Combining Face and Eye Detectors in a High-Performance Face-Detection System

Loris Nanni
University of Padua, Italy

Alessandra Lumini
University of Bologna, Italy

A combined face and eye detector system based on multiresolution local ternary patterns and local phase quantization descriptors can achieve noticeable performance improvements by extracting features locally.

Face detection has become a fundamental task in computer vision and pattern recognition over the past two decades. Applications for such systems include face recognition, surveillance and security, human-computer interaction, behavioral analysis, and content-based video indexing.[1]

Given an arbitrary image, a face-detection system attempts to locate all existing faces in the image and return a bounding box (or an ellipsoid) enclosing each face. We can reformulate this problem as a one-class pattern recognition problem: for each subwindow of a given image, discriminate whether or not it contains a face. However, most proposed approaches treat face detection as a two-class classification task, training the face detector using face samples and nonface samples.[2]

Currently, several proposed face-detection methods provide accurate detection under varying conditions.[3] However, some applications also require additional information from a face image, such as eye or mouth position, with the aim of capturing eyes or lip movements.[1] To this end, researchers have developed eye, nose, and mouth detectors.[4] Eye detection has become a crucial step in several applications including eye tracking, feature-based face recognition, face modeling, and facial-expression recognition. In addition, many studies have reported a strong relation between face-recognition performance and eye-localization accuracy.[5]

Eye-detection research concerns determining if an image contains eyes and then accurately detecting eyes position, measured using the pupil or iris center. In particular, accurately detecting eye position is crucial to avoiding the *curse of misalignment*—that is, the loss of performance of face-recognition systems due to inaccurate eye localization.[6] Although researchers have made significant efforts in developing precise eye-localization methods,[5] this remains an open problem due to increasing demands for accuracy and speed in real-life applications.

Works published on eye detection in the last decade can be categorized as either holistic or abstract.[5] The holistic group includes works that attempt to locate the eyes using global representations, such as modular eigenspaces and hidden Markov model (HMM) based algorithms.[3] The abstract group includes works that extract discrete local features and use standard pattern recognition techniques to locate the eyes using these features.[5]

We propose an abstract eye-detection approach for upright frontal faces to improve face-detection accuracy and reliability. Specifically, we have developed an eye detector and incorporated it into an ensemble face-localization system that combines various face- and eye-detection components.

Eye Classification

Our eye classification module is based on two of the best-performing texture descriptors proposed in the literature—multiresolution local ternary patterns (LTP)[7] and local phase quantization (LPQ)[8] descriptors. Both descriptors are evaluated locally, dividing the suspected eye region into four subwindows for extracting the features separately inside each subwindow. We then couple the extracted features with support vector machines (SVMs) trained by eye and noneye samples to perform

1070-986X/12/$31.00 © 2012 IEEE

Published by the IEEE Computer Society

classification. Figure 1 shows a complete schema of the approach.

As a preprocessing step before feature extraction, to reduce the nonuniform lighting problem, the contrast of each input image is enhanced using histogram equalization. (We used the Matlab function histeq.m.) Then the image is divided into four equal subwindows (see Figure 1), extracting three texture descriptors from each subwindow and classifying them by a SVM ensemble combined using the sum rule. The descriptors are evaluated locally in each subwindow instead of in the whole image to improve the method's robustness.[9]

We used LTP texture descriptors with two different parameter settings, named LTP8 and LTP16, as well as LPQ texture descriptors. Each of the three descriptors (LTP8, LTP16, LPQ) extracted from each of the four subwindow is classified by a SVM. Therefore, the final ensemble consists of 12 classifiers (three descriptors for each of the four subwindows), combined by the sum rule.

LTP is a recent and interesting variant of the original local binary pattern (LBP),[7] which researchers have widely used as texture descriptor for several applications including eye detection.[10] LTP uses a ternary rather than a binary pattern, so the final pattern can be split into two binary patterns by considering its positive and negative components according to a threshold τ. ($\tau = 3$ in our experiments.) As a final descriptor, we calculate the concatenation of the "positive" and "negative" binary histogram. (See related work for mathematical details.[7]). In our experiments, two LTP descriptors were extracted by varying the parameter setting: the number of pixels in the neighborhood P and the search radius R was set to ($P = 8$, $R = 1$) in LTP8 and to ($P = 16$, $R = 2$) in LTP16. (We use the uniform bins.)

