Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection

Christophe Garcia and Manolis Delakis

Abstract—In this paper, we present a novel face detection approach based on a convolutional neural architecture, designed to robustly detect highly variable face patterns, rotated up to ± 20 degrees in image plane and turned up to ± 60 degrees, in complex real world images. The proposed system automatically synthesizes simple problem-specific feature extractors from a training set of face and nonface patterns, without making any assumptions or using any hand-made design concerning the features to extract or the areas of the face pattern to analyze. The face detection procedure acts like a pipeline of simple convolution and subsampling modules that treat the raw input image as a whole. We therefore show that an efficient face detection system does not require any costly local preprocessing before classification of image areas. The proposed scheme provides very high detection rate with a particularly low level of false positives, demonstrated on difficult test sets, without requiring the use of multiple networks for handling difficult cases. We present extensive experimental results illustrating the efficiency of the proposed approach on difficult test sets and including an in-depth sensitivity analysis with respect to the degrees of variability of the face patterns.

Index Terms—Face detection, neural networks, machine learning, convolutional networks.

1 Introduction

Face detection is becoming a very important research topic, due to its wide range of possible applications, like security access control, model-based video coding, content-based video indexing, or advanced human and computer interaction. It is also a required preliminary step to face recognition and expression analysis.

Numerous approaches for face detection have been proposed in the last decade, many of them described and compared in two interesting recent surveys by Yang et al. [1] and Hjelmas et al. [2]. Most face detection methods are based on local facial feature detection and classification using statistical and geometric models of the human face. Low level analysis first deals with the segmentation of visual features using image properties such as edges [3], intensity [4], color [5], [6], motion [7], or generalized measures [8]. Other approaches are based on template matching where several correlation templates are used to detect local subfeatures, considered as rigid in appearance (eigenfeatures [9]) or deformable [10], [11]. Then, visual features are organized into a more global concept of face through facial feature and constellation analysis using face geometry constraints [11], [12], [13], [14].

The main drawback of feature-based approaches is that either little global constraints are applied on the face template or extracted features are significantly influenced by noise, occlusions, and changes in face expression and viewpoint. In order to handle difficult scenarios where multiple faces of different sizes and poses have to be detected in heavily cluttered backgrounds, some advanced image-based pattern recognition techniques have been developed. They avoid the specific and possibly inaccurate face modeling by learning underlying rules contained in highly variable face patterns from large training sets of face examples. They have proven to be very tolerant to noise and distortions affecting the face patterns.

In this paper, we propose a novel image-based approach that is designed to precisely detect face patterns of variable size and appearance, rotated up to ± 20 degrees in image plane and turned up to ± 60 degrees, in complex real world images. Our system is based on a convolutional neural network architecture, inspired from the work of LeCun et al. [15]. It automatically derives problem-specific feature extractors, from a large training set of face and nonface patterns, without making any assumptions about the features to extract or the areas of the face patterns to analyze. Once trained, our system acts like a fast pipeline of simple convolutions and subsampling modules, that treat the raw input image as a whole, for each analyzed scale, and does not require any costly local preprocessing before classification. Such a scheme provides very high detection rate with a particularly low level of false positives, demonstrated on difficult test sets, while maintaining an acceptable speed of approximately four frames per second for 384×288 pixel images, on a conventional 1.6 GHz Intel Pentium IV.

There have been previous successful image-based methods for face detection, which differ on the pattern classification techniques they apply or the constraints they consider on the face patterns. The first advanced image-based face detection system has been developed by Sung and Poggio [16]. This system consists of two components, a clustering and a distribution-based models for face/nonface patterns and a

- C. Garcia is with France Telecom R&D, 4 rue du Clos Courtel, 35512 Cesson Sévigné Cedex, France.
- M. Delakis is with IRISA-INRIA de Rennes, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France.
E-mail: {cgarcia, delakis}@csd.uoc.gr.

*Manuscript received 3 Feb. 2003; revised 8 Sept. 2003; accepted 8 Apr. 2004.
Recommended for acceptance by W.T. Freeman.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118219.*

multilayer perceptron classifier. Each face and nonface example (19×19 pixels) is first preprocessed via lighting correction, a best fit linear function being subtracted from the original signal, followed by histogram equalization. Then, the training patterns are classified into face and nonface clusters using a modified k-means algorithm. A vector of appropriate distances, computed between the tested image window and each cluster centroid is presented as input to a multilayer perceptron network for classification. Some other efficient techniques are based on standard multivariate statistical analysis. Yang et al. [17] proposed two methods which seek to represent the manifold of human faces as a set of subclasses. In the first method, a mixture of factor analyzers is used to perform clustering and local dimensionality reduction within each obtained cluster. The second method uses Kohonen's self-organizing maps for clustering, Fisher's linear discriminant to find an optimal projection for pattern classification and a Gaussian distribution to model the class-conditional density function of the projected samples for each class. Maximum likelihood estimates are used for the parameters of the class-conditional density functions and the decision rule.

There are several other statistical approaches for face detection, based on information theory or Bayes decision rule. Colmenarez and Huang [18] proposed a system based on Kullback relative information (Kullback divergence), to measure the difference between joint-histograms, computed for each pair of pixels in the face images of the training set, for the classes of faces and nonfaces.

Garcia and Tziritas [5] proposed a face detector based on skin color segmentation and statistical classification of the face texture. Facial texture is described by vectors composed of simple statistical measures (variances) extracted from each subband of a discrete three-level wavelet packet decomposition of the face intensity image. Wavelet packet decomposition, which captures information regarding visual attributes in space, frequency, and orientation is found to be efficient for describing the characteristics of the human face. They use a pair of suitably chosen conjugate quadrature low-pass and high-pass filters that they have also incorporated in a face recognition system [19]. The extracted feature vectors are then classified as faces or nonfaces, using the Bhattacharrya distance and some prototypes of face pattern vectors derived from training.

Schneiderman and Kanade [20] proposed a face detector also based on a locally sampled three-level wavelet decomposition. Several sets of wavelet coefficients are extracted from chosen subbands of the wavelet tree. The coefficients are requantized to three levels and probabilistic density functions are built using histograms. Bayes' rule is applied for classification between face and nonface patterns.

More recently, Liu [21] proposed a Bayesian Discriminating Features (BDF) method for frontal face detection. The discriminating feature analysis combines the input image, its 1D Haar wavelet and its amplitude projections. Then, statistical modeling estimates the conditional probability density functions of the face and nonface classes, considering multivariate normal distributions, in order to build a Bayesian classifier. Other approaches, closer to the proposed method, rely on neural networks for learning separating surfaces between face and nonface subspaces. Osuna et al. [22] developed a support vector machine (SVM) approach to face detection. An SVM with a second-degree polynomial as a kernel function is trained from a large training set of faces with a decomposition algorithm, which guarantees global optimality. Approximately 2,500 support vectors are obtained and used for face detection. This system scans input images over scales with a 19×19 pixel window and performs lighting corrections of the window content before classifying it using the support vectors.

Rowley et al. [23] proposed the first advanced neural approach which reported results on a large and difficult data set. Their system incorporates face knowledge in a retinally connected neural network, looking at windows of 20×20 pixels. In their single neural network implementation (referred to as system 5), there are two copies of a hidden layer with 26 units, where four units look at 10×10 pixel subregions, 16 look at 5×5 subregions, and six look at 20×5 pixels overlapping horizontal stripes. A large number of adjustable weights (2,905) are learned through standard back-propagation. The input window is preprocessed via lighting correction like in the system of Sung and Poggio [16]. The image is scanned with a moving 20×20 pixel window at every possible position and for scales obtained with a subsampling factor of 1.2. To reduce the false alarm rate, they combine multiple neural networks with an arbitration strategy, which, however, increases the computational cost. They also proposed a faster version, based on a two-stage scheme, where a simple network is used to quickly discard nonface like areas and a more complex network is used to perform final classification on the image areas that passed the first stage successfully. A significant gain in speed is observed, but at the cost of a reduced detection rate.

Roth et al. [24] proposed a face detector based on a learning architecture called SNoW (Sparse Network of Winnows), which consists of two linear threshold units, representing the classes of faces and nonfaces, that operate on an input space of Boolean features. Features like intensity mean, intensity, and variance are first extracted from a series of subwindows from the face window and then discretized into a predefined number of classes to give Boolean features in a 135,424-dimensional feature space. The system is trained with a simple learning rule, which promotes and demotes weights in cases of misclassification, in order to classify face and nonface Boolean features. Like in the aforementioned methods, images are preprocessed using the technique of Sung and Poggio.

Féraud et al. [6] proposed another neural approach, based on constrained generative models (CGM), which are auto-associative fully connected MLPs with three large layers of weights, trained to perform a nonlinear PCA. Classification is obtained by considering the reconstruction errors of the CGMs. The best results are reported using a combination of CGMs via a conditional mixture and an MLP gate network. As the computational cost of this method is high, some prefiltering operations are required, such as skin color and motion segmentation. Like in the previous neural-based approaches, every tested subwindow is preprocessed using the approach of Sung and Poggio.

Most of these methods are based on a costly exhaustive multiresolution window scanning technique. The input image is successively subsampled by a factor of 1.2, giving a pyramid of images. In each subsampled image, a window is scanned at all positions, and its content is preprocessed via lighting correction and histogram equalization, before being processed by the neural architecture.

Lately, very fast approaches based on coarse-to-fine search mechanisms and focus of attention have been proposed [25], [26], [27]. The main philosophy of these approaches is to combine classifiers in a cascade, which allows background regions of the image to be quickly discarded while spending more computation on promising face-like regions. The detectors in the early stages are simpler so as to reject a vast number of nonface regions, whereas those at latter stages are more complex and take more time.

Fleuret and Geman [25] proposed a face detection technique which relies on a chain of binary tests applied on particular arrangements or disjunctions of fine scale edges. The cascade of tests is learned by density estimation and density discrimination. Fast processing is obtained but the false positive rate appears to be higher than for other state-of-the-art approaches.

