FRONTAL FACE DETECTION USING SUPPORT VECTOR MACHINES AND BACK-PROPAGATION NEURAL NETWORKS

C. Kotropoulos N. Bassiou T. Kosmidis and I. Pitas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, Thessaloniki 540 06, Greece
costas@zeus.csd.auth.gr

ABSTRACT

Face detection is a key problem in building automated systems that perform face recognition/verification and model-based image coding. Two algorithms for face detection that employ either support vector machines or back-propagation feed-forward neural networks are described, and their performance is tested on the same frontal face database using the false acceptance and false rejection rates as quantitative figures of merit. The aforementioned algorithms can replace the explicitly-defined knowledge for facial regions and facial features in mosaic-based face detection algorithms.

Keywords : Face detection, support vector machines, back-propagation neural networks, bootstrapping, horizontal/vertical profiles

1. INTRODUCTION

Face detection has been an active research topic in computer vision for more than two decades. Many approaches have been proposed for face detection in still images that are based either on texture, depth, shape and color information or a combination of them. A comprehensive survey on face detection methods can be found in [1]. A probabilistic method based on density estimation in a high dimensional space using an eigenspace decomposition is proposed in [2]. A closely related work is the example-based approach in [3] for locating vertically oriented and unoccluded frontal face views at different scales by using a number of Gaussian clusters to model the distributions of face and non-face patterns. A mixture of linear subspaces has been used to model the latter distributions in [4] where a mixture of factor analyzers is employed to detect faces with wide variations. For the detection of upright, frontal views of faces in grayscale images, a neural network-based algorithm that applies one or more neural networks directly to portions of the input image and arbitrates their results is presented in [5]. This approach recognize both separate facial features and pairs of them, after a preprocessing step is applied in the input images. For frontal-view face detection, Pantic and Rothkrantz [18] analyze the vertical and horizontal histograms of the image and apply an algorithm based on HSV color model, which relative to the relative RGB model [19], in order to determine the head's vertical and horizontal outer boundaries and the contour of the face respectively. For profile-view images, a profile detection algorithm representing a spatial approach to sampling the profile contour from a thresholded image is used [17]. The application of Support Vector Machines (SVM) in frontal face detection in images is studied in [6, 7].

In this paper we build on the face detection algorithm proposed in [8] that is based on multiresolution images (also known as *mosaic images*). The algorithm attempts to detect a facial region at a coarse resolution and subsequently to validate the outcome by detecting facial features at the next resolution by employing a hierarchical knowledge-based pattern recognition system. A variant of this method has been proposed in [9] that allows for rectangular cells instead of square cells and provides estimates of the cell dimensions and the offsets so that the mosaic model fits the face image of a person by preprocessing the horizontal and the vertical profile of the image. The original algorithm [8] is based on images of reduced resolution that attempt to capture the macroscopic features of the human face. It is assumed that there is a resolution level where the main part of the face occupies an area of about 4×4 cells having as origin the cell marked by "X". Accordingly, a mosaic image, the so called *quartet* image is created for this resolution level. The grey level of each cell is equal to the average value of the grey levels of all pixels included in the cell. An abstract model for the face at the resolution level of the quartet image is depicted in Figure 1 where we denote by n and m the vertical and the horizontal quartet cell dimensions, respectively. We propose to replace the "hardwired" rules for the properties of either the image regions or the facial features by employing a general purpose pattern recognition algorithm to discriminate among face and non-face patterns. Such patterns are created by ordering lexicographically the grey levels of the quartet image cells that fall inside a window scanning

This work was supported by the European Union Research Training Network "Multi-modal Human-Computer Interaction" (HPRN-CT-2000-00111).

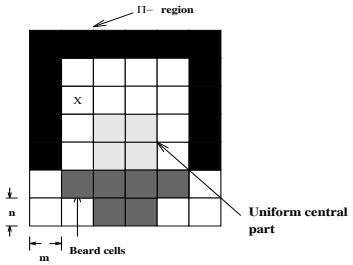
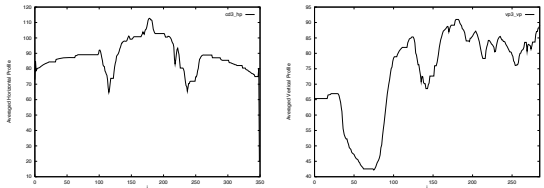


Fig. 1. Abstract face model at the quartet image.

the quartet image. Alternatively, one may use the horizontal and vertical image profiles in order to extract a bounding box for the face region, as has been demonstrated in [9]. The horizontal profile of the image is obtained by averaging all pixel intensities in each image column. Similarly, the vertical profile of the image is obtained by averaging all pixel intensities in each image row. Figure 2 depicts a typical horizontal and vertical image profile from the M2VTS database [10]. Instead of processing the extrema of the aforementioned profiles and defining rules that assign them to facial features, we propose to create patterns by scanning the horizontal and the vertical image profile with a running window.



