

Document downloaded from:

<http://hdl.handle.net/10251/58789>

This paper must be cited as:

Monzó Ferrer, D.; Albiol Colomer, A.; Sastre, J.; Albiol Colomer, AJ. (2011). Precise eye localization using HOG descriptors. *Machine Vision and Applications*. 22(3):471-480. doi:10.1007/s00138-010-0273-0.



The final publication is available at

<http://dx.doi.org/10.1007/s00138-010-0273-0>

Copyright Springer Verlag (Germany)

Additional Information

Precise eye localization using HOG descriptors*

David Monzo Alberto Albiol Jorge Sastre
Antonio Albiol

December 9, 2015

Abstract

In this paper, we present a novel algorithm for precise eye detection. First, a couple of AdaBoost classifiers trained with Haar-like features are used to preselect possible eye locations. Then, a SVM machine that uses HOG descriptors is used to obtain the best pair of eyes among all possible combinations of preselected eyes. Finally, we compare the eye detection results with three state of the art works and a commercial software. The results show that our algorithm achieves the highest accuracy on the FERET and FRGCv1 databases, which is the most complete comparative presented so far.

1 Introduction

Eye detection is the natural link between face detection and recognition algorithms. Surprisingly, most face recognition approaches assume that faces are precisely located and most holistic face detection algorithms ignore the issue of precision. Precision is important for face recognition as it has been proved in some recent works [26, 17, 34], which show that for most algorithms the recognition rate decreases rapidly as the eyes shift from the ideal locations. For this reason, many authors consider today that the problem of precise face localization is of great importance.

In the last years, two types of images have been used for the location of eyes, namely IR and RGB images. In the case of IR images, the key idea is to detect eyes looking for the reflection of the pupils under near IR illumination [2, 37]. However, these methods require a specific technology and equipment that is not always available. Methods that use RGB images can be broadly grouped into three categories: template matching, feature and appearance based. In template algorithms [12, 5], several standard patterns of an eye are used for searching using cross-correlation. These algorithms are computationally expensive as the number of operations they require is proportional to the size and number of templates, usually high. Feature based algorithms [9, 5, 27, 10] use previous

*This work has been partially supported by the grant TEC2009-09146 of the Spanish Government

knowledge to define a set of heuristic rules. For instance, in [9] the author exploits the facts that the eyes have strong horizontal edges and that the pupil is darker than the surrounding facial regions, while in [27] they use the information of the edges and the intensity to locate the iris. However, the a priori knowledge used in feature based algorithms is not always easy to determine. Also, most approaches cannot cope with a large number of environments for eye detection as it is difficult to find features capable to adapt to different conditions. Finally, appearance based algorithms use supervised classifiers which are fed with low level features. Some examples of low level features include wavelets [30, 6], Gabor features [29, 28] and Haar-like features [20, 19], among others.

Once features are extracted supervised classification algorithms are used to differentiate between eyes and non-eyes. For instance, in [16] a Support Vector Machine (SVM) classifier [32] which integrates the characteristics of single eye and eye-pair images is used to develop a hybrid classifier. The features that describe the eyes are selected via a filter and a wrapper method for a simplified maximum likelihood, and verified with template matching scores to reject false detections. Support Machines are also proposed in [6]; in this work the SVM is trained on properly selected Haar wavelet coefficients. The AdaBoost classifier [33] has also been extensively used for the classification of eyes [34, 8, 19]. In [34] AdaBoost is used with features extracted after a Recursive Nonparametric Discriminant Analysis (RNDA); in [8] the authors model images with a generative approach that uses a boosting stage to learn these models. In [19] AdaBoost is used in a preliminary coarse eye detection stage. This idea is also used in our work as it will be shown in Section 2.

Other solutions have also been proposed for precise face and eye localization. For instance, in [11] a genetic algorithm combined with decision trees is used to classify salient locations as eye regions. In [3] eye localization is achieved using hierarchical neural networks with multi-resolution and local recurrent connectivity.

Although some good performances are reported in the previous approaches, still further efforts are needed to increase the precision/recall of eye detection systems to improve the recognition rates of fully automatic face recognition systems. In this paper, we present a novel and accurate eye localization approach. The novelty of our approach lies in the use of a SVM classifier trained with Histograms of Oriented Gradients (HOG) features [7] obtained after normalization of the face. HOG features, which have been previously used in applications such as pedestrian detection [7] and face recognition [1], are closely related to the Scale Invariant Feature Transform (SIFT) [18]. SIFT features have emerged as a cutting edge technology for extracting distinctive features from images, to be used in algorithms for tasks like matching different views of an object or scene and face authentication [4]. One of the essential parts of SIFT is that local keypoints are represented using histograms of image gradients which are normalized to be invariant to changes in scale and rotation. While these normalizations have proved to be very useful for images that are arbitrarily scaled or rotated, the fact is that they remove information which might be useful for recognition when images are not scaled or rotated. HOG features, similarly to

SIFT, also use gradient to describe local keypoints. The main difference to SIFT features is that scale and rotation normalization stages are skipped, which is an advantage when they are extracted from normalized faces. Another important difference is that SIFT features are extracted at keypoints which are local extrema of the space-scale representation of the image, while our HOG features are extracted at selected keypoints as will be described below.

Figure 1 shows the block diagram of our approach. The input to the system is a bounding box that contains a frontal face. The reader can see [36] for a complete survey on face detection approaches although we recommend the algorithm proposed by Viola and Jones in [33].

