Pattern Anal Applic (2010) 13:197–211
DOI 10.1007/s10044-009-0150-5

THEORETICAL ADVANCES

The architecture and performance of the face and eyes detection system based on the Haar cascade classifiers

Andrzej Kasinski · Adam Schmidt

Received: 11 February 2008 / Accepted: 15 December 2008 / Published online: 3 March 2009

Abstract The precise face and eyes detection is essential in many human–machine interface systems. Therefore, it is necessary to develop a reliable and efficient object detection method. In this paper we present the architecture of a hierarchical face and eyes detection system using the Haar cascade classifiers (HCC) augmented with some simple knowledge-based rules. The influence of the training procedure on the performance of the particular HCCs has been investigated. Additionally, we compared the efficiency of other authors' face and eyes HCCs with the efficiency of those trained by us. By applying the proposed system to the set of 10,000 test images we were able to properly detect and precisely localize 94% of the eyes.

Keywords Face detection · Eyes detection · Haar cascade classifiers

1 Originality and contribution

The main contribution of the work reported in this paper is the synthesis of a fast and highly successful face and eyes detection system based on the Viola's Haar cascade classifiers (HCC). More specifically our research gave the following results:

- the influence of the particular HCC's training parameters and the complexity of the training set on the detectors efficiency was identified;
- the face and eyes detectors outperforming the publicly available HCCs w.r.t. both the accuracy and the processing time were trained;
- the regionalized search concept and the simple rule regarding the in-plane rotation of eye pairs were used greatly reducing the false-positive ratio and the computational cost;
- the experiments were conducted on a new, extensive database of almost 10,000 images which is enough to provide statistically significant results.

2 Introduction

In recent years significant attention has been paid to the task of automatic face recognition (FR). However, in order to be efficient, many of the proposed algorithms require the proper initialization. Providing information on the precise face location, in-plane rotation and scale is essential for achieving high performance. The required data can be easily obtained by detecting not only the face but also the eyes of a person. As a result, the face and eyes detection is the first processing step in many automatic face recognition systems and plays important, yet often neglected, role in their operation.

The influence of the eyes localization error on the performance of some FR methods has been investigated by Campadelli et al. [1]. She also concluded that some of the published FR results do not clearly state the fact of manual initialization, which greatly improved the performance reported by the particular authors.

A. Kasinski (✉) · A. Schmidt
Institute of Control and Information Engineering,
Poznan University of Technology, ul. Piotrowo 3a,
60-965 Poznan, Poland
e-mail: Andrzej.Kasinski@put.poznan.pl

A. Schmidt
e-mail: Adam.Schmidt@put.poznan.pl

Recently the HCC introduced by Viola [2] have been successfully applied to the face detection task. Reported high detection ratio and computational efficiency suggested the possibility of using the HCC in a reliable real-time face and eyes detection system. Therefore, our goal was to design efficient face and eyes HCC detectors and to combine them into a hierarchical system. To improve the accuracy of the system some additional, knowledge-based criteria were introduced. Furthermore, the influence of the weak classifiers complexity, of the desired cascade stages detection ratios and of the strategy for creating the negative training set on the overall system performance was assessed.

The paper is organized as follows. Firstly, we review the state of the art in the field of face and eyes detection. The principles of the HCC are presented in Sect. 4. The architecture of the proposed system is described in Sect. 5 followed by the description of the conducted experiments environment. The procedure of training the proposed detectors as well as the other authors' detectors used in the experiment are described in Sect. 7. Afterwards, we present the obtained experimental results and conclude the paper with Sect. 9.

3 The state of the art in the face and eyes detection

3.1 Face detection

A human face is a flexible 3D object whose image is strongly influenced by both pose and expression variations. This combined with the diversity of personal face features and possible structural disturbances (such as glasses, facial hair, make-up) significantly hinders the detection task. As there are numerous different approaches to the task of the face and eyes detection we present only a brief review of the selected ones.

Kotropoulos and Pitas [3] proposed a hierarchical, rule-based system for the face localization. The input image was scanned for a 6 × 7 pixels rectangle conforming to a defined set of rules. The search procedure was then repeated for different image resolutions. After a successful detection, another set of rules was used to determine the positions of eyes, eyebrows, nostrils and mouth.

The algorithm presented by Hsu [4] was based on the color information. After converting the input image to the YCbCr color space, the regions with color similar to that of the human skin were extracted. Then, also using color information, the regions possibly corresponding to the eyes and mouth were detected inside the face candidates. The detection was claimed if a given face candidate contained both eyes and mouth.

The system proposed by Heisele et al. [5] consisted of two stages. Firstly, three independent support-vector machines (SVM) detected potential eyes, nose and mouth regions. Then, the second-level classifier checked if their relative position could correspond to that typical for the human face. The system was further improved by training the first-level SVMs not against the diverse negative set but only against other facial features [6].

Su and Chou [7] applied the associative memories to the task of the face detection. They have trained two memories: the first one by using the gray-scale face images and the second one by using the edge-images. The image regions under consideration were treated as an input of both memories. If the similarity measure between the input and both memories outputs was high enough, the investigated region was considered to be a face. To speed-up the detection some image regions were discarded during the preprocessing. This happened if the mean and the variance of the illumination did not fit certain ranges.

Rowley et al. [8] presented a detection system based on neural networks with retinal connections and overlapping receptive fields. The number of false detections was successfully reduced by requiring that the face should be detected by several networks trained with different starting weights. An extra condition was to have multiple positive responses in several neighboring rectangles. Moreover, the authors presented a solution to the problem of the non-representative negative training set. At the successive training steps the neural network was tested and false positives were added to the negative training set.

Huang et al. [9] presented an algorithm using the polynomial neural network (PNN). PNN is a single-layer network taking the polynomial expansion of pattern features as inputs. Three separate feature pools were created: the first based on pixel intensity values, the second based on Sobel filter responses, and the last one using directional gradient decomposition. The principal components analysis (PCA) was then used to reduce the dimension of the features vector. It was proved that the system based on gradient decomposition outperformed the systems using simpler features.

Viola and Jones [2] were first to introduce the HCC and to use them in the task of face detection. Creating a cascade of boosted classifiers resulted in a fast and precise detection system. The idea has been further improved by Lienhart et al. [10], who has enlarged the feature pool with the rotated Haar-like features.

The system proposed by Meynet et al. [11] was also based on the weak classifiers ensembles. In this paper the HCC was used as a first stage of processing and discarded these non-faces which were easy to classify. The remaining detection windows were tested with a set of parallel boosted classifiers using the anisotropic Gaussian features.

The final classification depended on the voting of those classifiers.

3.2 Eyes detection

Only a few algorithms detect eyes directly in the input image. In most of the cases eyes are looked for on the already localized faces, which significantly facilitates the detection task. As a result, the eyes detector must only discriminate between the eyes and other facial features. However, the errors of the face detectors are passed on and affect the final results of the eyes detection.

Wang et al. [12] used the homomorphic filtering to compensate for illumination variations. After that, the binary template matching was applied to the preprocessed images in order to extract potential eyes. The candidate regions were verified with the SVM and the precise eyes location was acquired with the variance filters.

The detection scheme proposed by Kumar et al. [13] was based on the notion that a face contains two eye regions which are darker than their surroundings. Thresholding in the HSV and the normalized RGB color spaces was used to detect the regions with low intensity and color similar to that of the human skin. In the next step the regions with the aspect ratio strongly differing from 0.75 were discarded. Eye pairs were created from the regions matching the rules addressing the between-eyes distance and the in-plane rotation. The final verification was based on the analysis of the mean value and the variance of the intensity values in the columns of the rectangle containing the candidate eye pair.

Peng et al. [14] have proposed an algorithm for the localization of the eyes on a frontal image of a face without glasses. Firstly, they computed the gradient image and its vertical and horizontal projections. Two maxima of the vertical gradient projection corresponded to the face border and enabled assessing its width. The region in the upper face with the high variability of the horizontal gradient projection should contain eyes. The additional verification was based on matching with the template scaled to fit the estimated face width.

The algorithm devised by Wu and Zhou [15] was based on finding so called eye analogues. The authors noticed that eyes and eyebrows are darker than their surrounding (face as a local background). Thus, they searched for pixels darker than their neighborhood and grouped them. The clusters whose shape or aspect ratio ruled out being an eye analogue were discarded. The remaining regions were matched into pairs if they lay on a horizontal line at the appropriate distance. To confirm the detection, regions surrounding such pairs were normalized and compared with the template.

Campadelli et al. [16] have used the Haar wavelet decomposition for the eyes detection. The decomposition coefficients served to train two SVMs. The first one was used to validate the face detection and to roughly detect eyes, the second one precisely localized the eyes.

The SVMs have been also used by Arandjelovic and Zisserman [24]. They used feature vectors consisting of the image intensity and gradient. The surrounding of the manually marked eyes and mouth regions was deformed with random affine transformations to increase the number of training examples. The trained SVMs were applied to subregions of the previously found faces and the mean of the largest cluster was considered to be the final feature location.