Recently, Viola and Jones [26] proposed the first real-time frontal face detection system (running at 15 frames per second on conventional desktop), providing good detection results with an acceptable false alarm rate. They present a scheme for constructing weak classifiers by selecting a small number of simple features using a variant of the Adaboost algorithm. The simple features, which they use, are reminiscent of Haar basis functions, as they correspond to differences between the sums of pixels within adjacent rectangular regions of the image. Different features are obtained by considering different arrangements of adjacent rectangles of the same size and shape, and computed in a very fast way by using an "integral" representation of the original image. Weak classifiers are then boosted into stronger classifiers using a linear combination derived during the learning process. Finally, strong classifiers are organized into a cascade structure of increasing complexity, similar to a degenerated decision tree. Each image area to inspect is variance normalized before classification to minimize the effect of different lighting conditions.

Inspired by the work of Viola and Jones [26], Li et al. [27] proposed a real-time system for multiview face detection, using a new boosting algorithm, called FloatBoost, designed to overcome the problem of monotonicity of the sequential Adaboost training. The system also uses a coarse-to-fine, simple-to-complex architecture that demonstrated good performances in detecting nonfrontal faces.

Most of these techniques rely on image preprocessing for reducing the variability of the face patterns before the training and classification stages. Our system provides high detection rate with a particularly low level of false positives, while avoiding local preprocessing to ensure that no unrealistic assumption is made about the data and that the richness of the original signals is preserved. In addition, it does not require the use of independently trained multiple networks for handling difficult cases or reducing false alarm rates.

In Section 2, we describe the design of our architecture and the training methodology in detail. Then, we present the process of face detection using this architecture. In Section 3, we intend to assess the robustness of our method with respect to the degrees of variability of the face patterns. We also examine the performance of our face detector on different test sets of images, including the *CMU* test set [23], that we use to compare our method with state-of-the-art approaches. Finally, we conclude this paper with comments and description of future work in Section 4.

2 The Proposed Approach

Our face detector is designed to locate multiple faces of 20×20 pixel minimum size, rotated up to ± 20 degrees in image plane and turned up to ± 60 degrees. It consists of a pipeline of convolutions and subsampling operations, applied at various scaled versions of the original image, to handle faces of different sizes. This pipeline performs automatic *feature extraction* and *classification* of the extracted features, in a single integrated scheme. The full process is implemented via a convolutional neural network architecture, which offers the advantage of being trained to automatically derive all parameters, governing feature extraction and classification.

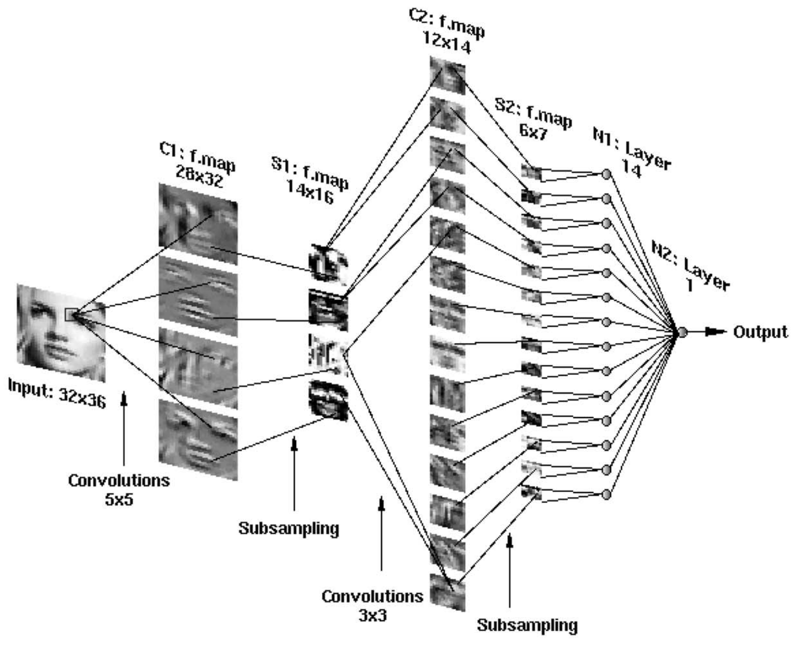
Convolutional Neural Networks (CNN), introduced and successfully used by LeCun et al. [15], [28], [29], are powerful bioinspired hierarchical multilayered neural networks that combine three architectural ideas to ensure some degree of shift, scale, and distortion invariance: *local receptive fields*, *shared weights*, and *spatial subsampling*. Different architectures of convolutional networks have been used successfully in many difficult applications such as handwriting recognition [29], [30], machine-printed character recognition [31], and face recognition [32].

Vaillant et al. [33] used convolutional networks for image-based object detection and considered the case of face detection. The key contribution of this approach was to show that a 2-layer CNN architecture outperformed fully connected multilayer perceptrons and that a good classification between reduced sets of face and nonface patterns could be obtained. The paper focuses mainly on the analysis and the comparison of training results. Faces used for training are obtained in a controlled environment with only two different lighting conditions (light source behind the camera and diffuse lighting). Partially occluded faces and faces wearing glasses are excluded from the training and testing sets. Nonface patterns are extracted randomly from sequences taken in the same indoor environment. Experimental results mainly focus on the problem of classification between faces and a fixed set of nonfaces. The application of the CNN architecture to face detection is proposed by considering differently trained versions of the same architecture for complete, rough, and fine localization. Each image is preprocessed by applying a zero-mean Laplacian filter, and normalized to have a mean of zero and a standard deviation of one. Results concerning the classification of face and nonface sets with this simple 2-layer architecture were promising, in comparison with traditional MLPs, but unfortunately no quantitative results have been reported on real world test images.

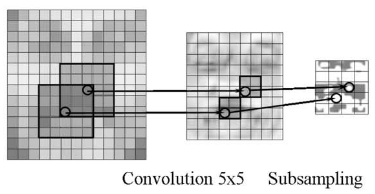
Hereafter, we present in detail the proposed architecture, the methodology to train it and, finally, the way it can be efficiently applied for multiscale face detection.

2.1 Convolutional Neural Network Architecture

The Convolutional Neural Network used in our experiments, which we called *CFF (Convolutional Face Finder)*, is shown in Fig. 1a. It consists of six layers, excepting the input plane (retina) that receives an image area of size 32×36 pixels to be classified as *face* or *nonface*. Layers C1 through S2 contain a series of planes where successive convolutions and subsampling operations are performed. These planes are called *feature maps* as they are in charge of extracting and combining a set of appropriate features. Layer N1 contains a number of



(a)



(b)

Fig. 1. (a) The six successive layers of *CFF*: convolutions and subsampling operations are alternated. (b) receptive fields for convolution and subsampling for a feature map of layers C1-S1.

partially connected sigmoid neurons and layer N2 contains the output unit of the network. These last two layers carry out the classification task using the features extracted in the previous layers.

Each unit in a layer receives input from a set of units located in a small neighborhood in the previous layer as shown in Fig. 1b. The idea of connecting units to *local receptive fields* on the input was largely inspired by Hubel and Wiesel's discovery of locally-sensitive, orientation-selective neurons in the cat visual system [34] and local connections have been used many times in neural models of visual learning [28], [35], [36]. With local receptive fields, neurons can extract elementary visual features such as oriented edges, end-points, or corners. These features are then combined by the subsequent layers in order to detect high-order features.

Distortions or shifts of the input can cause the position of salient features to vary. In addition, elementary feature detectors that are useful for one part of the image are likely to be useful for the entire image. This knowledge is applied by forcing a set of units, whose receptive fields are located at different locations in the image, to have identical weight vectors [28]. Units in a layer are organized in planes within which all the units share the same set of weights. Therefore, each feature map has a fixed feature detector that corresponds to a convolution with a trainable kernel, applied over the planes in the previous layer. Several feature maps (with different weight vectors) are used in each layer so that multiple features can be detected at each location. These feature maps form convolutional layers Ci.

Once a feature has been detected, its exact location is less important. Only its approximate position relative to other features is relevant, as absolute positions are likely to vary in the different instances of face patterns. A simple way of reducing the precision with which the positions of different features are encoded in a feature map is to reduce the spatial resolution of the feature map. Hence, each convolutional layer Ci is typically followed by a *subsampling layer*, Si, that performs local averaging and subsampling operations, reducing the resolution of the feature maps and, therefore, reducing the sensitivity of the output to shifts, distortions and variations in scale and rotation.

We precisely describe hereafter the components forming the proposed architecture. The different parameters governing the proposed architecture, i.e., the number of layers, the number of planes and their connectivity, as well as the size of the receptive fields, have been experimentally chosen. Practically, different architectures have been iteratively built, trained, and tested over large training sets. We retained the architecture that performed efficiently in terms of good detection rates and especially in terms of false alarm rejection, while still containing an acceptable number of free parameters.

Layer C1 is composed of four feature maps. Each unit in each feature map is connected to a 5×5 neighborhood into the input retina. The receptive fields of contiguous units in a feature map are centered on corresponding contiguous units in the retina. The size of the feature maps is 28×32 pixels, which prevents connections from falling off the boundary. Each feature map unit computes a weighted sum of its input by 25 (5×5) trainable coefficients, and adds a trainable bias. The implementation corresponds to a convolution by a 5×5 trainable kernel, followed by the addition of a trainable bias. Therefore, layer C1 has 104 (4×26) trainable parameters.

Layer S1 is composed of four feature maps, one for each feature map in C1. The receptive field of each unit is a 2×2 area in the previous layer's corresponding feature map. Each unit computes the average of its four inputs, multiplies it by a trainable coefficient, adds a trainable bias, and passes the results through a hyperbolic tangent function, used as an activation function. Contiguous units have nonoverlapping contiguous receptive fields. The weight sharing technique, described above, is also used here. Thus, a feature map in the subsampling layer has half the number of rows and columns of the feature maps in the previous layer. Therefore, layer S1 has four feature maps of size 14×16 and eight trainable parameters.

Successive layers of convolution and subsampling are typically alternated resulting in a "bipyramid": at each layer, the number of feature maps is increased as the resolution is decreased. This convolution/subsampling combination is inspired by Hubel and Wiesel's notions of "simple" and "complex" cells.

Layer C2 is a convolutional layer with 14 feature maps. Each unit in each feature map is connected to a 3×3 neighborhood at identical locations in a *subset* of the feature maps of S1. The implementation corresponds to convolutions by 3×3 trainable kernels, followed by the addition of trainable biases. Here, outputs of different feature maps are fused in order to help in combining different features, thus in extracting more complex information. Each of the four subsampled feature maps of S1 provides inputs to two different feature maps of C2. This results in the first eight feature maps of C2. Each of the other six feature maps of C2 takes input from one of the possible pairs of different feature maps of S1. Therefore, layer C2 has 14 feature maps of size 12×14 and 194 trainable parameters.