Similar to most object localization approaches, we perform eye detection by scanning a window at different scales over the image. This exhaustive scanning allows to deal over variations of position and scale. In our approach, we first scan the upper part of the face region with an AdaBoost classifier [33] that uses Haar-like features. The goal of this step is to preselect eye candidates at different scales with high recall and reject most negative subwindows efficiently. Note that this step is computed over a small region of the face which allows to keep the computational burden low as it will be shown in section 6. Next for each possible pair of eyes candidates, a normalized face is extracted by placing the eyes at fixed coordinates and thus compensating for rotation and scale changes. Finally, SVM is used to select the best candidate using HOG descriptors obtained from the eyes of the normalized faces.

The rest of the paper is organized as follows. Sections 2 through and 4 explain in detail each block of the diagram presented in the Figure 1. The methodology and the datasets used to test the system are described in Section 5 and the results exposed in Section 6. Finally a summary and conclusions are drawn in Section 7.

2 Preselection of eye candidates

As mentioned in Section 1, the objective of this stage is to select potential eye candidates and discard most of the easy non-eye subwindows. This is done using two AdaBoost classifiers trained with samples from left and right eyes respectively. Their use in this work was motivated by their well known fastness and effectiveness. As in [33] these classifiers use Haar-like features calculated from the integral image of the upper part of the face, as shown in Figure 2. This reduced search area allows to obtain eye candidates efficiently, which is particularly important for this first stage. Despite its efficiency this step is still the one with highest computational burden because it has to be exhaustively applied on every possible position/scale. In our preliminary work, we tested the detection using a common classifier for left and right eyes, however the results showed that a lower recall was achieved using this option and finally, we choose to use different classifiers for each eye. Additional details about the training methodology are provided in Section 5.

Since the AdaBoost classifiers are designed to be quite robust to small

changes in translation and scale, many detections usually occur around each eye (and quite often around false positives). For this reason it is convenient to cluster eye detections. Towards this end we use a complete linkage hierarchical clustering [14]. This bottom-up clustering approach sets the distance between clusters as the distance between the two farthest elements in the two clusters. This clustering method typically leads to a higher number of clusters, as they are smaller, tighter, and more compact. This is more appropriate in our case because false positives are usually located at the eyes corners which are relatively close to the eye centers. Figure 3 illustrates the difference between using the single linkage method (distance between clusters is measured as the minimum distance between elements) and the proposed complete linkage method. As it can be seen, the single linkage method tends to maximize connectivity in a closeness sense and it usually merges true and false positives reducing accuracy.

After the clustering, a simple classification into right and left eyes is carried out on the candidates considering only their location on the face-detected region as it is illustrated in Figure 2.

3 Face candidates extraction and normalization

Preselection of eyes using AdaBoost usually yields multiple right and left eye-candidates (false positives). In order to select two final eye candidates per face, we examine all possible pair combinations of left and right detected-eyes as it is shown in Figure 4. In that example, six combinations are extracted from two right and three left eyes candidates. Using geometrical constraints some pairs are easily discarded, for instance if the eyes in the pair are too close or too distant. We also apply restrictions on the maximum rotation angle. In our case, if the pair is rotated more than $\pm 20^\circ$ about the horizontal axis it is also discarded. In the case that all eye pairs are removed using the geometrical constraints, we consider the detected-face region as a false positive.

For each valid eye pair, a 125×145 normalized face is extracted. In this normalized face the eyes are located at fixed locations (coordinates (25, 35) and (100, 35) respectively). As mentioned in Section 1, HOG features can be affected by changes in scale and rotation, and thus, this normalization is needed as preprocessing for the next stage.

4 SVM selection of the best face candidate using HOG descriptors

Although boosted classifiers trained with Haar-like features have the advantage of being computationally very efficient, their false positive rate is usually too high. In order to reduce these false positives, our eye detection algorithm selects the best face candidate obtained from all possible combinations of pre-selected eyes with a SVM classifier [32] as a final validation stage using HOG descriptors [7]. Although these descriptors are computationally more expensive

than the Haar-like features, they are more powerful and robust to illumination changes, see [22] for a comparison survey of local descriptors. Notice that these features are computed only on a reduced number of eye candidates and therefore the amount of computational burden of this stage is very small.

As mentioned in Section 1, HOG descriptors are local statistics of the orientations of the image around a keypoint. HOG descriptors are not scale/rotation invariant and for this reason, the normalization described in Section 3 is crucial prior to the extraction of these features.

Each HOG descriptor is a bundle of histograms composed of pixel orientations given by their gradients. The number of possible orientations (histogram bins) is referred to as N_o . Each histogram in the bundle describes a specific area around the keypoint. These areas correspond to the cells of a $N_p \times N_p$ squared grid centered on the keypoint (see Figure 5). In this work we set $N_p = 4$ cells for each spatial direction and $N_o = 8$ bins for each histogram in the bundle resulting in a total of $N_p^2 N_o = 128$ elements in a HOG descriptor. The choice of these settings is made based on the results of the work of Lowe [18].

We also set the size of each spatial cell to a square of 6×6 pixels. This size is chosen accordingly to the distance between eyes of the normalized faces, which in our work is 75 pixels.