Wavelet decomposition has also been used by Motwani et al. [17]. They noticed that the intensity of the eyes strongly differs from the intensity of the surrounding regions, which resulted in the large decomposition coefficients. Firstly, the maxima of the decomposition coefficients were found. Then the detection was verified by using the neural network, which took coefficients neighboring the maximum as inputs.

The detection system proposed by Tivive and Bouzerdoum [18] was based on the convolutional neural network with two hidden layers and a linear output neuron. The network took a 32 × 32 pixels image rectangle as an input. The authors have claimed that they achieved 99% accuracy, however they have not presented the error measure used.

Bianchini and Sarti [19] have used the auto-associative neural networks. Their system was based on the analysis of the horizontal and vertical projections of the image gradient with two separate auto-associators. The detection was based on scanning fragments of projections and fusing the detections in both axes. The authors have admitted that their algorithm can be used only for the localization of frontal face views on a uniform background.

Many authors tried to use the HCC in the task of eyes detection. Wilson and Fernandez [20] used the specialized cascades trained against other facial features to extract the eyes, mouth and nose from the face region. They have also introduced the regionalized search approach, which explicitly means using the knowledge about the face structure, i.e. looking for the left eye in the upper-left, for the right eye in the upper-right, the nose in the central and the mouth in the lower part of the face.

Feng et al. [21] used the HCC at the first stage of their detection system. As the second stage they have used a classifier based on ordinal features rather than on Haar-like ones and trained with an algorithm similar to the AdaBoost.

Wang et al. [22] concluded that the rectangular Haar-like features are not precise enough to describe eyes, which are apparently elliptical. They have decided to statistically define features minimizing the Bayes rule classification error. The selections have been based on the recursive nonparametric discriminant analysis. As a result the cascade of two detectors have been created. The first one used only two features and discarded 80% of non-eyes, the second one used almost 100 features and was used for the precise classification. The eyes were looked for only in the upper face and the neighboring detections were averaged.

In their work Everingham and Zisserman [23] compared three different approaches to the task of the eyes localization. The first method used the kernel ridge regression to predict the eyes positions in the image. The second approach used probabilistic appearance models of eyes and non-eyes. The output of the detector was the log-likelihood ratio at each image pixel. The image patch with the greatest log-likelihood was considered the eye position. The last method was based on the HCC. The single stage classifier using the Haar-like features was trained using bootstrapping. The best results were obtained with the Bayesian approach which localized 90% of the eyes with the maximum error of 2 pixels. The other two methods performed only slightly worse.

4 The Haar cascade classifiers

The HCC detector proposed by Viola [2] is a successful combination of three basic ideas. Firstly, an extensive set of features which can be computed in a short and constant time is used. This feature-based approach helps to reduce the in-class variability and increases the variability between classes. Secondly, applying a boosting algorithm allows the concurrent selection of the salient features and the classifier training. Finally, forming a cascade of gradually more complex classifiers results in a fast and efficient detection scheme.

4.1 Haar-like features

According to Lienhart [10], any Haar-like feature in a $W \times H$ pixels detection window is defined by the following equation:

$$\text{feature} = \sum_{i=1}^{N} \omega_i \cdot \text{RecSum}(r_i) \tag{1}$$

where ω_i is an arbitrarily chosen weighting factor and RecSum(r_i) is the sum of intensity values over any given upright or rotated rectangle placed inside a detection window. A rectangle is described by five parameters $r = (x, y, w, h, \phi)$ where x and y are the coordinates of

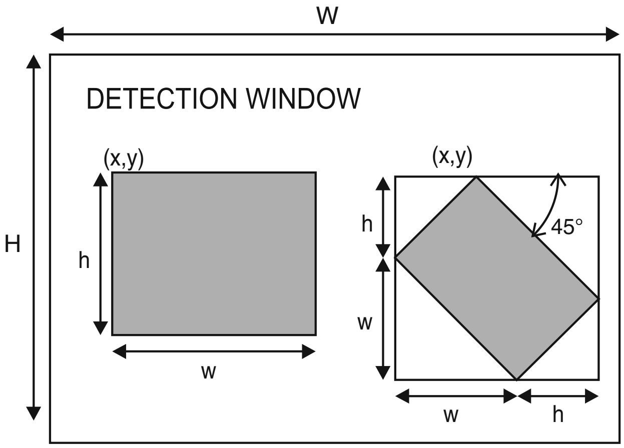

Fig. 1 Upright and 45° rotated rectangles in the detection window

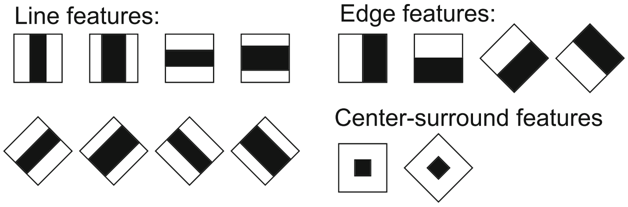

Fig. 2 The prototypes of Haar-like features

upper-left corner, w and h define the dimensions of the rectangle and $\phi = \{0°, 45°\}$ stands for the rotation angle (Fig. 1).

Using Eq. 1 leads to the almost infinite features pool. To reduce their number the following restrictions are applied:

- Pixel sums over only two rectangles are allowed ($N = 2$).
- The weights are used to compensate for the area difference of two rectangles and have opposite signs, which means that $-\omega_1 \cdot \text{Area}(r_1) = \omega_2 \cdot \text{Area}(r_2)$. Substituting $\omega_1 = -1$ one gets $\omega_2 = \text{Area}(r_1)/\text{Area}(r_2)$.
- The features should be similar to those used in the early stages of the human vision pathway.

Those constraints leave 14 prototype features (Fig. 2), which can be scaled in both directions and placed in any part of the detection window. This allows to create an extensive, but finite, feature pool. The features are calculated as the difference of pixel's intensity sum under the black rectangle and under the white one scaled to compensate for the areas difference. It is worth mentioning that the line features can also be computed as a combination of two rectangles: one of them containing both black and white, but the second one contains only a black area.

To efficiently evaluate features, two auxiliary image representations are employed. The summed area table

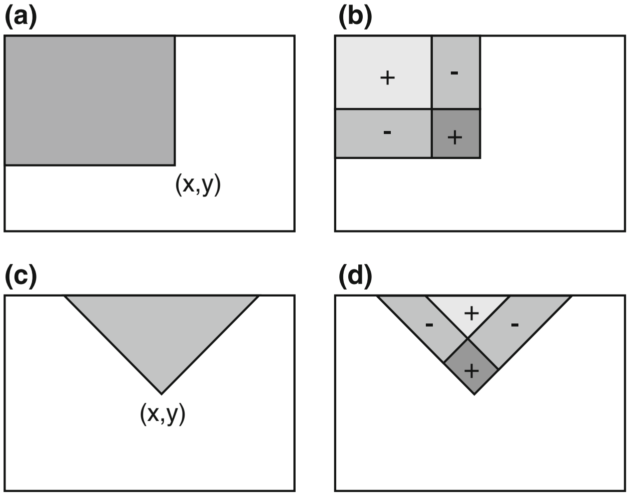

Fig. 3 Auxiliary image representations: **a** the idea of SAT, **b** fast feature calculation using SAT, **c** the idea of RSAT, **d** fast feature calculation using RSAT

(SAT(x, y)) [2] is used for the fast computation of the features based on the upright rectangles. Here, each entry of the table is defined as the sum of pixel intensities under the upright rectangle spanning from (0, 0) to (x, y) and is being filled according to the formula (Fig. 3a):

$$\mathrm{SAT}(x,y) = \sum_{x' \le x, y' \le y} I(x',y') \tag{2}$$

where $I(x, y)$ is the intensity value of pixel (x, y).

The whole table can be computed in a single pass using the following formula:

$$\begin{aligned}\mathrm{SAT}(x,y) =& \mathrm{SAT}(x,y-1) + \mathrm{SAT}(x-1,y)\\ &+ I(x,y) - \mathrm{SAT}(x-1,y-1)\end{aligned} \tag{3}$$

with SAT(−1, y) = SAT(x, −1) = SAT(−1, −1) = 0 for any x and y.

Once filled, the SAT enables computation of RecSum(r) for any upright rectangle $r = (x, y, w, h, 0°)$ with only four look-ups (Fig. 3b):

$$\begin{aligned}\mathrm{RecSum}(r) =& \mathrm{SAT}(x-1,y-1)\\ &+ \mathrm{SAT}(x+w-1,y+h-1)\\ &- \mathrm{SAT}(x+w-1,y-1)\\ &- \mathrm{SAT}(x-1,y+h-1)\end{aligned} \tag{4}$$

The rotated features are computed using another auxiliary representation called the rotated summed area table (RSAT(x, y)) [10]. Each entry is filled with the following value (Fig. 3c):

$$\mathrm{RSAT}(x,y) = \sum_{|x-x'| \le y-y', y' \le y} I(x',y') \tag{5}$$

RSAT can be iteratively filled according to the formula:

$$\begin{aligned}\mathrm{RSAT}(x,y) =& \mathrm{RSAT}(x-1,y-1) + I(x,y-1)\\ &+ \mathrm{RSAT}(x+1,y-1) + I(x,y)\\ &- \mathrm{RSAT}(x,y-2)\end{aligned} \tag{6}$$

where RSAT(−1, y) = RSAT(x, −1) = RSAT(−1, −1) = RSAT(x, −2) = RSAT(−1, −2) = 0 for any x and y.