Layer S2 is a subsampling layer with 14 feature maps. The receptive field of each unit is a 2×2 area in the previous layer's corresponding feature map in C2, like for S1 and C1. Therefore, layer S2 has 14 feature maps of size 6×7 and 28 trainable parameters.

In layer S2, a series of hopefully disjoint and steady features of low-dimensionality are extracted and used for classification by a simple MLP. Layers N1 and N2 contain classical neural units. These layers act as a classifier, the previous ones acting as feature extractors. In layer N1, each of the 14 neurons is fully connected to all units of only one corresponding feature map of S2. The single neuron of layer N2 is fully connected to all the neurons of layer N1. The units in layers N1 and N2 perform the classical dot product between their input vector and their weight vector, to which a bias is added. This weighted sum is then passed through a hyperbolic tangent function to produce the state of the unit, in between -1.0 and 1.0. The output of neuron N2 is used to classify the input image as a nonface, if its value is negative, or as a face, if its value is positive. Layers N1 and N2 have, respectively, 602 and 15 trainable parameters.

As a comparison, in the approach of Vaillant et al. [33], input images of size 20×20 are preprocessed with a Laplacian filter, normalized and passed to an architecture composed of one convolutional layer of four feature maps of size 16×16, followed by a subsampling layer of four feature maps. The units of the four subsampled feature maps of size 8×8 are fully connected to an MLP with one hidden layer of four neurons and one output neuron. In this design, convolutional paths cannot be fused before the classification layer, and the system is not able to extract higher order features. The preprocessing stage has been designed to reduce the variability of the input patterns, with the risk of introducing a certain bias in the learning and classification processes. Because of the full connectivity of the MLP, this simpler system has, however, 1,157 free parameters to learn, most of them used in the last layer for the classification of the extracted features.

Concerning our learning strategy, all weights are computed through gradient-based learning, using a modified version of the backpropagation algorithm with momentum. The main change here is the computation of the local gradient of the backpropagated error signal with respect to the shared weights. Considering that every feature map contains in fact a single neuron with multiple instances, the local gradient for this neuron is simply the summation of the local gradients over all instances of it [15]. During training, desired network output responses are set to -1.0 for nonfaces and to +1.0 for faces.

Since all weights are learned, the system can be seen as synthesizing its own set of problem specific feature extractors. The proposed network has only 951 trainable parameters, despite the 124,741 connections it uses. Weight sharing offers the advantage of reducing the number of parameters, thus reducing the *capacity* of the machine, allowing a better generalization and, therefore, reducing the gap between training and testing errors [15]. Local receptive fields, weight sharing, and subsampling provide many advantages to solve two important problems at the same time: the problem of robustness and the problem of good generalization, which is critical given the impossibility of capturing all the possible variations of the face pattern in a training set of finite size.

The proposed topology has another interesting property. In most image-based approaches ([6], [16], [22], [23], [24]), in order to search for faces at a given scale, the network must be replicated (or scanned) at all locations in the input image. All these approaches follow the preprocessing stage proposed by Sung and Poggio [16], which consists in locally normalizing the face window pixels before feeding the network retina for classification, therefore treating every neighbor *separately* and overlapping face windows. In our approach, no preprocessing local to subwindows is performed and arrays of raw pixels are fed directly to the retina. Since each layer essentially performs parallel convolutions of small size kernels, for each convolution, a very large part of the computation is in common between two neighboring retina windows in the input images. This redundancy is naturally eliminated by performing the convolutions corresponding to each layer on the entire input image at once. The overall computation consists in a pipeline of convolutions and nonlinear transformations over the entire images, as shown in Fig. 2. At every level of the pipeline, full image convolutions and nonlinear transformations of small size kernels can be easily performed in parallel. These standard operations can be implemented directly on standard image-processing hardware boards, allowing low-cost near real-time applications.

2.2 Training Methodology

We built a training set by manually cropping $3,702$ highly variable face areas in a large collection of images collected from various sources over the Internet or from scanned images of newspapers. The collected images are chosen to effectively capture the variability and the richness of natural data in order to train our system for operating in uncontrolled environments.

Using manually labeled eye and mouth positions, face images are cropped and normalized to the size of 32×36 pixels, after derotation and rescaling. The aim of

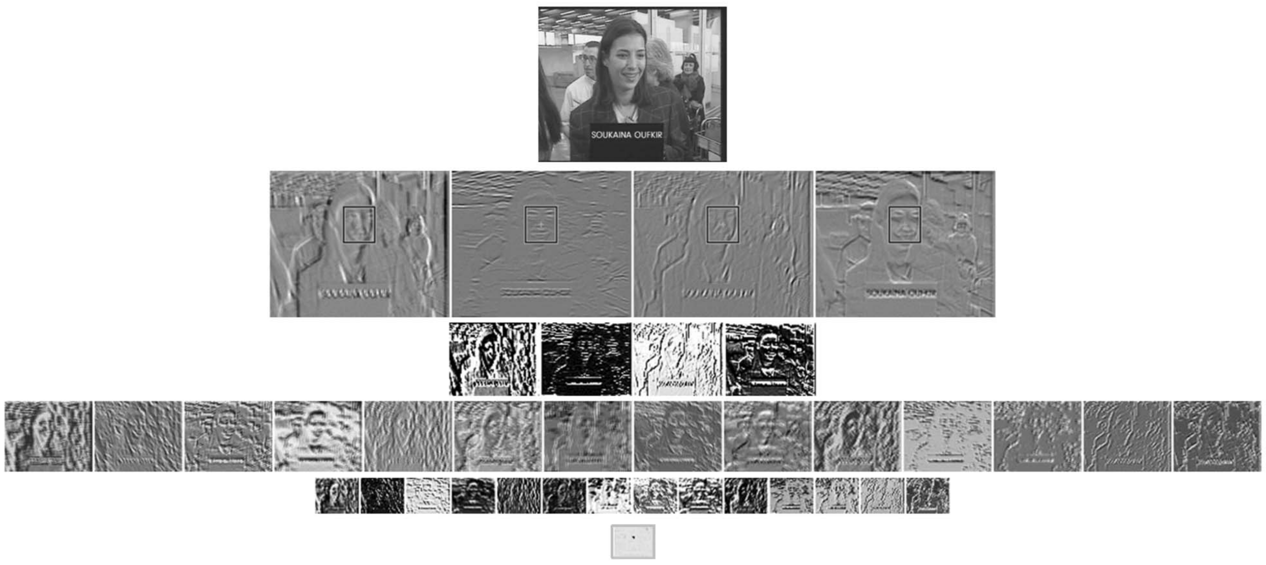


Fig. 2. The images produced at each stage of the pipeline, for a given scale. For illustration purposes, the retina window is displayed over the face in the images of the second row. The image at the bottom contains the outputs of the neural architecture.

this procedure is to position the two eyes approximately at the same locations inside the retina, while the distance from the mouth point to the eye-line is used to roughly preserve the original aspect ratio of the faces. More precisely, the eye-line is roughly 16 pixels wide while its distance from the mouth is close to 18 pixels.

Most of the image-based approaches in the literature ([6], [16], [22], [23], [24]) use a face window of about 20×20 pixels, reported as being the smallest window one can use without loosing critical information. Usually, this window contains the very central part of the face, excluding its borders and any parts of background, by masking border pixels. We have chosen approximately the same size for the central part of the face but centered in a 32×36 pixel window where borders of the face are included. The reason for this is twofold. First, the network is fed with some additional information about the face shape, which will help in reducing the number of false alarms produced that are more likely to appear when only the very central part of the face is considered. Second, some border effects that may arise in the convolutions are canceled. The rough manual labeling of the facial points introduces a physical error in the cropping process, which in turn affects the positions of the training examples inside the retina. As mentioned earlier, our system is quite robust against variations in scale and position, and small errors in the manual labeling reinforce this ability by providing examples that are not precisely normalized. Moreover, the task of gathering training examples is less tedious if no strict alignment of the face patterns is required.

No intensity normalization is performed on the cropped faces, such as overall brightness correction and histogram equalization that are applied in [6], [16], [22], [23], [24]. In order to create more examples and to enhance the tolerance to small inplane rotations and variations in intensity, a series of transformations are applied to the initial set of face examples. At first, some examples are mirrored and then all the examples are rotated by ± 20 degrees. As a final step, some of the examples are smoothed, while contrast reduction is applied to others. The last two transformations help in teaching the system how to cope with situations where the retina is fed with a weak, oversmoothed, or poorly-contrasted signal. In general, highly variable examples are essential for the performance of the system because neither histogram equalization nor lighting correction is applied to the signal before feeding the network. Finally, the training set reached the number of $25,212$ face patterns, partially occluded (glasses, hair, etc...), unequally lighted, turned up to ± 60 degrees, rotated up to ± 20 degrees and with average intensity values varying from dark to light. Some examples of the collected training patterns are shown in the three first rows of Fig. 3.

Collecting a representative set of nonfaces is more difficult. Practically, any randomly cropped image can serve as a nonface example; the space of nonfaces can be seen as "the rest of the world." Because of the impossibility of training the network with every possible nonface example, nonface patterns were collected via a new iterative bootstrapping procedure, inspired by the algorithm proposed by Sung and Poggio in [16]. This method consists in iteratively retraining the system with an updated training set containing false alarms produced after face detection has been performed on a set of scenery images that do not contain faces. In the proposed approach, we improved this strategy in some points. Before proceeding with the bootstrapping, an initial training set of $6,422$ nonface patterns is built by selectively cropping images. Most of these images contain parts of faces as we noticed that this kind of image is a serious source of false alarms. The training-bootstrapping algorithm that we implemented proceeds as follows:

1. Create a validation set of 400 face patterns and 400 nonface patterns randomly extracted and excluded from the initial training set. This set is used to select the best performing weight configuration in Steps 3 and 8.
2. Set $BootsIter = 0$, $ThrFa = 0.8$.
3. Train the network for 60 learning epochs. Use an equal number of positive and negative examples in each epoch.
   Set $BootsIter = BootsIter + 1$.