In the HOG descriptor, the contribution of each pixel gradient to the histogram is weighted by the gradient modulus and a Gaussian window. The Gaussian window is centered at the keypoint coordinates and its standard deviation equals to half the extension of the spatial range, which is 12 pixels. Also the pixel contribution is distributed into adjacent spatial cells and orientations bins using trilinear interpolation. This is important to avoid all boundary effects in which the descriptor abruptly changes as a sample shifts smoothly from being within one cell to another or from one orientation to another. Gaussian windowing and trilinear interpolation also increase the robustness of the descriptor against small displacements of the keypoint location.

Finally, HOG descriptors are normalized to increase invariance to illumination changes. As in the Lowe’s algorithm, first the 128-dimensional vector is normalized to unit length. This normalization cancels changes in image contrast. Notice that we do not care about changes in brightness, a constant added to pixel values, because they are suppressed by image gradients. Finally, the descriptor is saturated so that no values over 0.2 are allowed and again re-normalized to unit length. This final step is done to reduce non-linear illumination changes.

One key advantage of using SVM classification is that it is possible to obtain a *soft* classification output (real-valued). This output can be used as a confidence value and allows us to select the best face candidate (the one with the highest sum of eye confidences). We reject all face candidates in which any of the eyes is classified as non-eye by the SVM classifier. In some cases, it is possible that all face candidates are rejected as it will be shown in Section 6.

As it is well-known SVM classifiers use a kernel that maps features onto a high-dimensional space prior to classification. In our work, the kernel chosen for the separating problem was a RBF kernel with $\sigma = 3$. This selection is further justified in Section 5.2.2.

5 Experimental setup

In this section we provide details about the data used to train and test our system and about the methodology followed to train the classifiers.

5.1 Eye databases

In this work, we have use two different sets of public databases for training and testing the eye classifiers respectively. These databases are shortly described below.

5.1.1 Training databases

In order to train the AdaBoost and SVM classifiers we have used data from the BioID database [13] and a subset of the CVL database [23].

The BioID database comprises 1521 images from 23 subjects. The images are taken in an indoor environment and in different days, light conditions and with slightly different camera distances and angles. The CVL database contains seven images per individual from a set of 114 individuals. The images are taken under uniform illumination and different orientations around the vertical axis. In this work, we only use the frontal faces of the CVL (3 images/subject). Figure 6.c shows the selected images of the faces from two subjects of the CVL database.

5.1.2 Test databases

In order to test and compare our algorithm with other researchers we have used frontal face images from the FERET database [25] and the FRGCv1 [24] database. These databases have become a *de facto* standard for testing eye localization algorithms (as well as for face detection and recognition tasks).

The FERET database contains 3365 full frontal facial images of nearly 1000 subjects and the FRGCv1 consists of 5658 images of 275 subjects. The FERET database contains a very large variety of individuals in different sessions. In the FRGC database, the number of individuals is not so extensive but there is a large amount of images from each individual. Both databases contain images taken under controlled and uncontrolled conditions. The Figure 6.a and Figure 6.b show some examples of images from both databases.

5.2 Training methodology

5.2.1 Training of the AdaBoost classifier

The AdaBoost classifier was trained using positive and negative samples of 15×15 pixels from the BioID and CVL databases. We manually marked eye locations for the CVL database and used the ground truth data provided with the BioID database. To obtain the positive eye samples, we normalized all the images in rotation and scale, so that the inter-ocular distance (*iod*) of the

normalized faces was set to $iod = 75$ pixels. The initial positive set contained nearly 3400 eye images, although to increase the robustness of the classifier, it was augmented to nearly 60.000 samples for each kind of eye (left and right) by using in-plane rotations ($\pm 2^\circ$ and $\pm 4^\circ$) and adding Gaussian noise. Figure 7.a shows a pair of samples of two different eyes, each sample with different angle and noise-level.

The negative samples were obtained iteratively using bootstrapping from BioID images where the eyes were blurred as shown in Figure 7.b. One problem that we have found in our training set is the relatively low number of people wearing glasses. Glasses increase the variation of the eye-areas even due to shape changes, light reflections and others. However, it was difficult to us gather a representative set of glass-wearing samples. After training, we obtained two AdaBoost classifiers that use 175 and 181 Haar-like features for the left and right eyes respectively distributed along 12 stages for each classifier.

5.2.2 Training of the SVM classifier

To simulate real working conditions, we trained the SVM classifier with pre-selected eyes obtained from the AdaBoost classifiers following the diagram of Figure 1. The images used to extract the SVM training examples are a subset of 223 images (118 different individuals) from the BioID and the CVL face databases. To label each eye detected by AdaBoost as either positive or negative training example, we used the normalized eye localization error [15] defined as:

$$N_{error} = \frac{|E_{det} - E_{gt}|}{iod} \times 100 \quad (1)$$

That is, the distance between the detected eye, E_{det} , and the ground truth, E_{gt} , normalized by the inter-ocular distance iod . Using this measure, we labeled as positive examples all detected eyes with $N_{error} < 5\% \text{ } iod$ and the rest as negative for the training of the SVM machine.

Using the preselected eyes, we built different normalized faces from all possible combinations of detected eye-pairs as described in Section 3. Three cases appeared, as illustrated in Figure 8:

- Normalized faces in which both eyes were positive samples following the previous criteria.
- Normalized faces with both eyes negative samples.
- Normalized faces with one positive and one negative eye sample.

In the two first cases, the eyes are used as positive and negative samples respectively. In the third case, only the negative eye sample is used to train the SVM classifier. With these considerations, we gathered a total of 11078 positive samples and 22160 negative samples.