The pixel sum of any rotated rectangle $r = (x, y, w, h, 45°)$ can be computed according to (Fig. 3d):

$$\begin{aligned}\mathrm{RecSum}(r) =& \mathrm{RSAT}(x-h+w,y+w+h-1)\\ &- \mathrm{RSAT}(x-h,y+h-1)\\ &- \mathrm{RSAT}(x+w,y+w-1)\\ &+ \mathrm{RSAT}(x,y-1)\end{aligned} \tag{7}$$

4.2 Classifiers cascade

Usually the object of interest occupies only a small part of the analyzed image. Thus it is better to quickly discard the non-object regions and to focus only on those which are relevant, than to examine every window thoroughly. Creating a cascade structure enables such an approach. The cascade classifier consists of the N stages, i.e. of the serially connected classifiers distinguishing between the detected object and the background. Each stage is trained to achieve the true positive (TP) ratio p while having false positive (FP) ratio of at most f. The positively classified image windows are passed to the subsequent stage; the others are excluded from the further processing.

Due to the serial nature, the overall detection ratios are exponential function of the single stage efficiencies:

$$\mathrm{TP}_{\mathrm{cas}} = \prod_{i=1}^{N} p_i \approx p^N \tag{8}$$

$$\mathrm{FP}_{\mathrm{cas}} = \prod_{i=1}^{N} f_i \approx f^N \tag{9}$$

where $\mathrm{TP}_{\mathrm{cas}}$ is a TP ratio and $\mathrm{FP}_{\mathrm{cas}}$ is a FP ratio of the cascade (Fig. 4).

The adequate selection of p (usually set close to 1), f (usually 0.5) and N results in a detector preserving a high TP ratio (slightly less than 100%) with a FP ratio converging to 0 at the same time.

The stages are consecutively trained to achieve the desired detection rates. Only the first classifier is presented

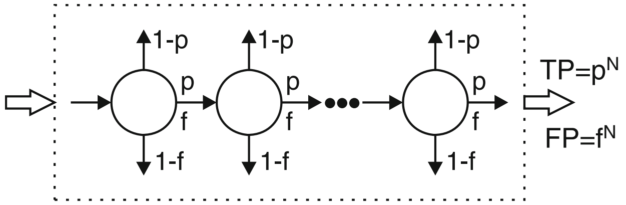

Fig. 4 The structure of the cascade detector

with the whole sets of the positive and negative samples. The others are trained only on the subsets which have passed the previous stages. As a result, the classifiers at the successive stages are faced with more challenging tasks and have to discover subtler differences to maintain the desired p and f ratios.

4.3 Single stage classifier

Using such a numerous feature pool requires a method of selecting the sufficient subset of the salient features. Boosting is a machine learning meta-algorithm proposed by Freund and Schapire [25]. It is used to aggregate many simple weak classifiers into an ensemble outperforming its components. The only assumption regarding the weak classifiers is that they must achieve the misclassification ratio less than 50% in any training set. Any type of classifier can be used as a weak classifier. The ensemble is created by iteratively adding the weak classifiers trained on the weighted examples set, followed by reweighting the training set according to the current performance of the ensemble.

In the HCC simple classification and regression trees (CART) [26] are used as weak classifiers. If the decision tree is used only for classification purposes its output is always a class label. Using CART results in responses being real numbers, which (especially in a two-class decision problem) can be viewed as certainty measures. The Gini impurity index is used for choosing the best splits in the tree nodes. As the size of the trees used is restricted to only several splits no tree pruning is applied.

In the simplest case (single-split CARTs called "stumps") the weak classifiers rely on a single feature only. Using slightly more complex classifiers (e.g. four-split CARTs) slows down the training but allows to preserve some relations between features encoded in a structure of a weak classifier. Even those more complex classifiers could not be sufficient to achieve the desired detection rates. To assemble weak classifiers the AdaBoost [25] boosting algorithm is used. In [10] Lienhart et al., proved that using the version called the Gentle AdaBoost results in creating a detector having a lower FP ratio than those created with other AdaBoost variants.

Gentle Adaboost algorithm specification according to [27]:

```
1. Given the TP ratio $p$, the FP ratio $f$ and $N$ examples $(x_1, y_1), \ldots, (x_N, y_N)$ where $x_i \in R^k$, $y_i \in \{-1, 1\}$
2. Start with weights $w_i = 1/N$, $i = 1, \ldots, N$
3. Repeat until $p$ and $f$ are achieved
    a. Fit the regression function $f_m(x)$ minimizing the expression $\sum_{i=1}^{N} w_i(y_i - f_m(x_i))^2$
    b. Set $w_i = w_i \exp(-y_i f_m(x_i))$
4. Output the classifier: $F(x) = \mathrm{sign}\left[\sum_{m=1}^{M} f_m(x)\right]$
```

4.4 Detection procedure

Due to the fact that any rectangle sums can be computed with a constant number of look-ups, the Haar-like features can be calculated using the same SAT and RSAT representations (Sect. 4.1) regardless of the scale. As a result the multi-resolution search is done via feature scaling rather than image scaling and resampling, which significantly speeds-up the process. For the further performance increase the minimum detection size, larger than the original cascade detection window, can be specified.

The object of interest usually triggers many detections in the image. The rectangles which have passed through the cascade are grouped according to the following criteria:

- the Chebyshev distance $(D_{\mathrm{Cheb}}(p,q) = \max_i(|p_i, q_i|))$ between the upper-left corners of the two rectangles cannot exceed 0.2 of the first rectangle width,
- the width of any rectangle cannot exceed 1.2 of any other rectangle width.

The rectangles in each group are averaged and constitute a single detection result. The number of the regions merged, called the neighbors number *Nbhd*, is preserved and can be used as a measure of the detection certainty. The selectiveness of the cascade can be adjusted by increasing the minimum number of the merged regions sufficient to declare a valid detection. Setting the appropriate value of the *Nbhd* can significantly improve the performance of the detector (Fig. 5). Moreover, the regions lying inside other detected rectangles and having the lower neighbors count are discarded.

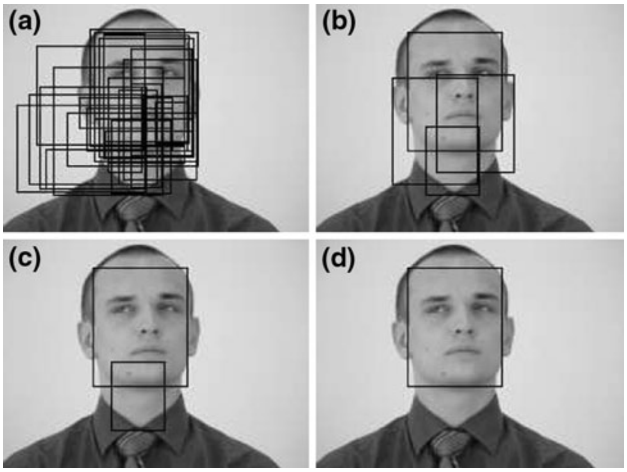

Fig. 5 The influence of the minimum neighbors number on the detection results: **a** *Nbhd* = 0, **b** *Nbhd* = 1, **c** *Nbhd* = 2, **d** *Nbhd* = 7

5 The system architecture

The proposed detection system consists of the three stages. At the first one the HCC face detector is applied to the whole image. The HCC is fine-tuned by setting the appropriate constraint on the minimum face neighbors number *NbhdF*. The detected face candidate regions are further processed independently.

At the second stage the left and right eye HCC detectors are used on the previously found face regions. For each face region two lists are created. One stores the left eye regions found, the second one stores the right eye regions. The constraint on the minimum number of the merged eyes neighbors *NbhdE* can be set. Moreover, instead of searching for eyes over the whole face, the regionalized search [20] can be used. This means, applying the left eye detector to the rightmost and the right eye detector to the leftmost 60% of the upper half of the face. Proportions of those subregions allow the correct eyes detection under varying face pose while restricting the possibility of falsely detecting some other facial features.

The third stage is a simple knowledge-based rule of combining left and right eye detections into the valid eye pairs. For each left and right eye combination in a given face rectangle an in-plane rotation ϕ is calculated. The eye pairs with $|\phi| > 20°$ are discarded, as too unlikely to belong to the upright view of the face. Face candidates with no eye pair found are also discarded.