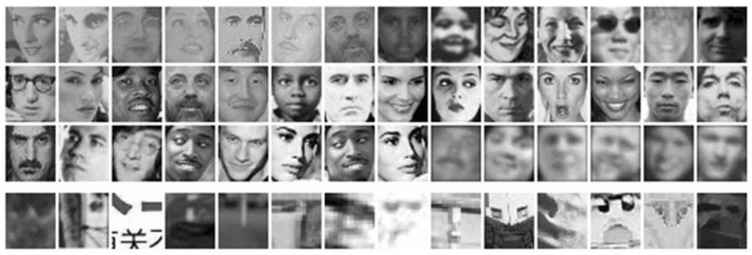


Fig. 3. Some patterns used for training. The first three rows present some highly variable face patterns. The last row contains some examples of nonface patterns produced by the bootstrapping procedure.

4. Gather false alarms from a set of 692 scenery images (containing no face) with network answers above $ThrFa$. Collect a maximum number of 5,000 new examples.
5. Concatenate the newly created examples to the nonface training set.
6. If $ThrFa \geq 0.2$, set $ThrFa = ThrFa - 0.2$.
7. If $BootsIter < 6$, go to Step 3.
8. Train the network for 60 more learning epochs using the full training set and exit.

In Step 1, a validation set is built and used for testing the generalization ability of the network during learning and, finally, selecting the weight configuration that performs best on it. This validation set is kept constant throughout all the bootstrapping iterations, in contrast with the training set that is updated.

In Step 3, stochastic learning using backpropagation is performed with the addition of a momentum term for neurons belonging to the N1 and N2 layers. At each bootstrapping iteration, the network is trained for 60 epochs using the updated training set, which contains an increasing number of negative examples. For every learning epoch, an equal number of randomly selected examples from both classes are presented to the network, so that no bias is introduced.

The selection of the new patterns that are to be added to the set of nonface examples is carried out by Step 4. The false alarms produced in this step force the network, in the next iteration, to revise its modeling of the face class and to refine the decision boundaries between face and nonface patterns. At each iteration, the false alarms, giving network answers greater than $ThrFa$ are selected. In the first iteration, false positives inside the current face class boundaries (i.e., giving network answers greater than $ThrFa = 0.8$) are selected. Then, as the network generalizes from these examples, $ThrFa$ is gradually reduced by 0.2 in each subsequent iteration, until reaching 0.0. The learning process is stopped after six iterations, when the number of false alarms is low and remains roughly constant. This procedure helps in correcting problems arising in the original algorithm proposed by Sung and Poggio where false alarms were grabbed regardless of the strength of the network answers. By doing so, a very large number of false alarms were eventually grabbed during the first iterations, with relatively low network answers. This high number of possibly redundant patterns included in the training set significantly slows down the learning speed as the interesting negative examples with high scores are presented fewer times to the network. With a selection of the strongest negative examples, our scheme provides faster training, especially in the first iterations where the network is still a weak classifier. As the network updates quickly its parameters from these early negative examples, $ThrFa$ can be then reduced until reaching 0.0, in a safer way, gathering a balanced set of false alarms. Finally, $18,665$ false alarms are grabbed during this bootstrapping process. Therefore, the final set of nonface examples contains $25,087$ examples. Some of them are shown in the last row of Fig. 3.

Some other trials have also been conducted in order to further accelerate the learning scheme by pruning the training set after each iteration, and excluding some of the negative examples correctly classified with strong negative responses and, therefore, distant from the decision boundaries. Our preliminary experiments showed that such a pruning scheme may be efficient, but further experiments are still to be conducted to reach equivalent training results.

The performance of the proposed training-bootstrapping procedure is depicted in Fig. 4. In particular, one can observe how the network was boosted in terms of false alarm rejection. Fig. 4a presents the *volume* of the grabbed false alarms with respect to the training-bootstrapping iterations. This metric characterizes the *strength* of the false alarms. It corresponds to the sum of the positive network answers for all the nonface examples grabbed at each iteration. This curve shows that the process grabs fewer and fewer *strong* false alarms as learning takes place. One can notice that the first iteration produces very strong false alarms, as expected for the first run of the network over the scenery images. During the following iterations, the network quickly learns how not to produce strong false alarms, as illustrated by the sharp decrease of the volume in iteration 3. Thereafter, the behavior remains roughly constant, which indicates that the bootstrapping procedure can safely terminate. Fig. 4b shows the evolution of the average output value of the network for the face and grabbed nonface examples of the training set. It can be noticed that the average output value for the face patterns increases after the first iteration and stays roughly constant, around 0.85, after the second iteration, while an increasing number of nonface patterns are presented to the network. It also illustrates how the average output value to nonface patterns is significantly decreasing as learning progresses, showing how the class boundaries are adapting.

2.3 Face Localization

Fig. 5 depicts the process of face localization in a gray-scale image, containing three faces of different sizes. In order to detect faces of different sizes, the input image is repeatedly subsampled by a factor of 1.2, resulting in a pyramid of images (Step 1). Each image of the pyramid is then filtered by our convolutional network *CFF* (Step 2). In [6], [16], [22], [23], neural filters are applied at every pixel of each image of the pyramid, after some operations of lighting correction. Thus,

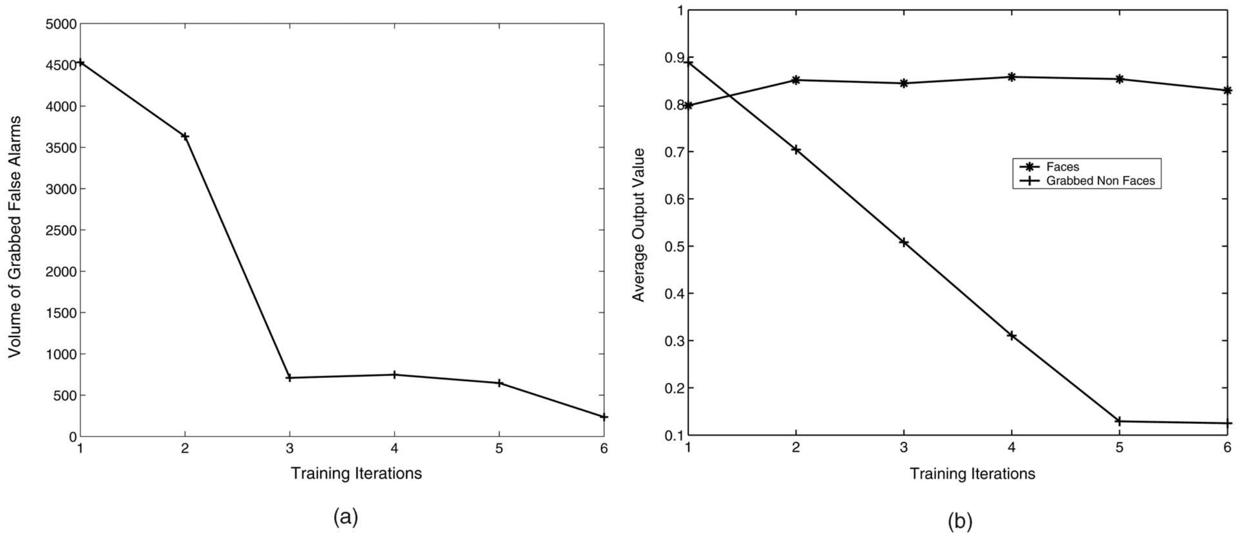


Fig. 4. The evolution of our training-bootstrapping process. (a) The volume of grabbed false alarms quickly decreases during the iterative learning. (b) The average output value for face patterns increases after the first iteration and then stays roughly constant, around 0.85, while an increasing number of nonface patterns are presented to the network. Meanwhile, the average output value for grabbed false alarms used as nonface patterns is quickly decreasing.

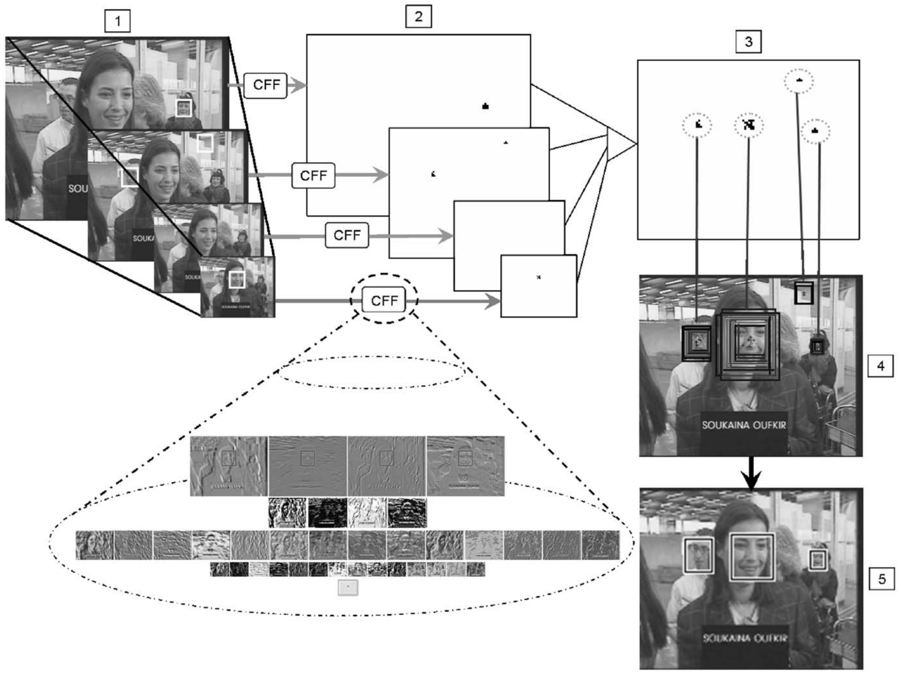


Fig. 5. The different steps of the process of face localization. (1) Creation of a multiscale pyramid from the original image. (2) Convolution of each image of the pyramid by the neural filter *CFF*. (3) Projection of face candidates to the original scale and fusion of overlapping face candidates. (4) Application of the neural filter in a fine pyramid centered at each face candidate position. (5) Classification of each face candidate according to the volume of positive answers in the corresponding fine pyramid.

every image subwindow has to be cropped, processed for lighting correction and histogram equalization before being passed to the retina of the neural classifier. In our approach, as mentioned earlier, each image of the pyramid is entirely convolved at once by the network. For each image of the pyramid, an image containing the network results is obtained. Because of the successive convolutions and subsampling operations, this image is approximately four times smaller than the original one. This fast procedure may be seen as corresponding to the application of the network retina at every location of the input image with a step of four pixels in both axis directions, without computational redundancy.