We trained several SVM machines using two different types of kernels: RBF-Gaussian and polynomial. The parameters for each kernel, σ and polynomial

Boosting Stages	Right Eye Classifier	Left Eye Classifier	Common Classifier
Detection Rate	99.1%	99.7%	98.35%
False Alarms per eye	1.98	2.35	1.75

Table 1: Performance results of the first stages of the system on FERET considering outputs within 10% *iod* as hits.

Boosting Stages	Right Eye Classifier	Left Eye Classifier	Common Classifier
Detection Rate	98.8%	99.2%	96.68%
False Alarms per eye	1.44	1.47	1.2

Table 2: Performance results of the first stages of the system on FRGCv1 considering outputs within 10% *iod* as hits.

order were set using a cross-validation methodology in which the training examples from one person were left out in each round. Figure 9 shows the precision-recall curves for different values of σ and polynomial orders. From those results, we choose RBF kernel with $\sigma = 3$ as a good trade-off.

6 Results

6.1 Performance of AdaBoost classifiers

In this section we evaluate the performance of the AdaBoost classifiers used to extract the left and right eye candidates. Using Equation 1, we consider that a detection is a hit when its $N_{error} < 10\%$ *iod*. See Figure 10 to get an idea of the magnitude of this tolerance.

Table 1 and Table 2 show the detection rate (percentage of eyes with at least one hit) and the average number of false alarms per eye (the average number of detections with $N_{error} > 10\%$ *iod* per eye). We also show the performance of a single common classifier which detects both left and right eyes. Notice that this common classifier achieves slightly lower detection rates. Since at this point higher detection rates are preferred we selected the solution of using different classifiers for each eye. Anyway, the results from Table 1 and 2 show that it is still necessary to add extra processing to reduce the high number of false alarms.

As described in Section 3 we build eye pair candidates using all eye candidates. In FERET, the average number of valid eye-pair candidates per person was of 13.41 and in the FRGCv1 of 10.31.

6.2 Eye Location Results

In this section we compare the results achieved with our final solution to other state of the art recent research works [16], [6], [34]. We also have compared the results with a commercial software developed by Neurotechnologija called VeriLook [21]. We would like to emphasize, that the comparisons have been

Maximum N_{error}	5% <i>iod</i>	10% <i>iod</i>	25% <i>iod</i>
Our approach	78.0%, _[76.73, 79.22]	96.2% _{, [95.58, 96.73]}	99.6% _{, [99.36, 99.75]}
L. Jin [16]	55.1% _{, [53.02, 57.17]}	93.0% _{, [91.86, 93.99]}	99.8% _{, [99.51, 99.92]}
P. Campadelli [6]	67.7% _{, [61.21, 73.57]}	89.5% _{, [84.70, 92.92]}	96.4% _{, [93.00, 98.18]}
Verilook [21]	74.6% _{, [73.29, 75.86]}	96.8% _{, [96.24, 97.28]}	99.9% _{, [99.75, 99.96]}

Table 3: Eye detection percentages and confidence intervals on FERET achieved with different criteria of maximum error distance of the detected eyes.

Maximum N_{error}	5% <i>iod</i>	10% <i>iod</i>	25% <i>iod</i>
Our approach	92.3% _{, [91.63, 92.92]}	98.5% _{, [98.17, 98.77]}	99.6% _{, [99.41, 99.73]}
P. Wang [34]	91.2% _{, [90.47, 91.88]}	99.0% _{, [98.72, 99.22]}	99.7% _{, [99.53, 99.81]}
P. Campadelli [6]	81.2% _{, [73.53, 84.39]}	92.8% _{, [90.19, 94.76]}	97.1% _{, [95.23, 98.25]}
Verilook [21]	82.6% _{, [81.66, 83.50]}	97.8% _{, [97.42, 98.13]}	99.9% _{, [99.79, 99.95]}

Table 4: Eye detection percentages and confidence intervals on FRGCv1 achieved with different criteria of maximum error distance of the detected eyes

carried out using exactly the same data and methodology on the FERET and FRGCv1 databases for all the algorithms.

After applying our eye detection method to all the images in the FERET and FRGCv1 databases, we are able to detect at least one valid eye pair in the 97.12% and 99.28% of the images respectively. Figures 11.a and 11.b compare the results using accumulated eye localization errors for those images where at least one valid eye pair is detected. In these figures, the horizontal axis is the normalized localization error, defined in Eq 1, and the vertical axis is the accumulated distribution, which means the percentage of eyes with smaller normalized error than the corresponding horizontal value. These results are also presented in Tables 3 and 4 for better visualization along with the intervals for a confidence level of 99% [35]. The confidence limits depend on the sample size and the estimated probability. As we can see, our solution gets the best results for both databases respect to all the other approaches for a maximum N_{error} of 5%. The Verilook software achieves a better performance in the FERET database, this could be explained by the fact that the software was probably trained using FERET and XM2VTSDB databases as it is mentioned in their web page. In the case of the FRGCv1 database, we get slightly better results than Wang’s algorithm [34] for a $N_{error} < 5\%$, for higher values of N_{error} the difference is statistically negligible attending to the confidence intervals. It should be mentioned that results are not exactly comparable because not all the algorithms use all the face images in the databases. For instance, in the case of Wang’s algorithm, the eye localization accuracy is measured over the 94.5% of the images of FRGCv1 (images in which he detects the face and the corresponding eyes). In our case the eye localization is evaluated on the 99.28% of the FRGC images, in the rest of the images we did not find any valid eye pair. However, notice that this number of images is the highest used by any of

	Execution times
Face Detection (Boosting Stage)	19.7ms
Eyes Detection (Boosting Stages)	115.9ms
Eye-pair Selection (Normalization + HOG + SVM classification)	46.0ms
Overall Eye Localization Time	181.5ms

Table 5: Execution times obtained during the different stages of our implementation.

the approaches under comparison.