LPQ is a recent local method based on the quantized phase of the discrete Fourier transform (DFT) computed in a local subregion of the image (see www.cse.oulu.fi/CMV/Downloads/LPQMatlab). In this work, we used Gaussian derivative quadrature filters (with $R = 3$) for local frequency estimation.[8]

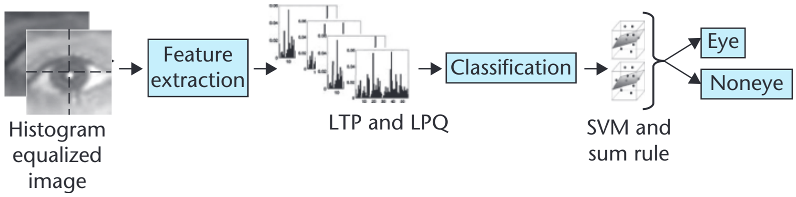

Figure 1. Proposed eye-classification system. Our eye-classification module combines multiresolution local ternary patterns (LTP) and local phase quantization (LPQ) descriptors and support vector machines (SVMs) trained by eye and noneye samples to perform classification.

Eye Detector

Our proposed eye-detection system (ED) takes the bounding box of a presumed face as input and gives the presumed eye positions as output. We then use ED as a module in the complete face-detection system to supply precise eye localization.

Figure 2 depicts a complete schema of the eye-detection system. Starting from the

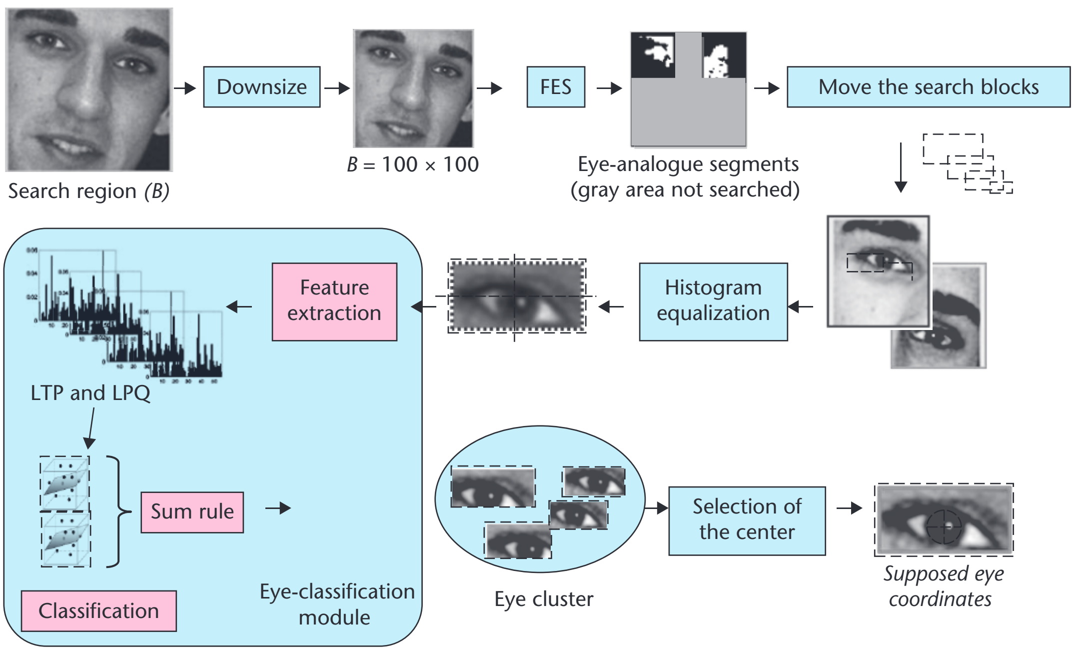

Figure 2. An outline of the eye-detection system (ED). We use the finding eye-analogue segments (FES) approach to reduce the search area within a face. The white pixels represent eye-analogue segments.