After processing by this detection pipeline, face candidates (pixels with positive values in the result image) in each scale are mapped back to the input image scale (Step 3). They are then grouped according to their proximity in image and scale spaces. Each group of face candidates is fused in a representative face whose center and size are computed as the centroids of the centers and sizes of the grouped faces, weighted by their individual network responses. After applying this grouping algorithm, the set of remaining representative face candidates serve as a basis for the next stage of the algorithm, in charge of fine face localization and eventually false alarm dismissal.

To do so, a local search procedure is performed in an area around each face candidate center in image scale-space (Step 4). A reduced search space centered at the face candidate position is defined in image scale-space for precise localization of the face candidate. It corresponds to a small pyramid centered at the face candidate center position covering 10 equally distant scales varying from 0.8 to 1.5 times the scale of the face candidate. For every scale, the presence of a face is evaluated on a rescaled grid of 16×16 pixels around the corresponding face candidate center position. We observed that true faces usually give a significant number of high positive responses in consecutive scales, which is not often the case for nonfaces. In the preliminary approach that we presented in [37], we proposed a solution in which the number of positive answers nok in the local pyramid was considered in order to take the classification decision. A face candidate was classified as face if nok was greater than a certain number. In order to discriminate true faces from false alarms, it resulted more efficient to take into account both number and values of positive answers. We therefore consider the *volume* of positive answers (the sum of positive answer values) in the local pyramid in order to take the classification decision. Based on the experiments described in the next section, a face candidate is classified as face if its corresponding volume is greater than a given threshold $ThrVol$ (Step 5). The bottom-right image of Fig. 5 shows the positions and sizes of the faces detected after local search. One can notice that the false alarm (up right in the image), previously detected in Step 4, with a low volume after local search, has been removed using the volume threshold criterion.

3 Experimental Results

In this section, we aim at presenting the accuracy and the robustness of the Convolutional Face Finder. First, we analyze the sensitivity of the method with respect to the degrees of variability of the face patterns. Finally, we examine the performance of our face detector on different test sets of images, including the *CMU Test Set* [23], used for comparison with other state-of-the-art techniques.

3.1 Sensitivity Analysis

Experimental results reporting detection and false alarm rates are usually presented, but little information is given about the influence of the face pattern variations on the classification results. Rowley et al. [23] present an analysis of the network output variations with respect to localized noise affecting the face patterns. Féraud et al. [6] analyze the sensitivity of their face detector with respect to shift of the face patterns. We propose a deeper study, by considering the sensitivity analysis of our method with respect to shift, rotation in image plane, blurring, contrast modification, and addition of Gaussian white noise. We also consider tolerance to variations in facial pose and expression.

We selected a set of 20 images, each of them having been cropped around a single face. They can be seen in the first two rows of Fig. 6. One can notice that these faces are approximately frontal and nonrotated. We conducted the sensitivity analysis by applying some transformations to the

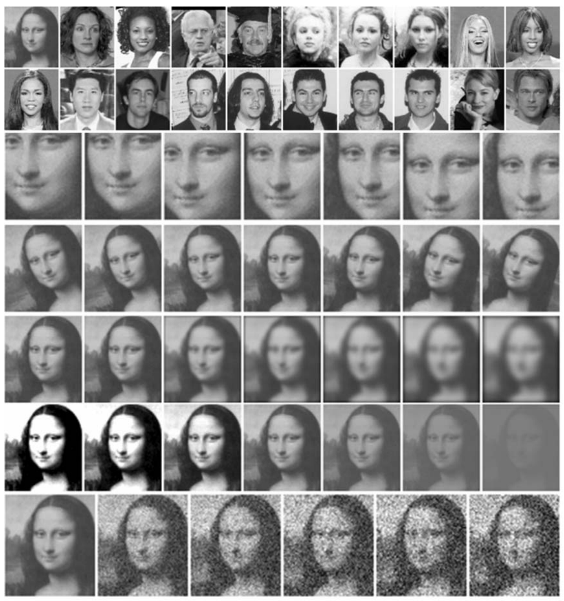


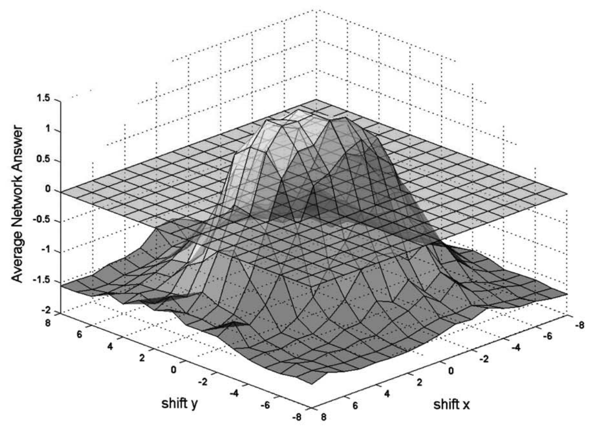
Fig. 6. Original images and some transformed images used for sensitivity analysis. From top to bottom, transformations are shift, rotation, blurring, contrast modification, and addition of Gaussian white noise.

input images and studying their influence on the average output values produced by our face detector on the set of faces. From the third to the seventh row of Fig. 6, we present some examples of these transformations for one of the images of the set. The transformations are shift, rotation in image plane, blurring, contrast modification, and addition of Gaussian white noise.

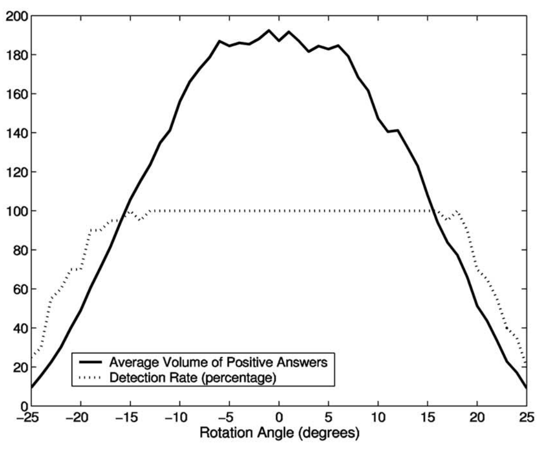
For analyzing the sensitivity to shift, we first rescaled each image so that the contained face has the size of the retina, i.e., 32×36. Then, we cropped image areas around the face location in each image, by successively translating the face center by an amount varying from -8 to +8 pixels in both x and y directions. We fed the retina of the neural network by each image and recorded the corresponding output value. In Fig. 7a, we display the average network output among all cropped faces with respect to the shift vector. A horizontal plane shows the zero value on the z-axis, the axis of the average network outputs. One can notice that the average network output is a function which has a peak for the centered face and decreases monotonically as the shift increases. This property illustrates the behavior of our face detector with respect to shift of the face pattern in the retina.

For studying the sensitivity of the proposed face detector to the remaining selected transformations that affect the face appearance, we apply the complete *CFF* system to each transformed image of the set. For various transformation parameter values, we record the detection rate over the set of images, the number of eventual false alarms, and for all detected true faces, the average volume of positive answers ($AVPA$) after fine search.

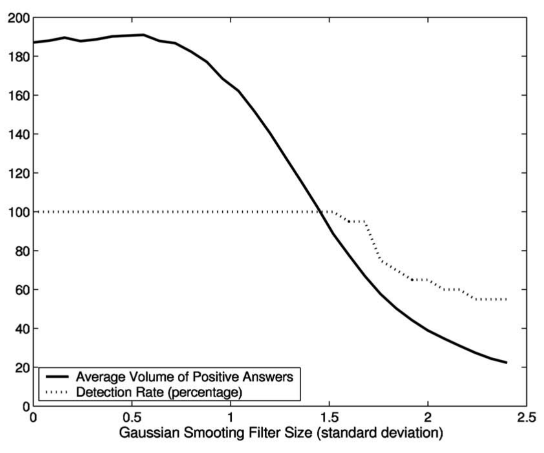
As mentioned earlier, we generated some training examples, in order to give to our system the ability of detecting faces rotated up to ± 20 degrees. For readers interested in fully rotation invariant face detection, an interesting approach has been presented by Rowley et al. in [38]. In order to analyze the robustness of our approach, we successively rotate the images by an angle varying from -25 to +25 degrees with a



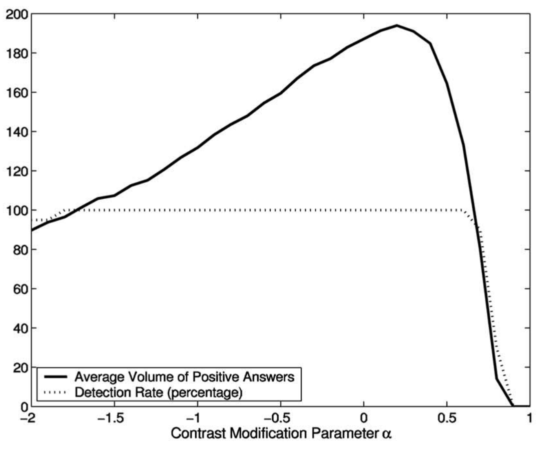
(a)



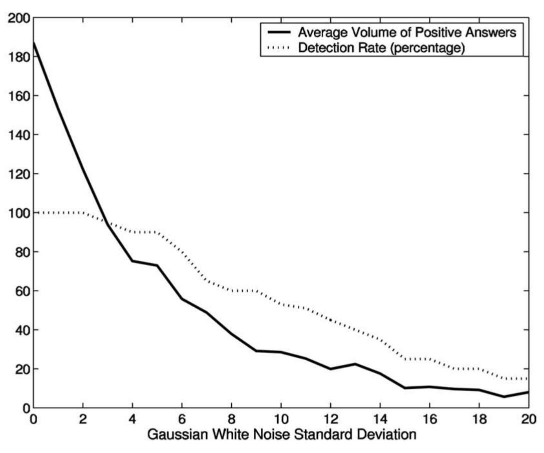
(b)



(c)



(c)



(d)

Fig. 7. Sensitivity analysis of the proposed face detector, with respect to (a) shift, (b) rotation, (c) blurring, (d) contrast modification, and (e) addition of Gaussian white noise.