6 The experiments environment

In order to get statistically significant results, the performance evaluation of our face and eyes detection system was conducted on a set of face images consisting of almost 10,000 images of 100 people . The images were acquired in partially controlled illumination conditions, over uniform background, and stored as 2,048 × 1,536 pixels JPEG files. The pictures of each person were taken in the following sequences, while:

- turning their head from the right to the left,
- nodding their head from the raised to the lowered position,
- turning their raised head from the right to the left,
- turning their lowered head from the right to the left,
- moving their head without any constraint on the face pose.

The main goal of creating such an extensive image base was to provide credible data for the systematic evaluation of the face detection, facial features extraction and FR algorithms performance. In order to provide a the ground truth for the face and eyes detection tasks the rectangular ROIs containing face and eyes were manually marked on the each image in the base . For each image the coordinates and dimensions of the rectangles bounding the face and eyes were saved in the OpenCV Storage files in the YAML format. All the face ROI rectangles were adjusted to have the aspect ratio of 0.8. The eye rectangles have the aspect ratio equal to 1.8. Figure 6 presents some exemplary pictures from the image base.

The proposed system was implemented in the Visual C++ 6.0 with the Open Computer Vision Library (OpenCV) [28]. We have used Lienhart's implementation of the HCC. All the detectors were trained using the tools included in the OpenCV.

7 Detectors training

In order to obtain a precise detection system we had to identify the influence of the various training parameters on the HCC performance. Changing the basic parameters such as the complexity of the weak classifiers and the required p ratio of the single stage classifiers was an obvious choice. Moreover, we tested the influence of the training set diversity on the HCC performance. The HCC training requires large and diverse sets of positive and negative samples (images). With our new image base it was easy to get the examples of the faces and eyes. The positive training sets were not changed during the experiment and were created using the images of the 50 first people. The input pattern size

Fig. 6 Examples from the image base

was set to 20 × 25 pixels for the face detectors and 18 × 10 pixels for the eyes detectors. As the eyes detector is applied to the previously detected faces, the negative, non-eyes training set was created from the face images with the left or right eye occluded. However, creating a set of "non-faces" is a tricky task, because what does it mean to represent an every possible non-face object? As all images in our base were taken on the same, uniform background, we wanted to check whether the negative set built from the same images with occluded faces is sufficient enough to distinguish between faces and non-faces. Another negative training set was created by randomly gathering about 3500 diverse pictures not containing any faces. Figure 7 presents some examples from the negative training sets. As Lienhart showed that Gentle AdaBoost gave better results than other AdaBoost versions [10], all the detectors were trained using the Gentle AdaBoost. The required theoretical *FP* ratio of the cascades was set to 10e-6. The following face detectors have been trained:

- Face1: occluded faces negative set, $p = 0.995$ and stump as a weak classifier
- Face2: occluded faces negative set, $p = 0.995$ and two-split CART as a weak classifier
- Face3: occluded faces negative set, $p = 0.995$ and four-split CART as a weak classifier
- Face4: rich negative set, $p = 0.990$ and four-split CART as a weak classifier
- Face5: rich negative set, $p = 0.995$ and four-split CART as a weak classifier
- Face6: rich negative set, $p = 0.999$ and four-split CART as a weak classifier

Only a single parameter at a time was modified during the experiments. The variant giving the best results was used in the subsequent tests. The C1, C2 and C3 detectors differ in the complexity of the weak classifiers used. The C3 was trained with the occluded faces negative set, while the C4 was trained using the rich negative set. The C4, C5 and C6 HCCs vary in the required TP ratio of a single-stage classifier.

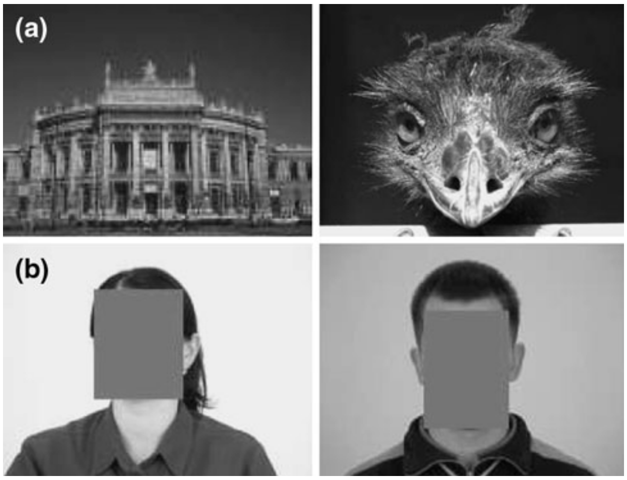

Fig. 7 Examples from the negative training sets: **a** diverse set, **b** occluded faces

Their performance was then compared to the performance of the following Lienhart's detectors available with the OpenCV:

- Lienhart1: stump-based, 24 × 24 window, trained with the Discrete AdaBoost
- Lienhart2: stump-based, 20 × 20 window, trained with the Gentle AdaBoost
- Lienhart3: two-split CART-based, 20 × 20 window, trained with the Gentle AdaBoost
- Lienhart4: two-split CART-based, 20 × 20 window, trained with the Gentle AdaBoost, with a tree made of stage classifiers instead of a cascade

Independently, the following eyes detectors were trained and their results were then compared with the results of Castrillón-Santana's detector (Santana) [29]:

- Eyes1: $p = 0.995$, with four-split CART as a weak classifier
- Eyes2: $p = 0.999$, with four-split CART as a weak classifier

8 Results

8.1 Face detection

Lienhart's and our detectors have been applied to the whole image base. The minimal detection window's size was set to 400 × 500 pixels for our detectors and 400 × 400 for Lienhart's. The detectors distributed with the OpenCV were trained on square windows. To assure the compatibility of the results their outputs aspect ratio was reduced to 0.8. The results for the whole image base and the subset of the 2,193 pictures containing only frontal faces are presented separately.

The face detection efficiency measure should reflect both the size difference between the detected window and the ground truth ROI as well as the displacement. If the intersection area of both the detected and the ground truth rectangles was greater than 80% of both rectangles areas a TP was claimed, otherwise the case was considered to be a FP (Fig. 8). If no face was found on the whole picture, the result was declared a false negative. As there were no images without any face, the true negative outcome was not possible.

The obtained results clearly show that increasing the complexity of the weak classifiers significantly improves

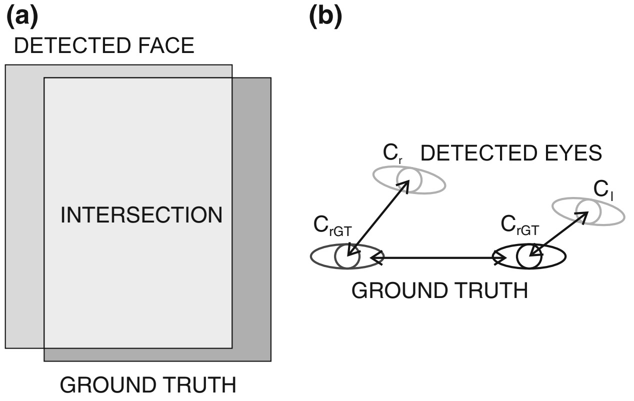

Fig. 8 Detection correctness measures: **a** face detection, **b** eyes detection

the detection ratio. The detector using the four-split CART achieved the higher TP ratio and lower FP ratio than other detectors using simpler trees (Figs. 9, 10).

Despite the uniform background of the images used, the detectors trained using the diverse negative training set gave better results (Figs. 11, 12). This is especially visible in the case of frontal face images.

Heightening the required *p* ratio of the single stage classifiers strongly influenced the performance of the whole cascade (Figs. 13, 14). The detectors with *p* closer to 1 achieved the higher TP ratio and had the FP ratio decreased. It can be explained with the exponential nature of the cascade efficiency. Increasing the difference between *p* and *f* allows preserving the higher TP ratio while the FP converges to 0.

The FP ratio can be greatly reduced by increasing the minimum number of the merged face detections (*NbhdF*). However, it should be pointed out that only increasing it to the number of 5 gave positive results. The further increase of *NbhdF* parameter resulted in the quick deterioration of the TP ratio without any significant change in the FP ratio. The difference in achieved results between the whole image base and its subset of the frontal images shows that the images with non-standard face poses are detected with a lower confidence ratio.

The comparison with the Lienhart's face detectors showed that the detector trained on our image base outperformed other available solutions. The difference is evident in the case of the whole image base (Fig. 15). But even for the frontal faces, the Face6 HCC achieved the higher TP and the lower FP ratio than any of Lienhart's detectors (Fig. 16). Moreover, our detectors were twice as efficient w.r.t. the processing time than the best detectors available with the OpenCV (Table 1).