Our complete face-detection system combines the SNOW and EML face detectors and our eye detector (ED), which we use to refine the eye position of SNOW.

bounding box B of the face region, the eye localization is performed only in the face's upper region, discarding the bottom and central regions. (We discarded 50 percent of the pixels in the bottom and 20 percent from the central region.) Moreover, we use the finding eye-analogue segments (FES) approach proposed by Jianxin Wu and Zhi-Hua Zhou to reduce the search area within a face.[11] The eye-analogue segments (represented as white pixels in Figure 2) are used to limit the computational cost of the search.

We next resize B to 100×100 pixels and perform the search by moving a block of $d \times (1/2\ d)$ pixel inside B, where d is fixed to 15 percent of the image size (or 15 pixels). The block is moved inside B according to steps of five pixels. The number of blocks to be processed is restricted to approximately 50 by selecting only those including the higher number of eye-analogue segments. Each block is classified as eye or non-eye according to the eye classifier we described in the last section. The classification process consists of preprocessing, extracting features, and classifying each block. We set the classification threshold φ low to maximize the detection rate, even though a loss of accuracy might occur when detecting the eye positions.

To find eyes of various sizes, the search is performed using blocks of different sizes. (We used a scale factor of 1.35 at each iteration). We defined a cluster as a set of adjacent blocks classified as eyes. By overlapping all the resulting blocks, we obtain a final region that includes the supposed eye coordinates. To fix the best position of the supposed eye inside this final region, we tested two strategies: the center of the cluster and the darkest pixel if it is within a fixed distance γ from the center of the cluster (otherwise, the center itself). According to the experiments we report later on, the second strategy obtained the best performance.

Moreover, we applied some heuristic criteria in postprocessing. First, we removed huge clusters that contained more than 10 percent of the image's pixels. Second, if only one eye is found, another one is arbitrary located in a symmetric position on the other side of the image. Lastly, if two pairs of eyes are found, only the nearest to the center of the image is retained (supposing that the other is a false positive due to the presence of eyebrows).

Face and Eye Detector Ensemble

Our complete face-detection system combines two face detectors, SNOW[12] and EML,[5] and our eye detector (ED), which we use to refine the eye position of SNOW. This system was inspired by the earlier work of Bart Kroon and his colleagues;[9] their system consisted of the standard OpenCV face detector and an eye localizer to refine the eye position.

The face detector SNOW is based on local successive mean quantization transform (SMQT) features for describing a given image and the split up sparse network of winnows (SNoW, www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=13701&objectType=FILE) as a classifier. SNOW performance can be tuned by a sensitivity parameter, σ, measuring the maximum similarity to the face class allowed for classifying an image. The parameter σ can vary in the interval $[\sigma_{\min}, \sigma_{\max}]$. (In the original implementation, $\sigma_{\min} = 1$ and $\sigma_{\max} = 10$.) Our system combines the results obtained by fixing σ to $\sigma_{\min}$ and $\sigma_{\max}$ to exploit both results. A low σ value determines the presence of some false positives, and a high sensitivity misses some existing faces but retains few false positives.

EML is a simplified version of the approach described in earlier work[5] that returns the eye position with a high accuracy when the face is found, but with a face detection rate lower than SNOW (see http://lipori.dsi.unimi.it/download.html).

Figure 3 outlines our complete face-detection system. First, we combine the output of two face detectors (EML and SNOW with $\sigma = \sigma_{\max}$).

