknowledgments This work is supported by the Swiss National Science Foundation through the National Center of Competence in Research on “Interactive Multimodal Information Management (IM2)”.

References

1. Rodriguez, Y, Cardinaux, F., Bengio, S., Mariéthoz, J.: Estimating the quality of face localization for face verification. In: Proc. IEEE International Conf. on Image Processing, vol. 01, pp. 581–584 (2004)
2. Hjelmas, E., Low, B.K.: Face detection: a survey. Comput. Vision Image Understanding **83**, 236–274 (2001)
3. Yang, M., Kriegman, D.J., Ahuja, N.: Detecting faces in images: a survey. IEEE Trans. Pattern Anal. Machine Intell. **24**(1), 34–58 (2002)
4. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Trans. Pattern Anal. Machine Intell. **20**(1), 39–51 (1998)
5. Schneiderman, H., Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Los Alamitos, pp. 746–751 (2000)
6. Rowley, H.A., Baluja, S., Kanade, T.: Human face detection in visual scenes. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) Advances in Neural Information Processing Systems, vol. 8, pp. 875–881. The MIT Press (1996)
7. Viola, P., Jones, M.J.: Robust real-time face detection . Int. J. Comput. Vision **57**(2), 137–154 (2004)
8. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proc. IEEE International Conf. on Image Processing, pp. 900–903 (2002)
9. Wu, B., Ai, H., Huang, C., Lao, S.: Fast rotation invariant multi-view face detection based on real adaboost. In: 6th IEEE International Conf. on Automatic Face and Gesture Recognition, pp. 79–84 (2004)
10. Li, S.Z., Zhang, Z.: Floatboost learning and statistical face detection. IEEE Trans. Pattern Anal. Machine Intell. **26**(9), 1112–1123 (2004)
11. Meynet J., Popovici V., Thiran J.P.: Face class modeling using mixture of SVMs. In: International Conf. in Image Analysis and Recognition ICIAR, Porto, Portugal, vol. 3212, pp. 709–716 (2004)
12. Meynet, J., Popovici, V., Thiran, J.P.: Combining svms for face class modeling. In: 13th European Signal Processing Conference—EUSIPCO, Antalya, Turkey (2005)
13. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Machine Intell. **20**(3), 226–239 (1998)
14. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. System Sci. **55**(1), 119–139 (1997)
15. Peotta, L., Granai, L., Vandergheynst, P.: Very low bit rate image coding using redundant dictionaries. In: Proc. of the SPIE, Wavelets: Applications in Signal and Image Processing X, vol. 5207, pp. 228–239 (2003)
16. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
17. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)
18. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Ann. Statist. **28**, 337–407 (2000)
19. Kuncheva, L.I.: Combining Pattern Classifiers Methods and Algorithms. Wiley, New York (2004)
20. Duin, R.P.W., Tax, D.M.J.: Experiments with classifier combining rules. In: Multiple Classifier Systems (2000)
21. Messer, K., Matas, J., Kittler, J., Luettin, J., Maitre, G.: XM2VTSDB: the extended M2VTS database. In: Second International Conf. on Audio and Video-based Biometric Person Authentication, pp. 72–77. Washington (1999)
22. Frischholz, R.W., Dieckmann, U.: Bioid: a multimodal biometric identification system. IEEE Comput. **33**(2), 64–68 (2000)
23. Phillips, P.J. et al. The FERET database and evaluation procedure for face-recognition algorithms. Image Vision Comput **16** (1999)
24. Popovici, V., Thiran, J.P., Rodriguez, Y., Marcel, S.: On performance evaluation of face detection and localization algorithms. In: Kittler, J. (ed.) Proc. of the 17th International Conf. on Pattern Recognition, vol. 1, pp. 313–317 (2004)
25. Bailly-Bailliere, E. et al.: The banca database and evaluation protocol. In: 4th International Conf. on Audio- and Video-Based Biometric Person Authentication, Guildford, Lecture Notes in Computer Science, vol. 2688, pp. 625–638. Springer, Heidelberg (2003)
26. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE Trans. Pattern Anal. Machine Intell. **20**(1), 23–38 (1998)