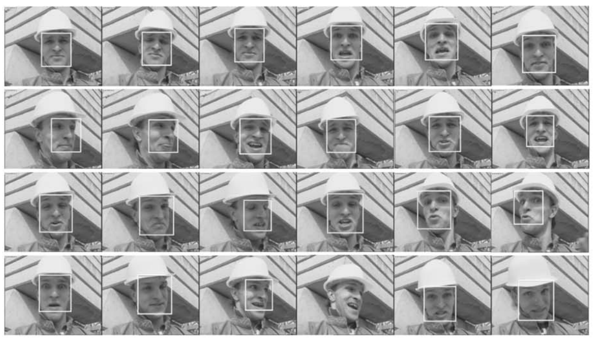
step of one degree. Each of these rotated images is then processed by the face detector and the resulting $AVPA$ is recorded over the set of images for each rotation angle. Fig. 7b shows the detection rate and the $AVPA$ values with respect to angles of rotation. It can be observed that the detection rate is close to 100 percent for rotation angles comprised in between -20 and $+20$ degrees and decreases significantly for larger rotations. One false alarm has been detected with a small volume of 1.8. The curve corresponding to $AVPA$ values has a bell shape and decreases slowly from the maximum value, corresponding to the nonrotated images. The face detector produces $AVPA$ values greater than 20.0 for faces rotated up to ± 20 degrees.

In order to analyze the influence of blurring on the performance of the face detector, a Gaussian smoothing filter with standard deviation ranging from 0 to 2.5 is applied to each image. Fig. 7c presents the detection rate and the $AVPA$ values with respect to the standard deviation of the Gaussian filter. One can observe that the detection rate is still 100 percent for a standard deviation of 1.5 and decreases slowly after. No false alarm has been produced. The face detector returns *AVPA* values, which decrease gently as the image is increasingly blurred. Note that the *AVPA* value is still greater than 20.0 for a standard deviation of 2.0.

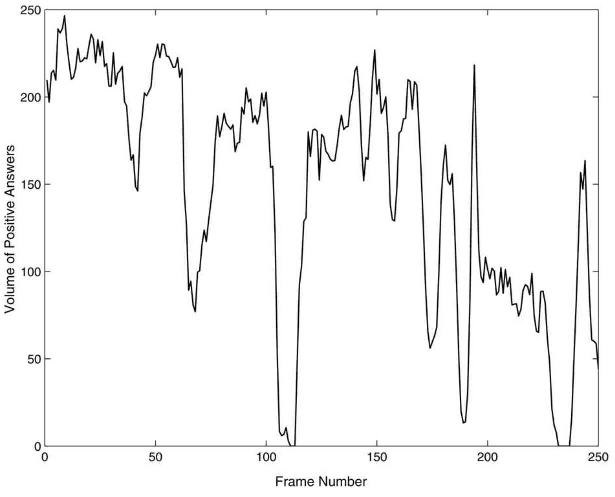
Tolerance to contrast variation is studied by applying contrast modification. For each image, each pixel intensity I_p is changed according to $I_p = \alpha I_m + (1-\alpha)I_p$, where I_m is the mean intensity of the image and α is a parameter varying from -2.0 to 1.0. Contrast enhancement is performed when α is negative and contrast reduction is performed otherwise. Fig. 7d shows the detection rate and the *AVPA* values with respect to the parameter α. It can be seen that the *AVPA* values decrease very slowly for increasing contrast enhancement ($\alpha < 0$) and decrease faster as we reach noncontrasted ($\alpha > 0$), and finally the image of constant gray level ($\alpha = 1$), as shown in the sixth row of Fig. 6. Two false alarms have been produced with *AVPA* values below 2.9. The detection rate stays close to 100 percent until the contrast is reduced by 75 percent. Note that *AVPA* is still high (more than 40.0) for this percentage of contrast reduction.

To study the influence of noise, a Gaussian white noise of standard deviation ranging from 0 to 0.025 is applied to each image. Fig. 7e presents the detection rate and the *AVPA* values with respect to the standard deviation of the Gaussian white noise. Two false alarms have been detected with *AVPA* values lower than 3.0, for high levels of noise. It can be seen that detection rate and *AVPA* values decrease slowly as the noise increases, which illustrates the robustness of our face detector in handling very noisy face patterns.

Another interesting aspect is tolerance to changes in facial expressions and poses. To illustrate the tolerance of our detector to pose variations, we processed the well-known Foreman MPEG sequence. This sequence contains 250 frames of size 352×288, where the speaker face undergoes many changes in expression and pose. Fig. 8a shows some of the processed frames while Fig. 8b shows the volume of positive answers for the true face detected in each frame. The curve evolves as pose and facial expressions changes. One can noticed that the output of the face detector is greater than 20.0 for most of the frames, except for three subsequences of frames (around frame 110, 188, and 231) where the speaker's pose is close to full profile. In these cases, the face was missed or detected with a low and therefore unreliable output. We obtained an overall detection rate of 96.8 percent (eight missed faces) with six false alarms of volume smaller than 10.2. By selecting candidate faces with volumes greater than 20.0, we obtained a detection rate of 94.0 percent (15 missed faces) with no false alarm.



(a)



(b)

Fig. 8. Performance of the face detector applied to the Foreman MPEG sequence. (a) Some processed frames. (b) The volume of positive answers for each true face in each frame.

3.2 Results on Test Data Sets

The level of performance of our method has been evaluated using three test sets of images, collected by Rowley et al. [23], Sung and Poggio [16], and ourselves. Most of the images in these data sets have complex backgrounds with faces taking up only a variable but small amount of the total image area. Faces in these data sets present a large variability in size, illumination, facial expression, pose, and may be partially occluded.

We use the *CMU* test set [23], which is, so far, the most widely used data set in the literature. It consists of 130 images with a total of 507 faces. This data set includes 23 images of the second data set used by Sung and Poggio [16], referred to as *MIT*. A subset of the entire *CMU* test set, referred to as *CMU-125*, has been used by many researchers. It excludes hand-drawn and cartoon faces and contains 483 faces. A subset of the *MIT* set, excluding three images containing hand-drawn and cartoon faces is referred to as *MIT-20*.

To test our method further, we collected two other sets of images. The first one, referred to as *Web* test set, is a randomly chosen subset of 215 images from the large set of images that have been submitted to the interactive demonstration of the proposed system, available on the Web at www.csd.uoc.gr/~cgarcia/FaceDetectDemo.html, allowing anyone to submit images and to directly see the detection results. This test set contains a great variety of examples and it has not been biased by a specific selection of images. It contains 499 faces. The second test set that we collected, referred to as *Cinema*, consists of 162 images extracted from various movies, containing 276 faces. This test set is considered as very challenging, as the images have been specifically chosen to test the limits of our face detector. It contains a high number of faces with extreme facial expressions as well as faces, which are partially occluded or heavily shadowed, surrounded by very complex backgrounds.

Only a few papers, including [17], address the definition of *what is a correctly detected face* [2]. In our experiments, a detected face is considered as valid if the face window is not

20 percent bigger that the real face area and contains both eyes and mouth. In some papers, reported results of selected methods on the face databases are difficult to interpret. For a given approach, some results correspond to the maximal detection rate with a high number of false alarms and others to a lower detection rate with a smaller number of false alarms. Indeed, most face detectors can adjust parameters (usually a threshold) depending on how conservative one needs the system to be. These adjustments influence detection and false alarm rates. The detection rate is the ratio between the number of successful detections and the number of labeled faces in the test set. The false alarm rate is the ratio between the number of false positive detection and the number of scanned windows. This can be reported in terms of an ROC curve to show the detection rate versus the false alarm rate for various values of the threshold. Fig. 9a reports the ROC curves that we obtained for the *CMU*, the *Web*, and the *Cinema* test sets. Each point on a curve corresponds to a given threshold $ThrVol$ applied to the *volume* of positive answers in order to classify an image area into face or nonface. One can notice that these curves have similar shapes. For small threshold values (extreme right part of the curves), high detection and false alarm rates are obtained. As the threshold value increases (toward the left), the detection rate decreases slowly while the false alarm rate decreases more quickly. The extreme left points of the curves correspond to false alarm rates of zero. In that case, only true faces are detected. Note that we obtain quite high detection rates with no false alarm, which are 88.8 percent, 90.5 percent, and 80.4 percent for the *CMU*, the *Web*, and the *Cinema* test sets, respectively. On the other hand, the maximum detection rates are 93.3 percent with 197 false alarms, 98.0 percent with 108 false alarms, and 95.3 percent with 104 false alarms, for the *CMU*, the *Web*, and the *Cinema* test sets, respectively. These results are reported in Table 1. Figs. 9b and 9c show the detection and false alarm rates versus $ThrVol$ for the different test sets. It can be noticed that false alarm rates decrease much faster than detection rates, that decrease linearly, for increasing values of $ThrVol$. For all three test sets, false alarm rates have a very similar behavior and values of $ThrVol$ greater than 20.0 give very low false alarm rate.

Table 1 shows the results obtained by our face detector *CFF* on these three test sets, for different points on the ROC curves. This table intends to show how the choice of $ThrVol$ influences detection and false alarm rates for each test set. It can be observed that, for instance, a value of $ThrVol = 17.0$, derived after some trials performed on another subset of images submitted to our online demo Web site, gives high detection rate with a very low number of false alarms for all three test sets. This suggests that the proposed system processes most images in a stable and efficient way. The best results are obtained for the *Web* test set, on images which cannot be considered as biased by a specific selection. This set contains a large variety of images covering most of the face appearance classes. Results obtained on the *Cinema* test set are still very good although this test set is more challenging, as the images have been chosen for testing the limits of our detector. One can notice that, for zero false alarms, the detection rate for this test is the lowest, i.e., 80.4 percent. It appeared that a very strong false alarm, in a background area of an image of this test set, was locally very similar to a face and, thus, required a high $ThrVol$ value to be rejected, which in turn rejected true faces and lowered the overall detection rate. Results concerning the *CMU* test set are very similar to the ones obtained on the *Cinema* test set. However, the false alarms for the *CMU* test set are rejected for a lower threshold, giving a high detection rate for zero false alarms.