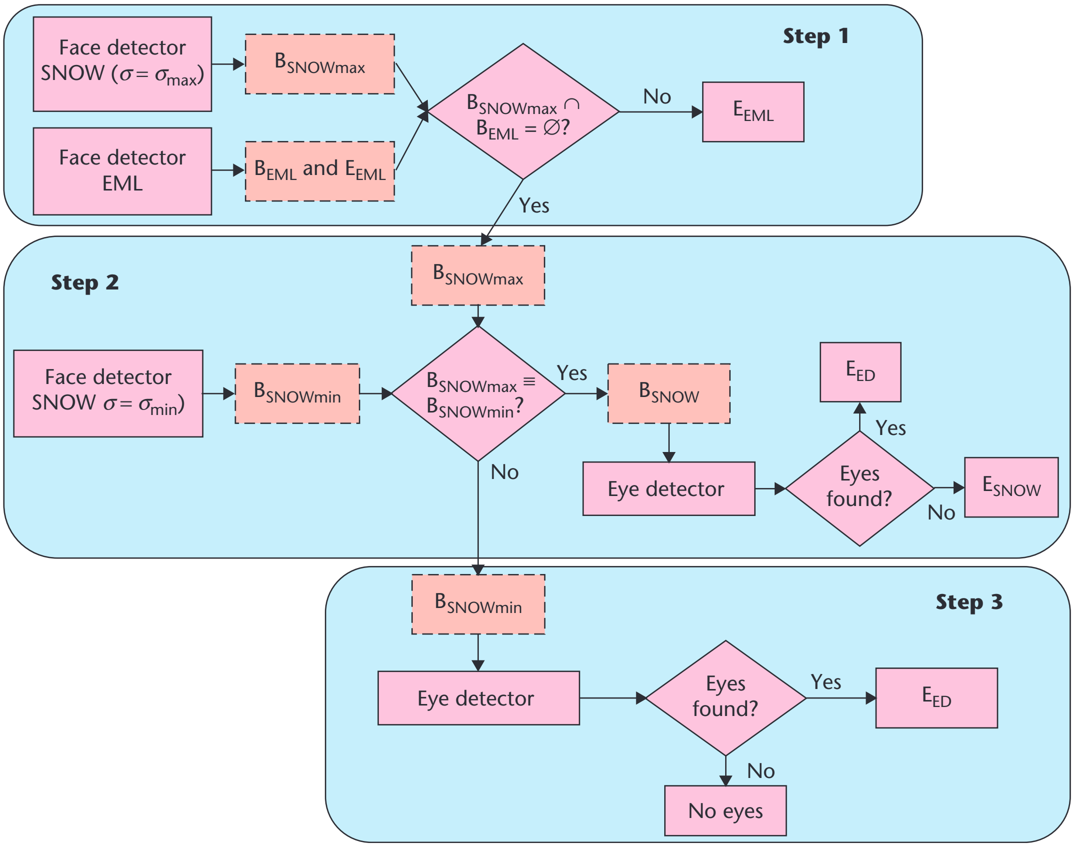

Figure 3. Outline of our complete face-detection system. The system first combines and compares the SNOW and EML detector results and then uses our eye detector (ED) to refine the eye position of SNOW.

The EML component gives as output both the bounding box of the assumed face (B_{EML}) and the assumed eyes position (E_{EML}), whereas the SNOW component gives only the bounding box ($B_{SNOWmax}$) and expects to find eyes at a fixed position (E_{SNOW}) inside it. Without loss of generality, we assume that each component finds only one or no supposed faces per image; the method is iterated if the number of bounding boxes is greater than one.

In this first step, B_{EML} and $B_{SNOWmax}$ are compared to determine if they overlap. That is, if the Euclidean distance between their supposed eyes positions is lower than a fixed threshold υ, we accept the eye positions returned by EML (E_{EML}). The rationale of this choice is that we use the SNOW approach to validate the EML output, which is accurate when it finds a face.

Otherwise, the system goes on to step two. We retain the bounding box $B_{SNOWmax}$ (if it exists), and we perform face localization using SNOW with $\sigma = \sigma_{min}$ (resulting in $B_{SNOWmin}$). If $B_{SNOWmin}$ and $B_{SNOWmax}$ overlap—that is, the same supposed face B_{SNOW} is found—we use ED to find the eye position (E_{ED}) or we validate the eye position given by SNOW (E_{SNOW}). The output bounding box $B_{SNOWmin}$ of the face region given by SNOW (with $\sigma = \sigma_{min}$) is the starting search region for ED.

The third step evaluates a supposed face returned by SNOW with $\sigma = \sigma_{min}$ if it has not been validated in step two—that is, for the set $B_{SNOWmin} - B_{SNOWmax}$, note that $B_{SNOWmax} \subseteq B_{SNOWmin}$—and we search for the eye position using our eye detector. If ED does not find an eye, the image is considered a false positive and rejected; otherwise, E_{ED} is returned as the system's output.

Experimental Results

To evaluate our approach, we performed several tests to measure the performance of combining descriptors and extracting features from four subwindows of the eye region rather than extracting from the whole image. We also tested the performance of the complete ED system versus partial implementations and previous research.

Eye Classification

The first test aimed at validating the set of texture descriptors used for eye classification. All the methods are based on a stand-alone SVM

Table 1. Area under ROC curve (AUC) performance in the eye-classification problem.