8.2 Eyes detection

The performance of the Castrillón-Santana's detectors and ours has been tested on the manually marked faces and the faces automatically detected with Face3, Face6, Lienhart1

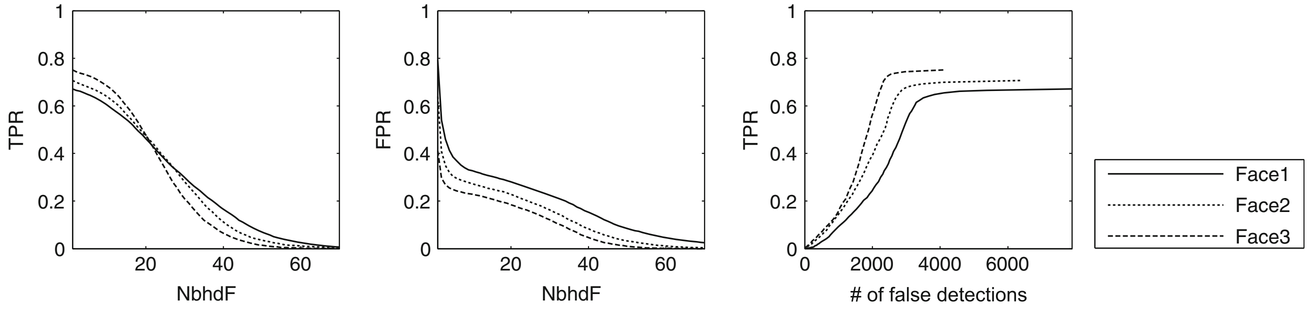

Fig. 9 The influence of weak classifier's complexity on the face detector's performance for the whole image base

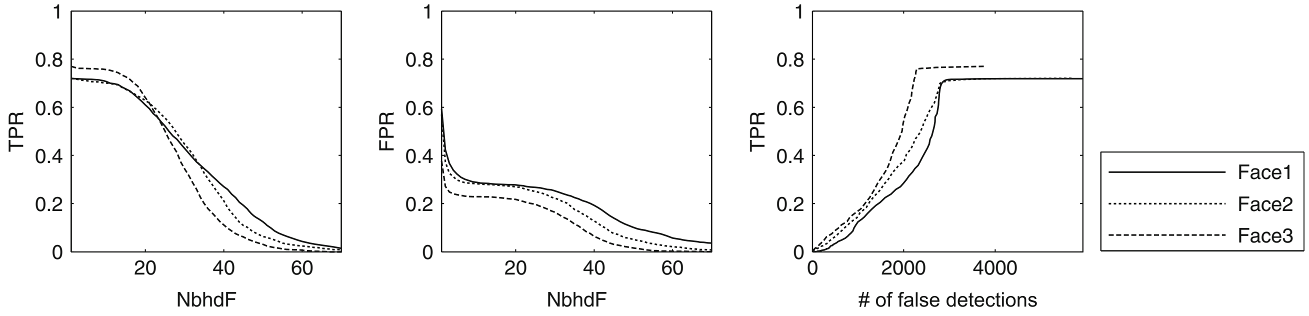

Fig. 10 The influence of weak classifier's complexity on the face detector's performance for the frontal face images

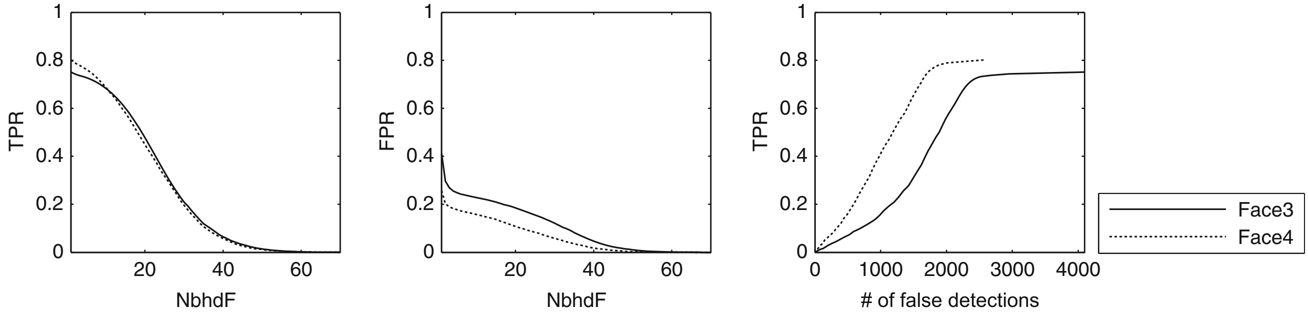

Fig. 11 The influence of negative training set diversity on the face detector's performance for the whole image base

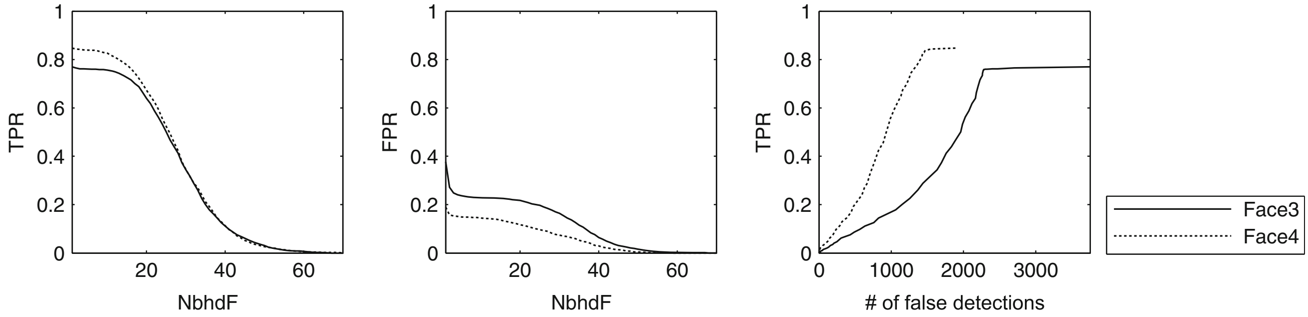

Fig. 12 The influence of negative training set diversity on the face detector's performance for the frontal face images

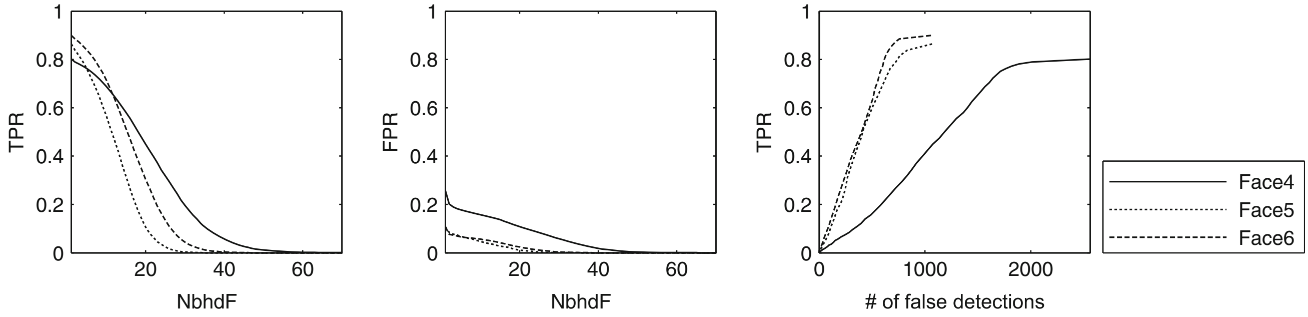

Fig. 13 The influence of the single stage *p* ratio on the face detector's performance for the whole image base

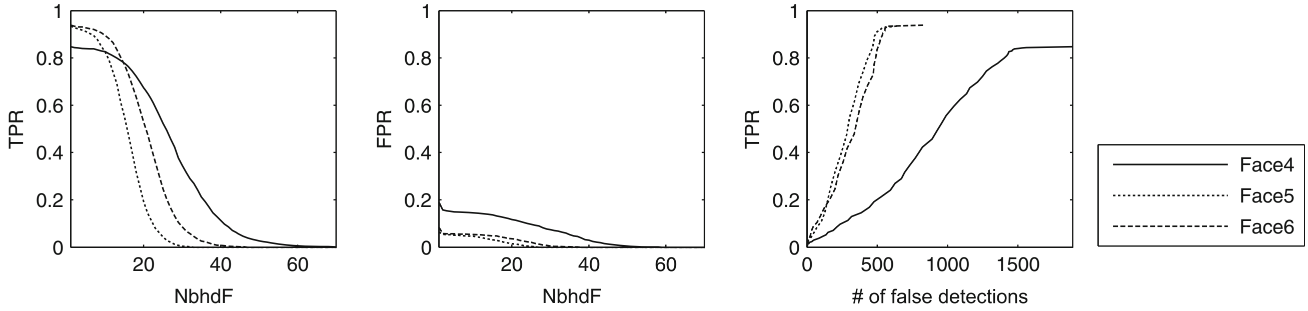

Fig. 14 The influence of the single stage *p* ratio on the face detector's performance for the frontal face images