Finally, we would like to mention that several authors [15] propose a threshold of $N_{error} < 25\%$ to consider an eye detection as a hit. However, we think that a value of the $N_{error} = 25\%$ is too loose to get accurate results for most applications. Therefore, we consider a hit only if $N_{error} < 10\%$, see Figure 10.

Regarding the computational complexity, Table 5 summarizes the execution time used by each part of our system when applied to the FERET images (size 256×384) using a standard PC (1.85GHz dual core Pentium). In this work we did not pay much attention to obtain a computationally optimal implementation. Our overall processing time could be greatly reduced for instance with a multithreaded implementation of our algorithms. Comparing to the authors that report execution times, in [6] the average running time per image was of 4sec on a 3.2GHz PC and in [34] of more than 100ms on a 2.7GHz PC.

6.3 Influence on Face Recognition

As mentioned in Section 1, for most Face Recognition systems a precise localization of the eyes is required to achieve high recognition rates. In this section, we show the influence of eye detection on face recognition. The images used for this experiment are taken from the *ba*, *bj* and *bk* sets of the FERET database [25]. Specifically, these sets contain 600 images corresponding to 200 subjects, such that each subject has three images. Two of these images are randomly chosen to be part of the gallery and the remaining are used for probe. We use Principal Component Analysis (PCA) [31] as a baseline algorithm to perform face recognition. This is one of the most widespread methods for face recognition. Figure 12 shows the recognition results for three different cases: hand-marked locations of the eyes (groundtruth data), randomly shifted locations of the hand-marked eyes using uniform noise with a maximum amplitude of 5% *iod* and the locations of the eyes provided by our solution.

7 Summary and conclusions

In this paper we presented a new method to precisely localize eyes. The method consists of two major steps. First, we detect efficiently a set of possible eye locations using a couple of Haar-like AdaBoost classifiers. Then, for each possible eye pair we extract and normalize a face. Using the normalized faces we select

the best eye pair using a SVM classifier trained with HOG descriptors. The test experiments have been carried out on the FERET and FRGCv1 databases. Using the same data and methodology we have compared the results with other recent state of the art approaches. The results show that our algorithm achieves the highest precision on both databases. Finally, we would like to emphasize that although our implementation pipelines several stages, the overall computational complexity is similar to that of other state of the art algorithms.

References

- [1] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol. Face recognition using hog-ebgm. *Pattern Recognition Letters*, 29(10):1537–1543, 2008.
- [2] A. Amir, L. Zimet, A. Sangiovanni-Vincentelli, and S. Kao. An embedded system for an eye-detection sensor. *Computer Vision and Image Understanding*, 98(1):104–123, 2005.
- [3] S. Behnke. Face localization and tracking in the neural abstraction pyramid. *Neural Computer Applications*, 14(2):97–103, 2005.
- [4] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli. On the use of SIFT features for face authentication. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop*, page 35, New York, June 2006.
- [5] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
- [6] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization through a general-to-specific model definition. In *Proceedings of the British Machine Vision Conference*, pages 187–196, Edinburgh, Scotland, 2006.
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 9th European Conference on Computer Vision*, volume 2, pages 886–893, San Diego, CA, June 2005.
- [8] I. Fasel, B. Fortenberry, and J. Movellan. A generative framework for real time object detection and classification. *Computer Vision and Image Understanding*, 98(1):182–210, 2005.
- [9] Y. Guan. Robust eye detection from facial image based on multi-cue facial information. *IEEE International Conference on Control and Automation*, pages 1775–1778, June 2007.
- [10] C. Han, H. Liao, K. Yu, and L. Chen. Fast face detection via morphology-based pre-processing. In *Proceedings of the 9th International Conference on Image Analysis and Processing*, volume 2, pages 469–476, London, UK, 1997. Springer-Verlag.

- [11] J. Huang and H. Wechsler. Visual routines for eye location using learning and evolution. *IEEE Transactions on Evolutionary Computation*, 4(1):73–82, April 2000.
- [12] W. Huang and R. Mariani. Face detection and precise eyes location. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 722–727, Washington, DC, USA, 2000.
- [13] Humanscan. BioID database. <http://www.bioid.com>.
- [14] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing systems*, 31(3):264–323, September 1999.
- [15] O. Jesorsky, K. J. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 90–95, London, UK, 2001. Springer-Verlag.
- [16] L. Jin, X. Yuan, S. Satoh, J. Li, and L. Xia. A hybrid classifier for precise and robust eye detection. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 731–735, Hong Kong, 2006.
- [17] C. Kim and C. Choi. Image covariance-based subspace method for face recognition. *Pattern Recognition*, 40(5):1592–1604, 2007.
- [18] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of Computer Vision*, 60(2):91–110, November 2004.
- [19] H. Lu, W. Zhang, and D. Yang. Eye detection based on rectangle features and pixel-pattern-based texture features. *International Symposium on Intelligent Signal Processing and Communication Systems*, pages 746–749, December 2007.
- [20] Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 339–344, Seoul, Korea, May 2004.
- [21] Neurotehnologija, Biometrical and Artificial Intelligence Technologies. Verilook SDK. <http://www.neurotehnologija.com>.
- [22] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.
- [23] P. Peer. CVL Face database, University of Ljubjana. <http://www.fri.uni-lj.si/en>.