AUC	Descriptors*					
	LBP8 (P = 8, R = 1)	LBP16 (P = 16, R = 2)	LTP8 (P = 8, R = 1)	LTP16 (P = 16, R = 2)	LPQ	LTP8, LTP16, and LPQ
Whole eye	0.850	0.895	0.860	0.902	0.910	0.923
Four subwindows	0.940	0.947	0.940	0.956	0.966	0.971

* LBP is local binary pattern, LTP is local ternary patterns, and LPQ is local phase quantization.

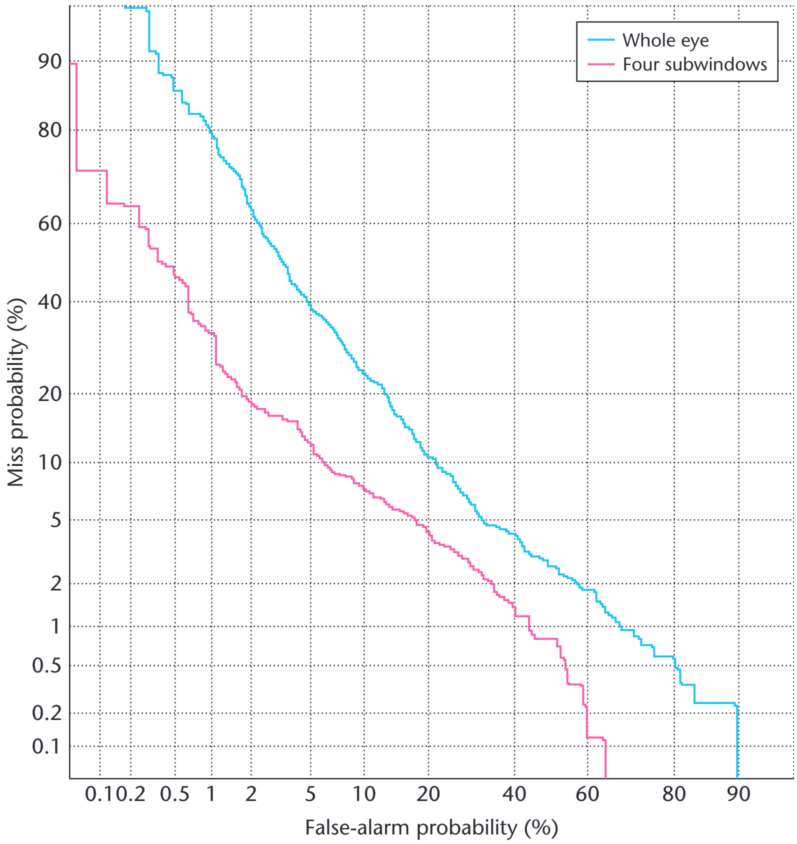

Figure 4. Detection error trade-off (DET) curve of the whole eye and four subwindows approaches.

classifier (if the features are extracted from the whole image) or an ensemble of SVMs if the feature extraction is performed in subwindows or more descriptors are used.

We trained all the classifiers using the FRGC_Eye_Centered dataset for positive samples (www.ecse.rpi.edu/~cvrl/database/ISL_IR_Eye_Database.htm), including 1,000 images of open eyes with a resolution higher than 15 × 30. We also extracted 25,000 noneye samples from the Yale-B dataset (http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html) and downloaded images from the Web. All the system parameters (φ and γ) were calculated using a different image dataset shared by Ilia V. Safonov.[13]

The test set is the "candidate red-eyes" dataset (shared by Sebastiano Battiato and his colleagues[14]), which includes 2,513 images (848 eyes and 1,665 noneyes) at a 30 × 30 pixel resolution extracted from photographs through an image-filtering pipeline. The presence of red eyes did not affect our results because all experiments were conducted on gray-scale images. Our experiment classified each image as an eye or noneye.

Because the dataset includes eyes of different sizes, we used moving windows (with different sizes and angles) to perform the search, and we combined by max rule the scores obtained from each window (as we described earlier).

Table 1 compares the classification performance of the different approaches obtained by varying the descriptors and the feature-extraction modality. The performance indicator is the area under the ROC curve (AUC).