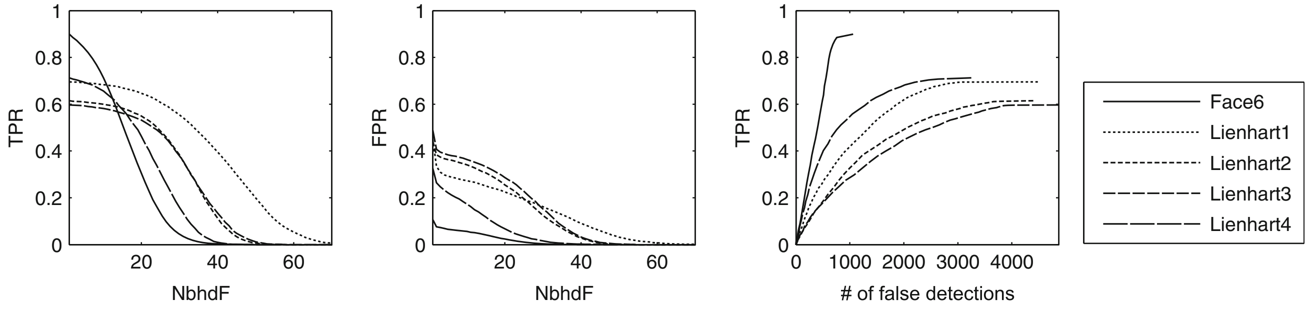

Fig. 15 The comparison of the Lienhart's detectors with the best trained face detector for the whole image base

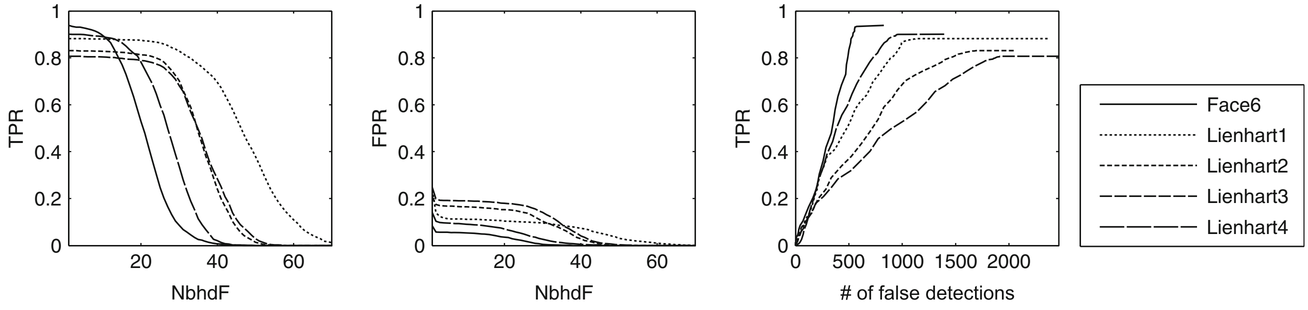

Fig. 16 The comparison of the Lienhart's detectors with the best trained face detector for the frontal face images

Table 1 Average time of face detection on a PC with Intel Celeron 2,800 MHz processor and 512 MB RAM

Detector	Average detection time (ms)
Face1	246.07
Face2	230.09
Face3	227.49
Face4	227.26
Face5	214.15
Face6	221.29
Lienhart1[10]	503.93
Lienhart2[10]	281.57
Lienhart3[10]	237.87
Lienhart4[10]	492.72

and Lienhart4 HCCs. All the detectors have been used in the direct, non-regionalized (non-reg) and the regionalized (reg) search. The error metric used here was the same as that of Campadelli [1]:

$$\text{error} = \frac{\max(\|C_l - C_{lGT}\|, \|C_r - C_{rGT}\|)}{\|C_{lGT} - C_{rGT}\|} \tag{10}$$

where C_l stands for the center of the left eye found, C_r stands for the center of the right eye found, C_{lGT} and C_{rGT} are the centers of the ground truth eyes (Fig. 8).

The detections with the relative error lower than 0.1 were treated as the TPs, when those with higher error were considered the FPs. The pictures without any positive eyes detection result were counted as FN.

All of the tested eyes HCCs gave similar TP ratios. However, our best detector delivered the FP ratio almost 10 times smaller than the Castrillón-Santana's HCC (Fig. 17). Our eyes detectors were also visibly superior w.r.t. the processing time (Table 2). The higher TP ratio obtained for the frontal face images (Fig. 18) can be explained with the strong influence of the face pose on the visibility of the eyes. In the case of the strong head turn eyes can be occluded by the nose.

The drop of the performance for automatically detected faces (Figs. 19, 20) is the result of the face detector FNs. If no face was detected the image processing was aborted and no eyes could be found.

The eyes detection results were almost identical despite the choice of the face detector (Figs. 21, 22). This shows that even rough estimates of the face region are sufficient for the proper eyes detection. Moreover, the eyes detection as the second stage of the proposed system demonstrates some filtering abilities. The FP ratio of the face detector did not propagate any further, as a consequence of discarding face candidates with no eye pairs found.

It is also better to leave the face detector unconstrained (by setting the *NbhdF* to 0) and to fine-tune the whole

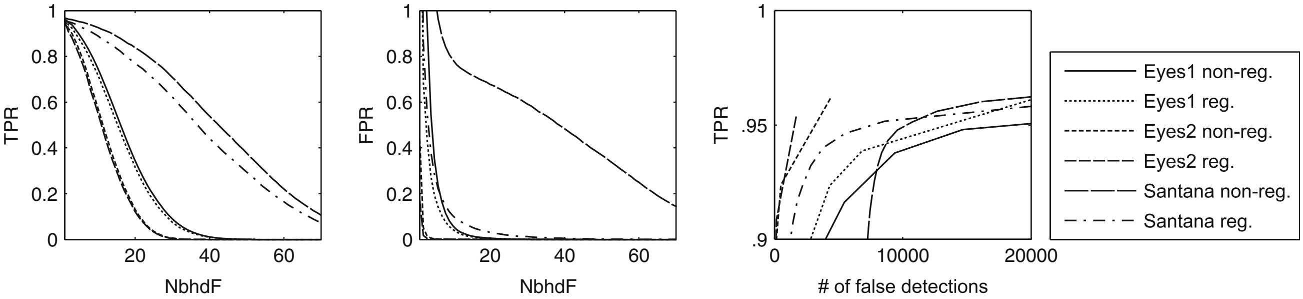

Fig. 17 The performance of eyes detectors applied to the manually marked face ROIs for the whole image base

Table 2 Average detection time for eyes on a PC with Intel Celeron 2,800 MHz processor and 512 MB RAM

Detector	Average detection time (ms)
Eyes1 non-reg.	377.35
Eyes1 reg.	105.83
Eyes2 non-reg.	337.59
Eyes2 reg.	100.06
Santana [29] non-reg.	2,171.08
Santana [29] reg.	625.07

system only by changing the minimum number of the merged eyes detections (*NbhdE*). Figures 23 and 24 show that the final FP ratio does not depend on the *NbhdF* parameter, while its increase leads to the quick deterioration of the TP ratio.

The regionalized search has proved to be a very useful concept. Its application resulted in the significant reduction of the FP ratio with only slight decrease of the TP. The processing time was also greatly shortened.

The exemplary results of the combined face and eyes detection are presented in Figs. 25, 26 presents the mean localization error of the regionalized E2 detector as a function of the *NbhdF* parameter.

9 Conclusions

Our tests clearly demonstrated that the HCC can be successfully used in the face and eyes detection system. Combining the two detectors in the hierarchical structure and augmenting them with the additional knowledge-based rules resulted in the fast and efficient system.

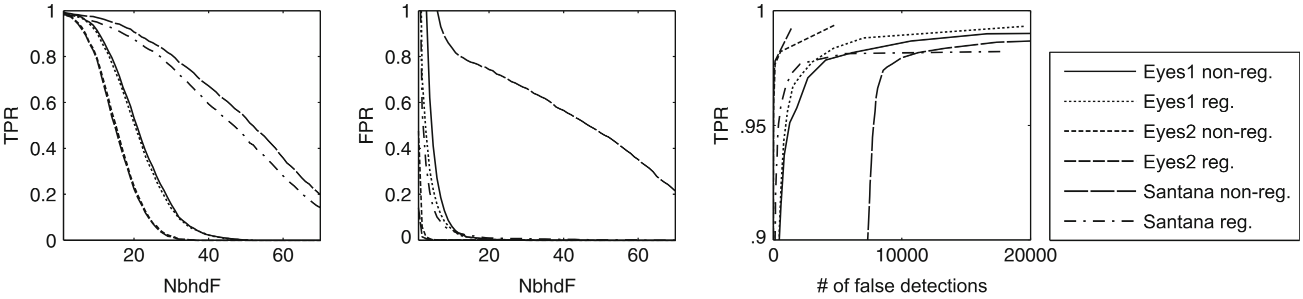

Fig. 18 The performance of eyes detectors applied to the manually marked face ROIs for the frontal face images