- [24] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, C. Jin, K. Hoffman, J. Marques, M. Jaesik, and W. Worek. Overview of the face recognition grand challenge. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 947–954, San Diego, CA, June 2005.
- [25] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
- [26] T. Riopka and T. Boult. The eyes have it. In *Proceedings of ACM SIGMM Multimedia Biometrics Methods and Applications Workshop*, pages 9–16, Berkley, California, 2003.
- [27] M. Rizon and T. Kawaguchi. Automatic eye detection using intensity and edge information. In *Proceedings of TENCON*, volume 2, pages 415–420, Kuala Lumpur, Malaysia, 2000.
- [28] S. A. Sirohey and A. Rosenfeld. Eye detection in a face image using linear and nonlinear filters. *Pattern Recognition*, 34(7):1367–1391, 2001.
- [29] F. Smeraldi, O. Carmona, and J. Bign. Saccadic search with gabor features applied to eye detection and real-time head tracking. *Image and Vision Computing*, 18(4):323–329, 1998.
- [30] J. Song, Z. Chi, and J. Liu. A robust eye detection method using combined binary edge and intensity information. *Pattern Recognition*, 39(6):1110–1125, 2006.
- [31] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [32] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [33] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, Hawaii, December 2001.
- [34] P. Wang, M. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 3, pages 164–171, San Diego, CA, June 2005.
- [35] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.

- [36] M-H Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.
- [37] Z. Zhu and Q. Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1):124–154, 2005.