Figure 4 shows the detection error trade-off (DET) curve of the whole eye and four subwindows approaches (coupled with the three descriptors). The DET curve plots on logarithmic scales the false-negative rate (missed probability) versus the false-positive rate (false-alarm probability). Our experiments considered 1,000 true eyes and 25,000 negative samples. The SVM parameters are chosen by a 10-fold cross validation on the training data.

Based on the results in Table 1 and Figure 4, we can draw several conclusions. First, LTP outperforms LBP for this problem. Second, by combining the best three descriptors, we outperform each stand-alone descriptor. And lastly, we can obtain better performance by extracting features from four subwindows of the eye region than by extracting from the whole image.

Eye/Face Detection

Our second test aimed at comparing the face-detection performance. We used the complete ED for face detection as a final step of SNOW—that is, the output bounding box $B_{SNOWmin}$ given by SNOW with $\sigma = \sigma_{min}$ was used as the starting searching region for ED. We also tested the performance of the following systems:

- *St1*, a partial version of the face and eye detector ensemble system that consists of only the first step (all the images not classified by this step are labeled as not containing a face);
- *St2*, a partial version of the ensemble system that consists of steps one and two (all the images not classified by this step are labeled as not containing a face);
- *St3*, the complete face and eye detector ensemble system (steps one through three);
- the face detector SNOW with different values of the sensitivity parameter σ; and
- the face detector EML, with both the results obtained by the free code (http://lipori.dsi.unimi.it/download.html) and those published in and earlier work.[5]

Moreover, we also report the performance on BioID gained by other well-known approaches, including the first work on BioID,[15] the work that inspired our approach,[9] the FES approach,[11] and some of the most recent state-of-the-art approaches.[16–18]

We performed our experiments on three datasets, all containing upright frontal images:

- the BioID dataset, which consists of 1,521 images of 23 different people recorded during several sessions;
- a subset of the Feret dataset[5] that included 1,175 pictures (200 of type "ba," 400 "fa," and 575 "fb"); and
- SC, a self-collected dataset consisting of 46 images of approximately 20 people with different backgrounds.

(The parameters setting was performed on a different dataset.[13])

To confirm the presence of a face, we calculated the detection relative error (DER)[15] by comparing the manually extracted left and right eyes positions C_l and C_r and the detected eyes positions C_l' and C_r'. Let d_l be the Euclidean distance between C_l and C_l', d_r be the Euclidean distance between C_r and C_r', and d_{lr} be the Euclidean distance between C_l and C_r. The relative error of detection is defined as $\mathrm{DER} = \max(d_l, d_r)/d_{lr}$. According to the detection criterion in an earlier work,[13] DER must be lower than 0.25 to confirm the detection, which means that the larger of d_l and d_r is less than half the eye width.

We measured the eye/face detection performance according to the following indicators (see Table 2). The detection rate (DR, or recall) is the percentage of images where a face is detected ($\mathrm{DER} < 0.25$). The average detection relative error (ADER) is evaluated on the whole dataset; this performance indicator should be minimized. We also measured the number of false-positive faces (NFP) retrieved and precision (PR), which is the percentage of true positive faces retrieved with respect to the supposed faces: PR = true positive/(true positive + false positive). The NFP performance indicator should also be minimized.

The focus of our research was localization accuracy rather than real-time computation. Our method provides an acceptable solution for face detection, allowing for acceptable computational performance. Table 3 reports the computation time (QuadCore, 2.8 GHz, Matlab code, when the parallel toolbox is used) of each step of our algorithm together with the probability of performing the step—that is, the percentage of images that stop the computation at that step. The last column reports a weighted average of the times.

We can draw several conclusions from these experiments. First, our proposed method ED is one of the few methods in the literature that, coupled with a face-detection approach, generates a DR higher than 99 percent on the BioID. Second, St2 provides a good trade-off between the two performance indicators DR and NFP—that is, the system performs well with a low number of false positives. Lastly, the comparison of the results on different datasets shows that our system has a low sensitivity to the parameter tuning. (A few parameters of the systems have been optimized on a different dataset.)