(a) (b)

Fig. 2. (a) Horizontal profile. (b) Vertical profile.

Two supervised pattern recognition algorithms are tested in this paper. First, a support vector machine (SVM) is trained to separate face and non-face patterns extracted from the quartet image. Second, an ensemble of feed-forward neural networks trained by the back-propagation algorithm processes the horizontal and the vertical profile aiming at separating patterns that fall in the interval between the cheeks from the remaining patterns.Similarly another ensemble of feed-forward neural networks processes the vertical profile aiming at separating between the eyebrows and the chin from others.

The outline of the paper is as follows. The SVM face detection algorithm is described in Section 2. The back-propagation neural network approach is presented in Section 3. Experimental results are reported in Section 4 and conclusions are drawn in Section 5.

2. SUPPORT VECTOR MACHINE APPROACH

A two-dimensional (2-D) rectangular window is defined that consists of 5 cells in horizontal and 6 cells in vertical dimension. The window scans the quartet image whose cell intensities have been normalized to the interval $[0, 1]$. Between two successive movements, the windows are half overlapping. By moving the window over the quartet image, several 30-dimensional patterns are obtained that enable the description of faces appearing at different locations in the image. By varying the cell size, we enable the description of faces at different scales. To avoid the manual assignment of a class label to each feature vector, an empirical approach is used that exploits the face detection outcome provided by the method in [9]. Let us consider the coordinates of the upper left pixel in the detected rectangular face image region and its area. For each instance of the 2-D running window on the quartet image, we project the coordinates of the upper left cell at the resolution level of the original image and we estimate the area of the overlap between the 2-D running window and the detected face region. If the area of the overlap exceeds the 88% of the area of the detected face region, the corresponding feature vector is assigned to face patterns.

Let $\mathbf{x}_i$, $i = 1, 2, \ldots, l$ denote the i-th training pattern and t_i the class label assigned to it that takes the values ± 1, $\mathbf{t} = (t_1, t_2, \ldots, t_l)^T$. In order to deal with this non-linearly separable case, we have to determine the decision function

$$y = f(\mathbf{x}) = sign(\mathbf{w}^T \mathbf{x} + b) \tag{1}$$

where $\mathbf{w} = (w_1, w_2, ..., w_l)$ is a vector containing the weights that are to be estimated. To determine these weights, we built a SVM [14, 11] to solve the following quadratic programming problem with linear equality and inequality constraints

$$\text{minimize} \qquad \Phi(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}||\boldsymbol{w}||^2 + C(\sum_{i=1}^{l} \xi_i)^k \tag{2}$$

$$\text{subject to} \qquad t_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, l$$
$$\xi_i \geq 0 \tag{3}$$

where k, C are control parameters that penalize the violations of the linearly separable constraints after the introduction of slack variables $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_l)^T$ [12].

In order to solve the problem we introduce Lagrange multipliers. For each inequality constraint there is a Lagrange multiplier which expresses the change rate of the objective function subject to the changes in the respective constraint function. We construct the Lagrangian function

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\gamma}) \quad = \quad & \frac{1}{2}||\mathbf{w}||^2 + C(\sum_{i=1}^{l} \xi_i)^k - \\ & \sum_{i=1}^{l} \lambda_i [t_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \\ & \sum_{i=1}^{l} \gamma_i \xi_i \end{aligned} \tag{4}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_l), \boldsymbol{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_l)$ are the Lagrange multipliers of the first and second constraint function respectively.

The minimum $\mathbf{w}^*$ which is the solution to the problem must be a stationary point of the above function, thus the following relations must hold [12]:

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\gamma}) = 0 \iff \mathbf{w} = \sum_{i=1}^{l} \lambda_i t_i \mathbf{x}_i \quad (5)$$

$$\nabla_b \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\gamma}) = 0 \iff \sum_{i=1}^{l} \lambda_i t_i = 0 \quad (6)$$

$$\nabla_{\boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\gamma}) = 0 \Longleftrightarrow \frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}, \xi, \gamma)}{\partial \xi} = 0$$
$$\Longleftrightarrow \begin{cases} k\,C(\sum_{i=1}^{l} \xi_i)^{k-1} - \lambda_i - \gamma_i = 0 & k > 1 \\ C - \lambda_i - \gamma_i = 0 & k = 1 \end{cases} \quad (7)$$