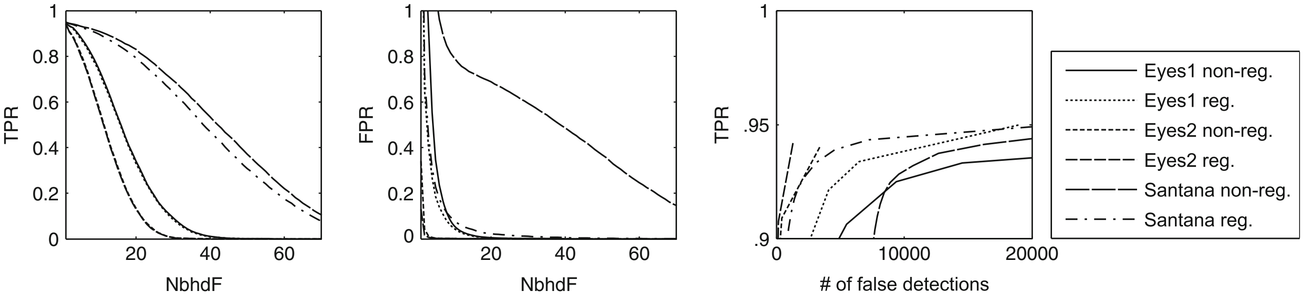

Fig. 19 The performance of eyes detectors applied to the face ROIs detected with the Face6 HCC for the whole image base

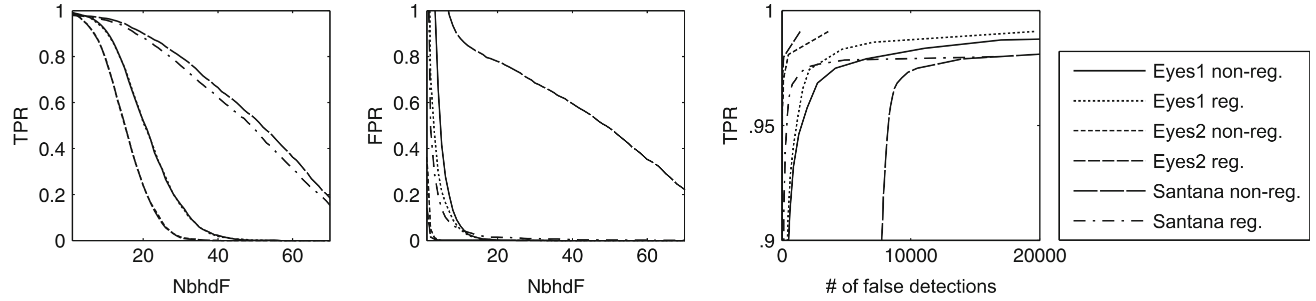

Fig. 20 The performance of eyes detectors applied to the face ROIs detected with the Face6 HCC for the frontal face images

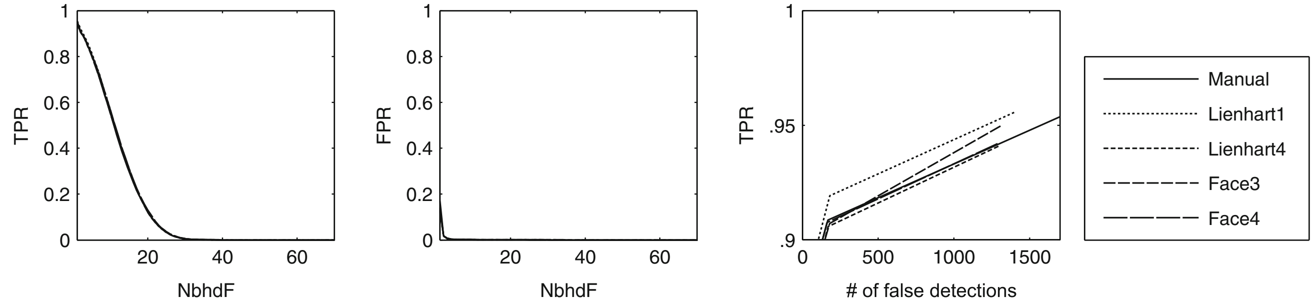

Fig. 21 The performance of the regionalized Eyes2 eyes detector applied to the face ROIs detected with various face HCC for the whole image base

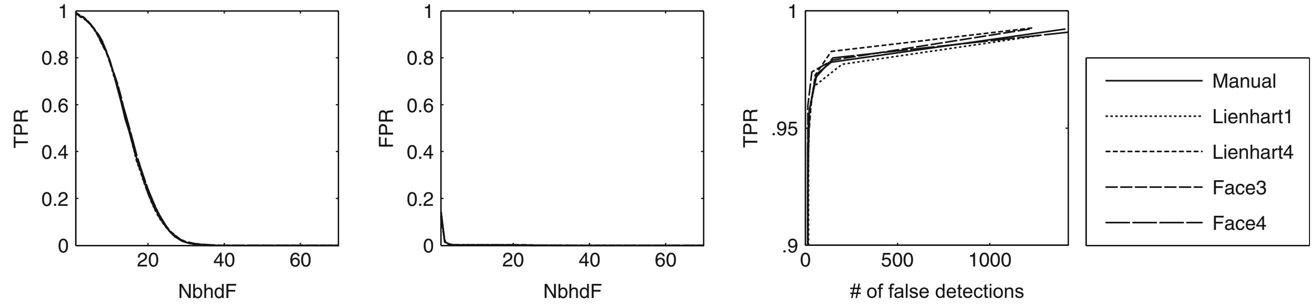

Fig. 22 The performance of the regionalized Eyes2 eyes detector applied to the face ROIs detected with various face HCC for the frontal face images

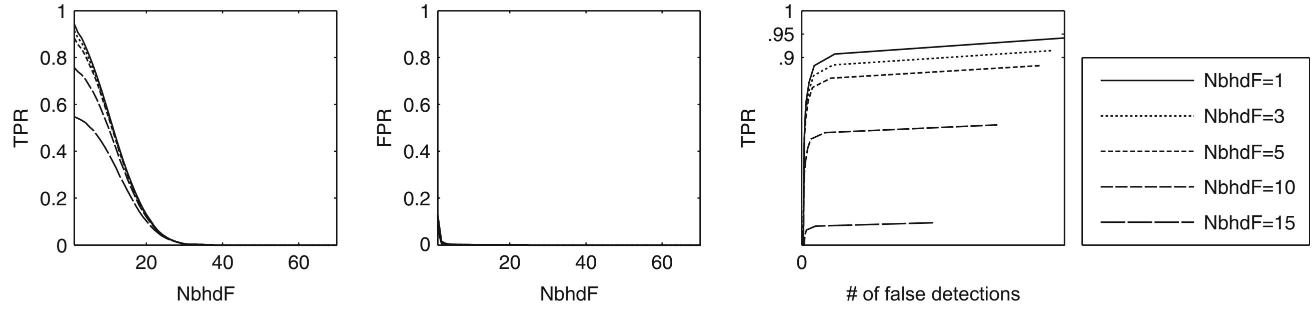

Fig. 23 The performance of the regionalized Eyes2 eyes detector applied to the face ROIs detected with the Face6 face HCC with varying *NbhdF* parameter for the whole image base

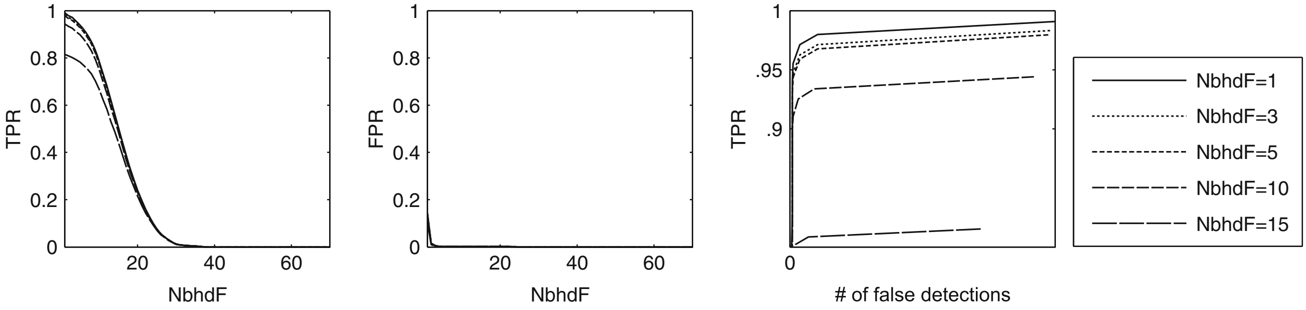

Fig. 24 The performance of the regionalized Eyes2 eyes detector applied to the face ROIs detected with the Face6 face HCC with varying *NbhdF* parameter for the frontal face images