*Table 2. Face-detection performance.**

Eye-detection approaches		BioID DR (%)	BioID ADER	BioID NFP	BioID PR (%)	FERET DR (%)	FERET ADER	FERET NFP	FERET PR (%)	Self-collected dataset DR (%)	Self-collected dataset ADER	Self-collected dataset NFP	Self-collected dataset PR (%)
ED	Strategy i	99.1	0.109	26	98.3	99.4	0.089	22	98.2	100	0.126	8	85.2
	Strategy ii	98.9	0.096	29	98.1	99.3	0.086	22	98.2	100	0.124	8	85.2
St1	$\nu = 10$	67.1	0.036	3	99.7	73.1	0.033	2	99.8	54.3	0.038	0	100
	$\nu = 15$	87.1	0.041	14	98.9	91.1	0.031	13	98.9	84.8	0.037	0	100
St2		97.8	0.061	8	99.5	98.8	0.051	6	99.5	99.5	0.075	0	95.6
St3		99.3	0.061	26	98.3	99.7	0.050	23	98.1	98.1	0.079	8	100
SNOW	$\sigma = \sigma_{min}$	98.5	0.118	39	97.5	99.5	0.108	33	97.3	97.8	0.136	12	78.9
	$\sigma = \sigma_{max}$	97.0	0.118	20	98.6	99.0	0.110	15	98.7	97.0	0.136	1	97.8
EML	Free code	93.0	0.042	81	94.6	94.0	0.040	75	94.0	93.6	0.039	8	84.3
	Campadelli[5]	99.3	–	–	–	99.7	–	–	–	–	–	–	–
Safonov 2001[13]		91.8	–	–	–	–	–	–	–	–	–	–	–
Wu and Zhou 2003[11]		94.5	–	–	–	–	–	–	–	–	–	–	–
Tang and colleagues 2005[18]		98.1	–	–	–	–	–	–	–	–	–	–	–
Cristinacce and Cootes 2006[16]		98.5	–	–	–	–	–	–	–	–	–	–	–
Kim and colleagues 2007[17]		98.8	–	–	–	–	–	–	–	–	–	–	–
Kroon and colleagues 2009[9]		96	0.0365	171	89	–	–	–	–	–	–	–	–

* Detection rate (DR, or recall), average detection relative error (ADER), false positive faces retrieved (NFP), and precision (PR).

Table 3. Computation time to perform face detection on an image of the BioID dataset.

Computation time	St1	St2	St3	St4
Time (sec)	0.7	5.3	5.3	2.2
Percentage	67.3	31	1.7	–

The significance of each performance indicator is strictly related to the final application. For example, an augmented-reality application could require high accuracy in eye detection (high ADER and PR), whereas a face-authentication application might prefer a system with a high detection rate if the matching algorithm uses additional poses.[6] The parameter optimization performed in this work mainly aims at maximizing DR because we were interested in the face-authentication problem. We have tested the recognition performance of the Eigenface method trained using additional poses.[6] Our experiments performed on the BioID dataset obtained the same equal error rate (6 percent) using both a "perfect" manual detection (ADER = 0) and the proposed system ED (ADER = 0.1), thus proving that minimizing ADER is not so important for face authentication.

Conclusions

Our experimental results showed that our eye detector, based on multiresolution LTP and LPQ descriptors extracted by dividing the eye region into four subwindows, achieved noticeable performance improvement with respect to existing methods based on a feature extraction from the whole eye region. Moreover, our experiments showed that both our system ED, which is simpler than others presented in the literature,[5,7] and our ensemble system St3 obtain a face-detection rate greater than 99 percent (one of the highest reported in the literature) on the BioID dataset.

One of the main drawbacks of the proposed system is that it retains some false positives to reach a high detection rate. To reduce the number of false positives, in future work, we plan to couple our system with a skin-detection module. A preliminary test conducted on our self-collected dataset shows that a simple skin detector helps us reduce the number of false positives using St3 from 8 to 3 (although it requires color images). Nonparametric histogram-based models were trained using manually annotated skin and nonskin pixels (see http://clickdamage.com/sourcecode/index.php). A Matlab version of our complete system for face detection is available at http://bias.csr.unibo.it/nanni/fd.rar. **MM**

Acknowledgments

We thank Silvia Franci, who helped us with this project as partial fulfillment of her bachelor's degree and who collected the SC dataset.