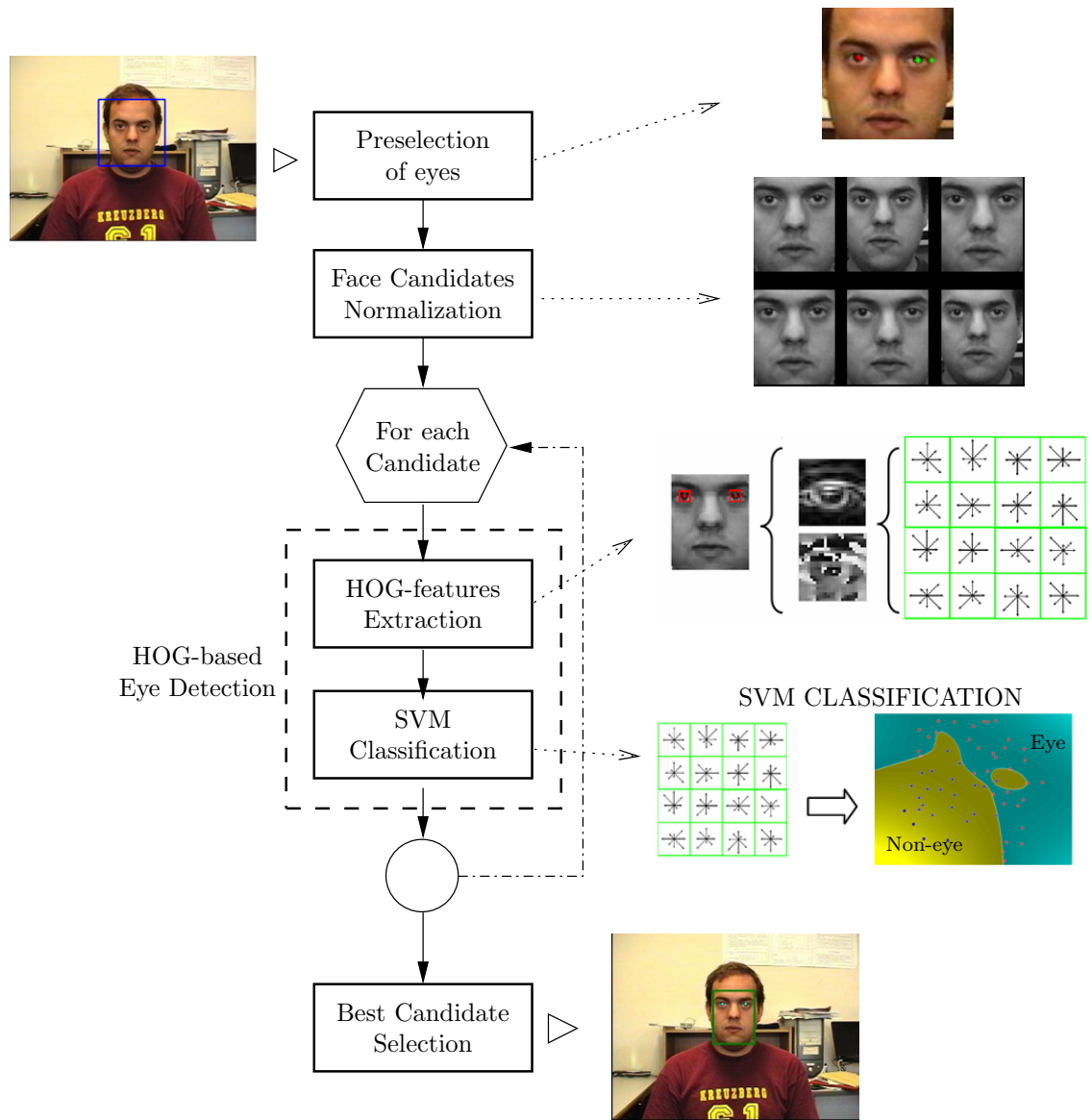


Figure 1: Accurate face detection block diagram

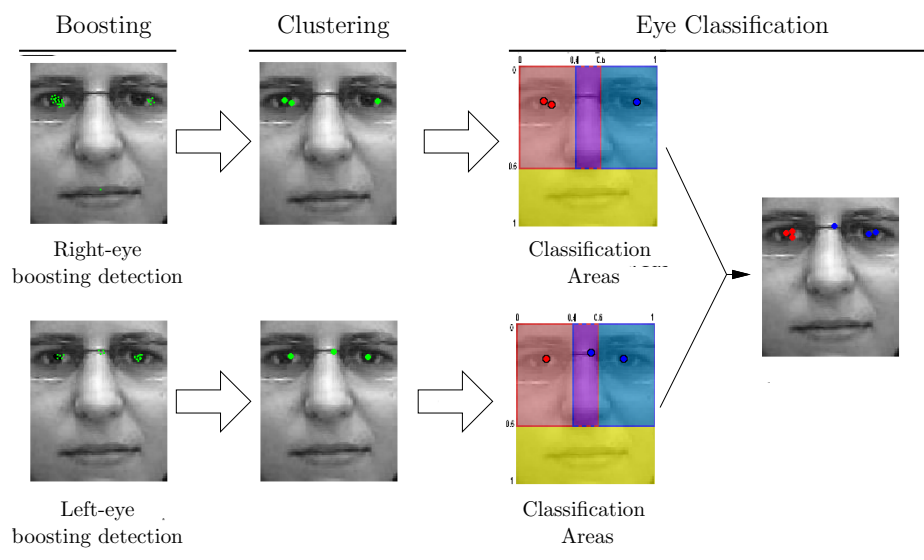


Figure 2: Scheme from the eye-detection stage

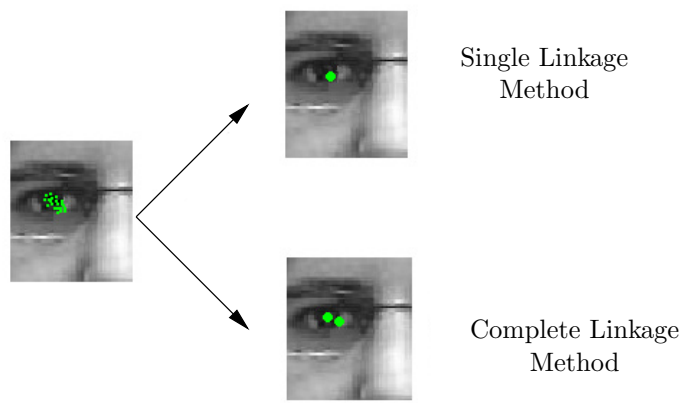


Figure 3: Criteria for clustering eye-detected candidates

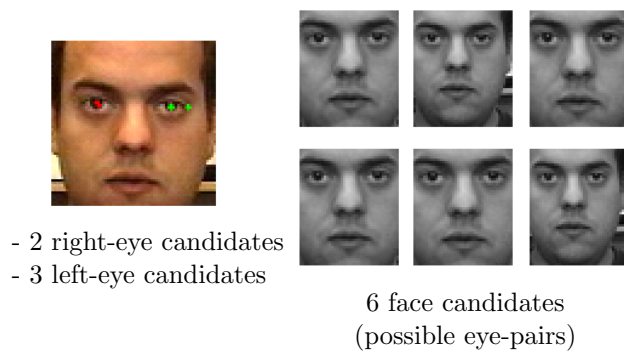


Figure 4: Face-candidates extracted from the eyes detected on a face-region

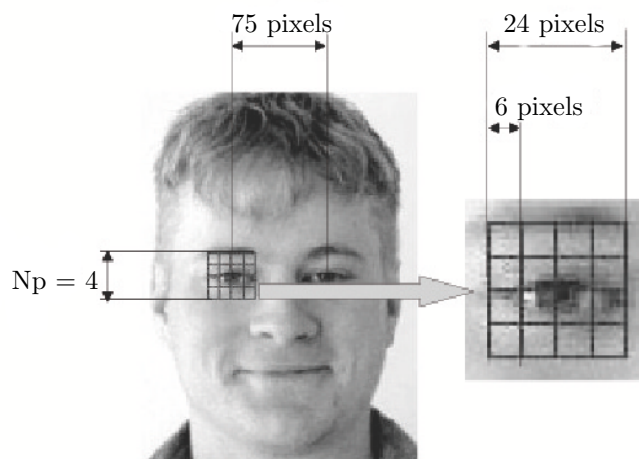


Figure 5: Normalized face and the spatial bins of the right eye descriptor



Figure 6: Example of two subjects from (a) FERET database, (b) FRGCv1 database and (c) CVL database.