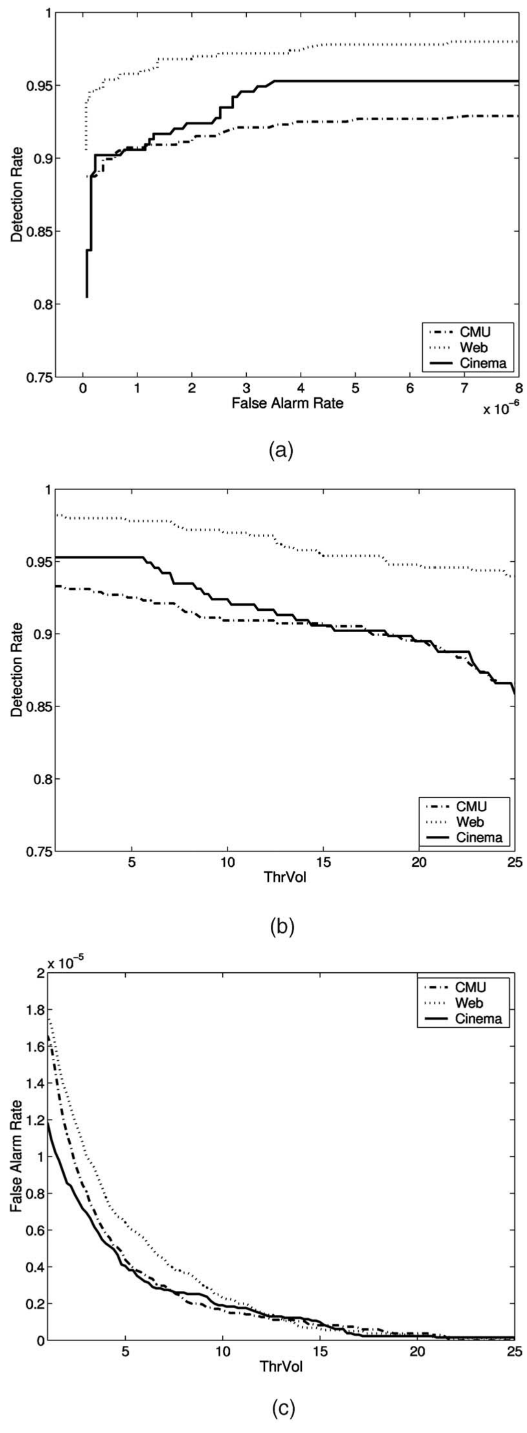


Fig. 9. (a) The ROC curves obtained for the *CMU*, the *Web*, and the *Cinema* test sets. Every point on a curve correspond to a given threshold value $ThrVol$ of the *volume* of positive answers produced by our face detector. (b) and (c), respectively, show the detection and false alarm rates versus $ThrVol$.

In order to compare our approach with other methods, we consider reported results on the *CMU* and *MIT* test sets. Most previous published results on these test sets have only included a single operating regime, which corresponds to a single point on the ROC curve. As mentioned earlier, these

TABLE 1
Results of *CFF* on the *CMU*, *WEB*, and *CINEMA* Test Sets for Different Points on the ROC Curves

Test set	$ThrVol$	Detected Faces	Detection Rate	False Alarms
CMU	1.4	473	93.3%	197
	17.0	458	90.3%	8
	21.6	450	88.8%	0
Web	4.2	489	98.0%	108
	17.0	476	95.4%	6
	37.0	452	90.5%	0
Cinema	2.6	263	95.3%	104
	17.0	249	90.2%	3
	41.8	222	80.4%	0

results are difficult to interpret. For a given approach, some results correspond to high detection rate with a high number of false alarms and others to a lower detection rate but with a smaller number of false alarms. To make a comparison with our detector easier, we have listed our detection rate for the same false alarm numbers reported by other systems, as in Viola and Jones [26] and Li et al. [27].

Table 2 lists the detection rates for various numbers of false detections for our system as well as for other published systems as reported in [26], [27]. It can be observed that our method compares favorably with the others, especially for low numbers of false alarms. This shows that our approach separates face and nonface space in a robust and balanced way. For larger number of false alarms, our results are equivalent to the ones reported for the other methods. It suggests that all these detectors reach very similar maximal detection limits.

As mentioned earlier, most of the state-of-the-art published methods report only a single operating regime. In Table 3, we summarize these published results on the *CMU* and *MIT* test sets. Our results are given for $ThrVol = 17.0$, which provides balanced detection and false alarm rates on different test sets. It can be noticed that these results are stable, with detection rates greater than 90 percent in all test sets, with a maximum of eight false alarms. For the entire *CMU* test set, we report here, to our knowledge, the best published results with a detection rate of 90.3 percent for only eight false alarms. We also report, as seen in Table 1, a detection rate of 88.8 percent with no false alarm. This detection rate is still equivalent to or higher than the ones reported by Viola and

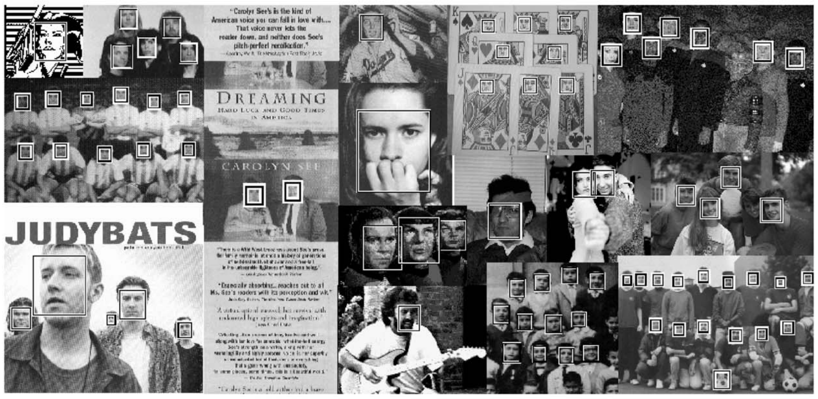
TABLE 2
Comparison of Selected Methods for Various Numbers of False Alarms on the *CMU* Test Set

Face detector	False alarms				
	0	10	31	65	167
Rowley *et al.* [23]	-	83.2%	86.0%	-	90.1%
Schneiderman and Kanade [20]	-	-	-	94.4%	
Li *et al.*[27]	-	83.6%	90.2%	-	-
Viola and Jones [26]	-	76.1%	88.4%	92.0%	93.9%
Convolutional Face Finder	88.8%	90.5%	91.5%	92.3%	93.1%

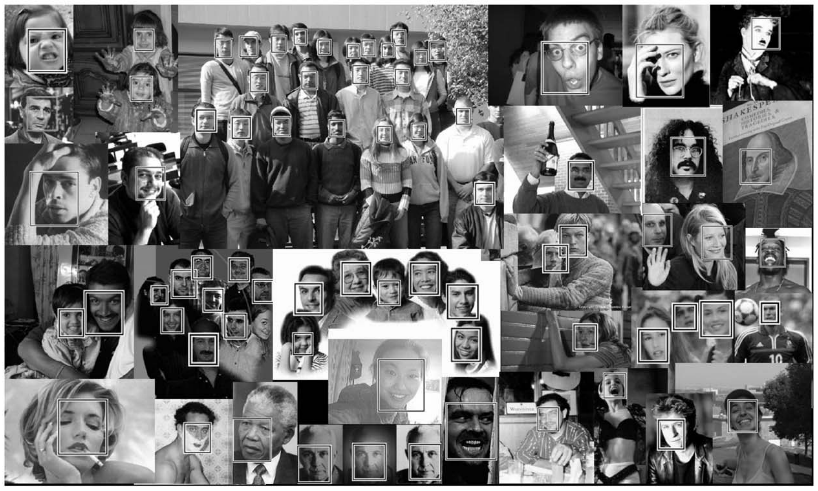
TABLE 3
Results Reported in Terms of Percentage of Good Detection/Number of False Alarms, on the *CMU* and *MIT* Test Sets

Face Detector	CMU	CMU-125	MIT	MIT-20
Colmenarez and Huang [18]	93.9%/8122			
Féraud *et al.* [6]	86.0%/8			
Yang *et al.* [17]		93.6%/74		91.5%/1
Osuna *et al.* [22]			74.2%/20	
Roth *et al.* [24]		94.8%/78		94.1%/3
Rowley *et al.* [23]	86.2%/23		84.5%/8	
Schneiderman and Kanade [20]		94.4%/65		
Sung and Poggio [16]			79.9%/5	
Viola and Jones [26], [39]	88.4%/31		77.8%/5	
Li *et al.* [27]	90.2%/31			
Convolutional Face Finder	90.3%/8	90.5%/8	90.1%/7	90.2%/5

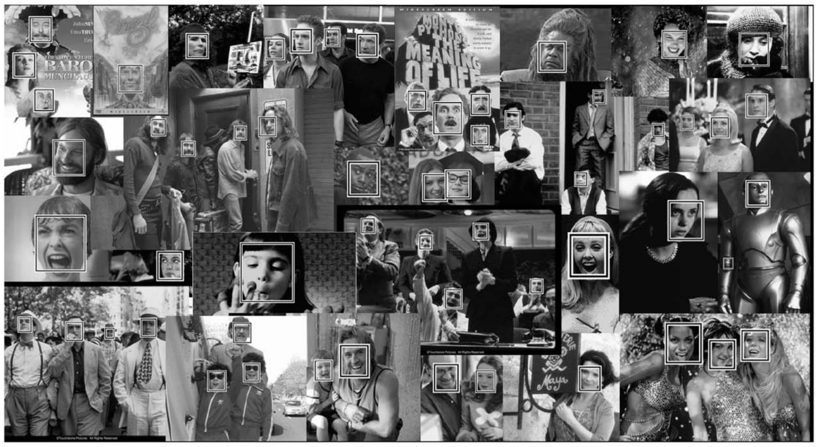
The results obtained with our approach are reported for all test sets for $ThrVol = 17.0$.



(a)



(b)



(c)

Fig. 10. Some results obtained on the (a) *CMU*, (b) the *Web*, and (c) the *Cinema* test sets.

Jones, Rowley et al., and Féraud et al., but with false alarms. Although detection rates with no false alarms are not classically reported, we found this information interesting, showing the discriminative power of the proposed approach. For the *CMU-125* test set, we obtain a detection rate slightly inferior than the ones reported by other methods which do not handle cartoon-like faces, but we obtain a much smaller number of false alarms. For the *MIT* test set, we report equivalent or better results than the ones reported for the other systems.