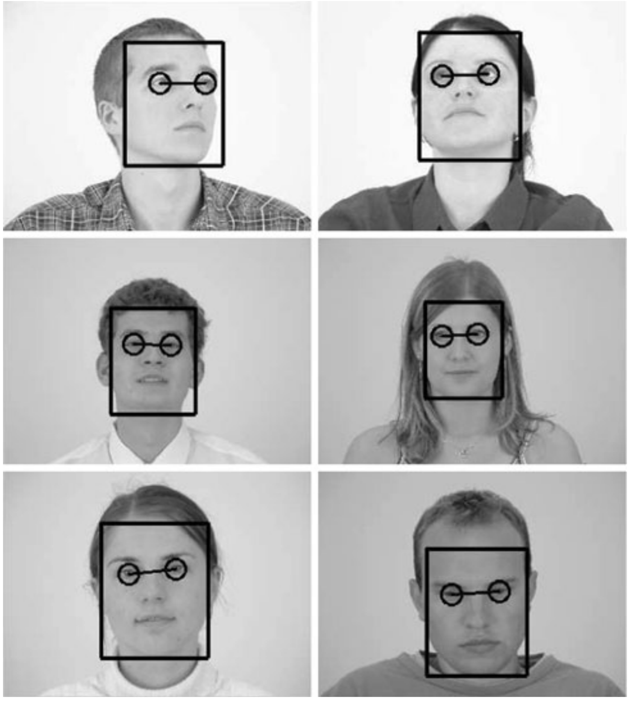

Fig. 25 The examples of face and eyes detection with Face6 face HCC and regionalized Eyes2 eyes HCC

The detector trained with the four-split CART as the weak classifier and the required p ratio of each stage set to 0.999 outperformed all Lienhart's detectors both w.r.t. the detection ratio and the computational efficiency. By using solely the face detector we were able to detect 90% of the faces, getting the FP ratio of 11% while considering the whole image base. For the frontal face images set the TP = 94% with FP = 8.4%.

Our results confirmed the hypothesis that using the regionalized search results in a significant reduction of both the FP ratio and the processing time.

Castrillón-Santana's and our detectors achieved comparable TP ratios; however, our solution turned out to give a several times lower FP ratio. It is worth to point out that the processing time with our detectors was also six times shorter.

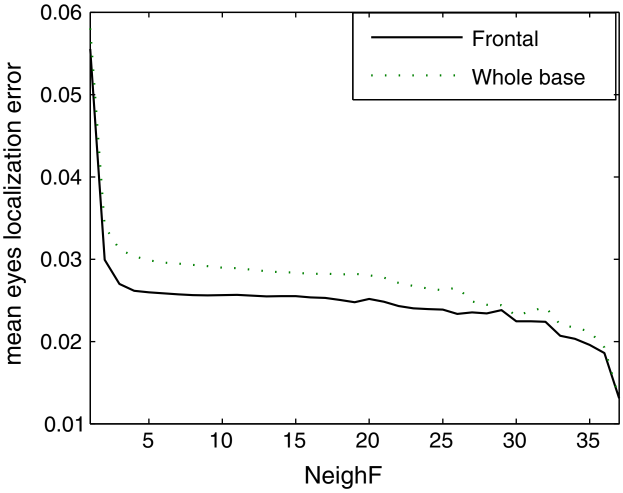

Fig. 26 The mean eyes localization error of the regionalized Eyes2 detector applied to the faces detected with the Face6 HCC

By using the combination of both our face and our regionalized eyes detector we were able to fully automatically detect the eyes in 94% of images still keeping the FP ratio of 13%. While analyzing only the frontal images the TP was equal to 99% and FP to 14%. The mean value of the eyes localization error was 0.058 for the whole base and 0.055 for the frontal images. By applying the minimum neighbors constraint solely to the eyes detector the TP ratio of 88% was achieved with less than 1% FP and the mean localization error of 0.031 (TP = 97%, FP = 0.5% and the mean error of 0.027 for the frontal face images only). The average processing time on a PC with the Intel Celeron 2.8 GHz processor and 512 MB RAM was 321 ms.

Our detection system has proved to be efficient both w.r.t. detection rates and computation costs. It turned out to be resistant to pose variations and to structural disturbances.

References

1. Campadelli P, Lanzarotti R, Lipori G (2006) Eye localization: a survey. In: Esposito A et al (eds) Fundamentals of verbal and nonverbal communication and the biometric issue. IOS Press BV, Amsterdam, pp 234–245
2. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE conference on computer vision and pattern recognition 2001, pp 511–518)
3. Kotropoulos C, Pitas I (1997) Rule-based face detection in frontal views. In: Proceedings of the international conference on acoustics, speech and signal processing 1997, pp 2537–2540
4. Hsu R-L, Abdel-Mottaleb M, Jain A (2002) Face detection in color images. IEEE Trans Pattern Anal 24:696–706
5. Heisele B, Serre T, Pontil M, Poggio T (2001) Component-based face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition 2001, pp 657–662
6. Bileschi S, Heisele B (2003) Advances in component based face detection. In: Proceedings of the IEEE workshop on analysis and modelling of faces and gestures 2003, pp 149–156
7. Su M-C, Chou C-H (2001) Associative-memory-based human face detection. IEICE Trans Inform Syst E84-D:1067–1074
8. Rowley H, Kanade T, Baluja S (1996) Neural network-based face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition 1996, pp 203–207
9. Huang L-L, Shimizu A, Hagihara Y, Kobatake H (2003) Gradient feature extraction for classification-based face detection. Pattern Recognit 36:2501–2511
10. Lienhart R, Kuranov A, Pisarevsky V (2002) Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Intel Labs, Microprocessor Research Lab Technical report
11. Meynet J, Popovici V, Thiran J-P (2005) Face detection with mixtures of boosted discriminant features. Ecole Polytechnique Fédérale de Lausanne Signal Processing Institute Technical report
12. Wang Q, Yang J (2006) Eye detection in facial images with unconstrained background. J Pattern Recognit Res 1:55–62
13. Kumar T, Raja K, Ramakrishnan A (2002) Eye detection using color cues and projection functions. In: Proceedings of the IEEE international conference on image processing 2002, pp 337–340
14. Peng K, Chen L, Ruan S, Kukharev G (2005) A robust algorithm for eye detection on gray intensity face without spectacles. J Comput Sci Technol 5:127–132
15. Wu J, Zhou Z-H (2003) Efficient face candidates selector for face detection. Pattern Recognit 36:1175–1186
16. Campadelli P, Lanzarotti R, Lipori G (2006) Precise eye localization through a general-to-specific model definition. In: Proceedings of the 7th British machine vision conference, pp 187–196
17. Motwani M, Motwani R, Harris F (2004) Eye detection using wavelets and ANN. In: Proceedings of the global signal processing expo and conference 2004
18. Tivive F, Bouzerdoum A (2005) A fast neural-based eye detection system. University of Wollongong, Faculty of Informatics Technical report
19. Bianchini M, Sarti L (2006) An eye detection system based on neural autoassociators. In Proceedings of the IAPR international workshop on artificial neural networks in pattern recognition 2006, pp 244–252
20. Wilson P, Fernandez J (2006) Facial features detection using Haar classifiers. J Comput Sci Coll 21:127–133
21. Feng X, Wang Y, Li B (2006) A fast eye location method using ordinal features. In: Proceedings of the ACM SIGCHI international conference on advances in computer entertainment technology 2006, p 95
22. Wang P, Green M, Ji Q, Wayman J (2005) Automatic eye detection and its validation. In: Proceedings of the IEEE conference on computer vision and pattern recognition 2005, p 164
23. Everingham M, Zisserman A (2006) Regression and classification approaches to eye localization in face images. In: Proceedings of the international conference on automatic face and gesture recognition, pp 441–448
24. Arandjelovic O, Zisserman A (2005) Automatic face recognition for film character retrieval in feature-length films. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 860–867
25. Freund Y, Schapire R (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
26. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth International Group, Belmont
27. Friedman, Hastie T, Tibshirani R (1998) Additive logistic regression: a statistical view of boosting. Stanford University, Department of Statistics, Technical Report
28. Open Computer Vision Library. http://sourceforge.net/projects/opencvlibrary/
29. Castrillón-Santana M, Lorenzo-Navarro J, Déniz-Suárez O, Falcón-Martel A (2005) Multiple face detection at different resolutions for perceptual user interfaces. In: Proceedings of the 2nd Iberian conference on pattern recognition and image analysis, pp 445–452

Author Biographies

Andrzej Kasinski, Ph.D., D.Sc., graduated from the Poznan University of Technology in Electrical Engineering (1973) and Adam Mickiewicz University in mathematics (1974). He is a professor and Head of the Institute of Control and Information Engineering, Poznan University of Technology. He is the author of over 80 technical papers in the field of control engineering, simulation, digital system design, robotics and computer vision. His current research interest is focused on image analysis, pattern recognition, multi-agent systems methodology and biocybernetics. He held a number of visiting positions at USA (Cornell), The Netherlands (Delft University of Technology) and Spain (University de Murcia).

Adam Schmidt was born in Poznan, Poland in 1984. He received the M.Sc. Eng degree in Control Engineering and Robotics from the Poznan University of Technology in 2007. Since then he has been pursuing his Ph.D. in Robotics and working as a research assistant at the Institute of Control and Information Engineering, Poznan University of Technology. His research interests include computer vision, machine learning and pattern analysis.