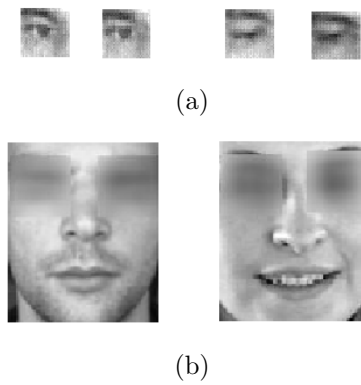
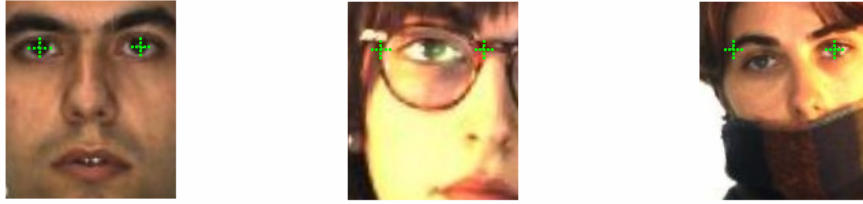


Figure 7: (a) Positive samples of the eye-database. (b) Negative samples of the eye-database.



Two positive eye samples Two negative eye samples One negative sample and a discarded positive sample (on the right)

Figure 8: Three possibilities of face-candidates from which training samples are extracted

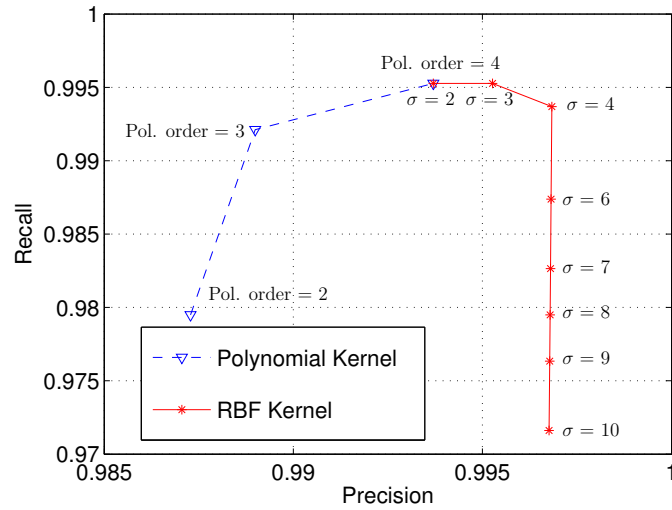


Figure 9: Precision-recall plot obtained for different SVM machines

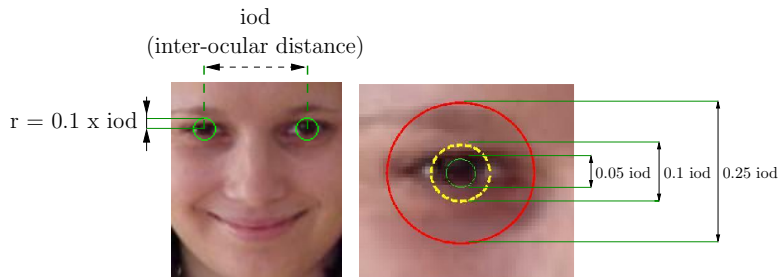


Figure 10: Magnitude of eye detection errors relative to the IOD

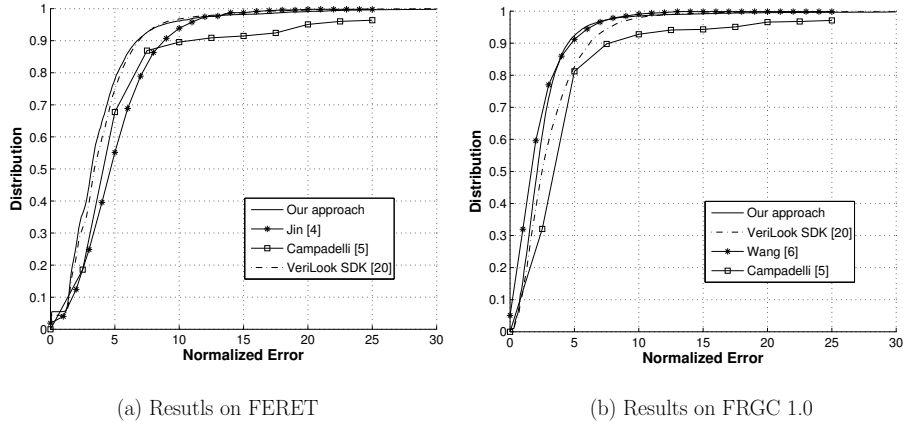


Figure 11: Comparison of accumulated eye localization errors for (a) the FERET database and (b) the FRGCv1 database.

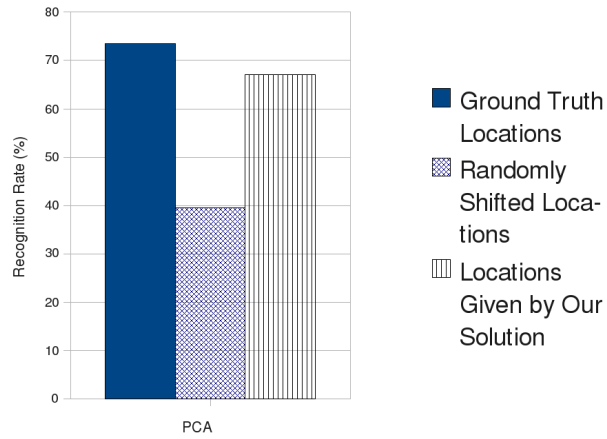


Figure 12: Influence of eye detection on face recognition using FERET data.