Regarding processing speed, our current implementation, running at four frames per second for 384×288 pixel images on a conventional 1.6 GHz Intel Pentium IV, is approximately four times faster than the fast version of Rowley et al., referred to as *System 17*, two times slower than the method of Li et al., and roughly eight times slower than the method of Viola and Jones. However, our approach provides robust detection results, especially if low false alarm rate is expected. On the *CMU* test set, we obtained a detection rate of 90.3 percent with eight false alarms, which compares favorably with the detection rates of 76.9 percent with eight false alarms for the Rowley's System 17 [23], 76.1 percent with 10 false alarms for Viola and Jones, and 83.6 percent with 10 false alarms for Li et al.

In Fig. 10, we present some results of the proposed face detection scheme on the *CMU*, *Web*, and *Cinema* test sets, for $ThrVol = 17.0$. These examples include complex images with multiple highly variable faces of different sizes. False alarms and false dismissals are presented as well. The images contained in the *Web* and *Cinema* test sets are available online in our face detector demo web page. Ground truth of face localizations is also provided, hoping that these data will serve the face detection community for future evaluations and comparisons.

4 Conclusion

We have presented a framework for robust face detection based on an efficient convolutional neural network architecture, designed in order to detect highly variable face patterns, rotated up to ± 20 degrees in image plane and turned up to ± 60 degrees, in complex real world images. This system automatically synthesizes simple feature extractors and classifiers from a very large training set of face and nonface patterns, without making any assumptions concerning the features to extract or the areas of the face pattern to analyze. Once trained, the system acts like a pipeline of simple convolution and subsampling modules that treat the raw input image as a whole, without requiring any local preprocessing in the input image.

The robustness of our system to varying poses and facial expressions as well as lighting variations and noise was evaluated by considering its sensitivity with respect to various transformations of the face patterns and using real sets of difficult images. Experiments have shown high detection rates with a particularly low number of false alarms, on difficult test sets, without requiring the use of multiple networks for handling difficult cases. These results illustrate the high discriminant power of our convolutional architecture where local receptive fields and weight sharing provide appropriate high level feature extraction and allow efficient generalization. Our results also suggest that performing local intensity normalization before classifying image areas is not mandatory to build an efficient face detection system.

From a more general perspective, the full scheme of the *CFF* method can be a good candidate to a large number of computer vision detection or recognition tasks, where patterns to detect undergo distortions difficult to model empirically. As a direct extension of this work, we are currently considering the detection of full profile faces via the proposed architecture. Current experiments dealing with full-profile faces, using additional feature maps are very encouraging.

Acknowledgments

The authors would like to thank all visitors of their face detector demo Web site, for their interest in their work and for kindly providing test images. This work has been carried out while the authors were with the Computer Science Department of the University of Crete, PO Box 1470, 71110 Heraklion, Greece.

References

[1] M. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 34-58, Jan. 2002.

[2] E. Hjelmås and B.K. Low, "Face Detection: A Survey," *Computer Vision and Image Understanding,* vol. 83, pp. 236-274, 2001.

[3] V. Govindaraju, "Locating Human Faces in Photographs," *Int'l J. Computer Vision,* vol. 19, no. 2, pp. 129-146, 1996.

[4] G. Yang and T.S. Huang, "Human Face Detection in Complex Background," *Pattern Recognition,* vol. 27, no. 1, pp. 53-63, 1994.

[5] C. Garcia and G. Tziritas, "Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis," *IEEE Trans. Multimedia,* vol. 1, no. 3, pp. 264-277, 1999.

[6] R. Féraud, O. Bernier, J. Viallet, and M. Collobert, "A Fast and Accurate Face Detector Based on Neural Networks," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 1, pp. 42-53, Jan. 2002.

[7] B. Low and M. Ibrahim, "A Fast and Accurate Algorithm for Facial Feature Segmentation," *Proc. Int'l Conf. Image Processing,* 1997.

[8] C.C. Lin and W.C. Lin, "Extracting Facial Features by an Inhibitory Mechanism Based on Gradient Distributions," *Pattern Recognition,* vol. 29, pp. 2079-2101, 1996.

[9] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 696-710, July 1997.

[10] L. Wiskott, J. Fellous, N. Kruger, and C. V. der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 775-779, July 1997.

[11] C. Garcia, G. Simandiris, and G. Tziritas, "A Feature-Based Face Detector Using Wavelet Frames," *Proc. Int'l Workshop Very Low Bit Coding,* pp. 71-76, 2001.

[12] K. Yow and R. Cipolla, "Feature-Based Human Face Detection," *Image and Vision Computing,* vol. 15, no. 9, pp. 713-735, 1997.

[13] S. Jeng, H. Yao, C. Han, M. Chern, and Y. Liu, "Facial Feature Detection Using Geometrical Face Model: An Efficient Approach," *Pattern Recognition,* vol. 31, no. 3, pp. 273-282, 1998.

[14] D. Maio and D. Maltoni, "Real-Time Face Location on Gray-Scale Static Images," *Pattern Recognition,* vol. 33, pp. 1525-1539, 2000.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.

[16] K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 39-51, Jan. 1998.

[17] M. Yang, D. Kriegman, and N. Ahuja, "Face Detection Using Multimodal Density Models," *Computer Vision and Image Understanding,* vol. 84, pp. 264-284, 2001.

[18] A. Colmenarez and T. Huang, "Face Detection with Information-Based Maximum Discrimination," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 782-787, 1997.

[19] C. Garcia and G. Tziritas, "Wavelet Packet Analysis for Face Recognition," *Image and Vision Computing,* vol. 18, no. 4, pp. 289-297, 2000.

[20] H. Schneiderman and T. Kanade, "A Statistical Model for 3D Object Detection Applied to Faces and Cars," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 746-751, 2000.

[21] C. Liu, "A Bayesian Discriminating Features Method for Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 6, pp. 725-740, June 2003.

[22] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 130-136, 1997.

[23] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 23-38, Jan. 1998.

[24] D. Roth, M.-H. Yang, and N. Ahuja, "A SNoW-Based Face Detector," *Advances in Neural Information Processing Systems 12,* pp. 855-861, MIT Press, 2000.

[25] F. Fleuret and D. Geman, "Coarse-to-Fine Face Detection," *Int'l J. Computer Vision,* vol. 20, pp. 1157-1163, 2003.

[26] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 511-518, 2001.

[27] S. Li, L. Zhu, Z. Zhang, and A. Blake, H. Zhang, and H. Shum, "Statistical Learning of Multi-View Face Detection," *Proc. Seventh European Conf. Computer Vision,* pp. 67-81, 2002.
[28] Y. LeCun, "Generalization and Network Design Strategies," *Connectionism in Perspective,* R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, eds., Zurich, Switzerland: Elsevier, 1989.
[29] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," *Advances in Neural Information Processing Systems 2,* D. Touretzky, ed. Denver: Morgan Kaufman, pp. 396-404, 1990.
[30] G. Martin, "Centered-Object Integrated Segmentation and Recognition of Overlapping Hand-Printed Characters," *Neural Computation,* vol. 5, pp. 419-429, 1993.
[31] J. Wang and J. Jean, "Multi-Resolution Neural Networks for Omnifont Character Recognition," *Proc. Int'l Conf. Neural Networks,* vol. 3, pp. 1588-1593, 1993.
[32] S. Lawrence, C. Gilles, A. Tsai, and A. Back, "Face Recognition: A Convolutional Neural Network Approach," *IEEE Trans. Neural Networks,* vol. 8, no. 1, pp. 98-113, 1997.
[33] R. Vaillant, C. Monrocq, and Y. LeCun, "Original Approach for the Localization of Objects in Images," *IEE Proc. Vision, Image, and Signal Processing,* vol. 141, no. 4, pp. 245-250, 1994.
[34] D. Hubel and T. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *J. Physiology,* vol. 160, pp. 106-154, 1962.
[35] K. Fukushima, "Cognitron: A Self-Organizing Multilayered Neural Network," *Biological Cybernetics,* vol. 20, pp. 121-136, 1975.
[36] M. Mozer, "The Perception of Multiple Objects: A Connectionist Approach," *Connectionism in Perspective.* Cambridge, Mass.: MIT Press-Bradford Books, 1991.
[37] C. Garcia and M. Delakis, "A Neural Architecture for Fast and Robust Face Detection," *Proc. Int'l Conf. Pattern Recognition,* vol. 2, pp. 44-48, 2002.
[38] H. Rowley, S. Baluja, and T. Kanade, "Rotation Invariant Neural Network-Based Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 38-44, 1998.
[39] P. Viola and M. Jones, "Robust Real-Time Object Detection," *Proc. ICCV Second Int'l Workshop Statistical and Computational Theories of Vision—Modelling, Learning, Computing, and Sampling,* July 2001.

Christophe Garcia received the PhD degree in computer vision from the University of Lyon I, France, in 1994. He has been involved in various computer vision research projects during his stay of two years at the IBM Vision Automation Group, France, and one year at the Computer Vision Center of the Autonomous University of Barcelona, Spain. As a fellow of the European Consortium in Informatics and Mathematics (ERCIM), he spent one year at the German National Research Center (now Fraunhofer Institute), working toward the development of autonomous and multiagents robots. From 1997 to 2000, he was a researcher at the Foundation for Research and Technology Hellas (FORTH), Greece, where he was involved in several advanced EU projects in the field of video and image analysis. From 2000 to 2002, he was a visiting professor at the Computer Science Department of the University of Crete, Greece, where he was teaching Artificial Neural Networks and Pattern Recognition. In 2003, he spent 10 months at IRISA-INRIA, Rennes, France, working in the field of automatic video structuring and indexing. Since December 2003, Dr. Garcia has been a senior researcher at France Telecom R&D. His current technical and research activities are in the areas of neural networks, pattern recognition, video analysis, and computer vision.

Manolis Delakis received the MSc degree "Postgraduate Degree of Specialization" in computer science from the Department of Computer Science of the University of Crete, Greece, in 2003. He is pursuing a PhD degree in computer science at IRISA-INRIA, Rennes, France. His research interests include image processing, machine learning, neural networks, and video indexing.