References

1. Z. Zeng et al., "A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 1, 2009, pp. 39–58.
2. H.L. Jin, Q.S. Liu, and H.Q. Lu, "Face Detection Using One-Class Based Support Vectors," *Proc. 6th IEEE Int'l Conf. Automatic Face Gesture Recognition,* IEEE CS Press, 2004, pp. 457–462.
3. C. Zhang and Z. Zhang, *A Survey of Recent Advances in Face Detection,* tech. report MSR-TR-2010-66, Microsoft Research, Jun. 2010.
4. D. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 3, 2010, pp. 478–500.
5. P. Campadelli, R. Lanzarotti, and G. Lipori, "Precise Eye and Mouth Localization," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 23, no. 3, 2009, pp. 359–377.
6. J. Min, K. Bowyer, and P. Flynn, "Eye Perturbation Approach for Robust Recognition of Inaccurately Aligned Faces," *Proc. 5th Int'l Conf. Audio- and Video-Based Biometric Person Authentication* (AVBPA), Springer-Verlag, 2005, pp. 41–50.
7. X. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions," *Analysis and Modelling of Faces and Gestures,* LNCS 4778, Springer, 2007, pp. 168–182.
8. V. Ojansivu and J. Heikkila, "Blur Insensitive Texture Classification Using Local Phase Quantization," *Proc. Int'l Conf. Image and Signal Processing* (ICISP), Springer-Verlag, 2008, pp. 236–243.
9. B. Kroon et al., "Eye Localization in Low and Standard Definition Content with Application to Face Matching," *Computer Vision and Image Understanding,* vol. 11, no. 8, 2009, pp. 921–933.
10. R. Sun and Z. Ma, *Robust and Efficient Eye Location and Its State Detection, Advances in Computation and Intelligence,* LNCS 5821, Springer, 2009, pp. 318–326.
11. J. Wu and Z.-H. Zhou, "Efficient Face Candidates Selector for Face Detection," *Pattern Recognition,* vol. 36, no. 5, 2003, pp. 1175–1186.
12. M. Nilsson, J. Nordberg, and I. Claesson, "Face Detection Using Local SMQT Features and Split Up SNOW Classifier," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing* (ICASSP), vol. 2, IEEE CS Press, 2007, pp. 589–592.
13. I.V. Safonov, "Automatic Red-Eye Detection," *Proc. Int'l Conf. Computer Graphics and Vision* (GraphiCon), 2007; http://xn–80afqipfk7a.xn–p1ai/2007/proceedings/Papers/Paper_11.pdf.
14. S. Battiato et al., "Red-Eyes Removal Through Cluster Based Linear Discriminant Analysis," *Proc. IEEE Int'l Conf. Image Processing* (ICIP), IEEE CS Press, 2010, pp. 2185–2188.
15. O. Jesorsky, K. Kirchberg, and R. Frischholz, "Robust Face Detection Using the Hausdorff Distance," *Proc. Int'l Conf. Audio- and Video-Based Biometric Person Authentication,* Springer-Verlag, 2001, pp. 90–95.
16. D. Cristinacce and T.F. Cootes, "Feature Detection and Tracking with Constrained Local Models," *Proc. British Machine Vision Conf.,* vol. 3, 2006, pp. 929–938;http://nguyendangbinh.org/Proceedings/BMVC/2006/papers/024.pdf.
17. S. Kim et al., "Multi-scale Gabor Feature Based Eye Localization," *Proc. World Academy of Science, Eng. and Technology,* vol. 21, Jan. 2007, pp. 483–487.
18. X. Tang et al., "Robust Precise Eye Location by AdaBoost and SVM Techniques," *Proc. Int'l Symp. Neural Networks,* Springer-Verlag, 2005, pp. 93–98.

Loris Nanni is an associate researcher in the Department of Information Engineering at the University of Padua, Italy. His research interests include pattern recognition, bioinformatics, and biometric systems (fingerprint classification and recognition, signature verification, and face recognition). Nanni has a PhD in computer engineering from the University of Bologna. Contact him at loris.nanni@unipd.it.

Alessandra Lumini is an associate researcher at the University of Bologna, Italy. Her research interests include pattern recognition, bioinformatics, biometric systems, multidimensional data structures, digital image watermarking, and image generation. Lumini has a PhD in computer engineering from the University of Bologna. Contact her at alessandra.lumini@unibo.it.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*