According to the Wolfe dual theorem stating that if x^* solves a convex primal problem and the objective and constraint functions are differentiable, then x^*, λ^* solve the dual problem of the Lagrange function subject to the constraints that its derivative is 0 and the Lagrange multipliers are greater or equal to 0. Furthermore, the minimum primal and maximum dual function values are equal [12]. As a result the original minimization problem can be substituted by the following:

$$\text{Maximize} \quad \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}) = \frac{1}{2}||\mathbf{w}||^2 + C(\sum_{i=1}^{l} \xi_i)^k - \sum_{i=1}^{l} \lambda_i [t_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{l} \gamma_i \xi_i \quad (8)$$

$$\text{subject to} \quad \mathbf{w} = \sum_{i=1}^{l} \lambda_i t_i \mathbf{x}_i$$
$$\sum_{i=1}^{l} \lambda_i t_i = 0$$
$$\begin{cases} k\,C(\sum_{i=1}^{l} \xi_i)^{k-1} - \lambda_i - \gamma_i = 0 & k > 1 \\ C - \lambda_i - \gamma_i = 0 & k = 1 \end{cases}$$
$$\lambda_i \geq 0 \quad i = 1, \ldots, l$$
$$\gamma_i \geq 0 \quad i = 1, \ldots, l \quad (9)$$

From the third constraint of Eq. 9 we conclude that

$$\left.\begin{array}{ll} \lambda_i + \gamma_i = k\,C(\sum_{i=1}^{l} \xi_i)^{k-1} = const & k > 1 \\ \lambda_i + \gamma_i = C & k = 1 \end{array}\right\} \Longrightarrow$$
$$\Longrightarrow \lambda_i + \gamma_i = \tau \quad (10)$$

The substitution of Eq. 10 and of the the first constraint of Eq. 9 into the objective function of Eq. 8 yields the so-called *soft margin hyperplane* [6]:

$$\text{maximize} \quad \mathcal{L}(\boldsymbol{\lambda}, \tau) = \boldsymbol{\lambda}^T \mathbf{1} - \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{D} \boldsymbol{\lambda} - \frac{\tau^{\frac{k}{k-1}}}{(kC)^{\frac{1}{k-1}}} \cdot (1 - \frac{1}{k}) \quad (11)$$

$$\text{subject to} \quad \boldsymbol{\lambda}^T \mathbf{t} = 0$$
$$\boldsymbol{\lambda} \leq \tau \mathbf{1}$$
$$\boldsymbol{\lambda} \geq \mathbf{0} \quad (12)$$

where $\mathbf{1}$ is $(l \times 1)$ vector of ones, $\mathbf{0}$ is $(l \times 1)$ vector of zeros and $\mathbf{D}$ is an $l \times l$ matrix whose ij-element is given by $D_{ij} = t_i t_j (\mathbf{x}_i^T \mathbf{x}_j)$. The decision function implemented by the SVM is:

$$y = f(\mathbf{x}) = \text{sign} \left[\sum_{i=1}^{l} t_i \, \lambda_i^* (\mathbf{x}^T \mathbf{x}_i) + b^* \right] \quad (13)$$

where $b^* = t_i - (\mathbf{w}^*)^T \mathbf{x}_i$, for any support vector $\mathbf{x}_i$, i.e., a pattern whose the associated Lagrange multiplier satisfies $0 < \lambda_i < C$ and $\mathbf{w}^*$ is given by:

$$\mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* \, t_i \, \mathbf{x}_i \quad (14)$$

If the input patterns are mapped to a higher dimensional feature space through some non-linear mapping, the inner products in the feature space can be computed by a positive definite kernel function $K(\mathbf{x}, \mathbf{x}_i)$ [14]. To implement the above described algorithm, the $SVM^{light} Toolbox$ [13] has been used.

To model efficiently the non-face class in the training phase, we have used bootstrapping, as is proposed in [3]. For non-face patterns, any instance of the window in the background or in any other image not containing a face can constitute a non-face example. However, all these non-face patterns are not equally useful in modeling the non-face distribution. We used bootstrapping in order to select the non-face patterns that are close to face class boundaries [3, 6, 5]. That is, initially, the system is trained with a small number of face and non-face patterns and then it is tested on unknown images. The number of non-face patterns that are falsely detected as faces are inserted into the training set as negative examples.

3. NEURAL NETWORK APPROACH

An ensemble of neural networks $\mathcal{N}_i$ is created where each network is fed with patterns of the form

$$\begin{aligned} \mathbf{x}(n;\, i) &= (-1, x_1(n;\, i), x_2(n;\, i), \ldots, x_{2M+1}(n;\, i))^T \\ &= (-1, x_{i-M}(n), x_{i-M+1}(n), \ldots, x_i(n), \ldots, \\ &\quad x_{i+M-1}(n), x_{i+M}(n))^T \end{aligned} \quad (15)$$

for several randomly selected instances n of either the image profile of the same person or instances of images profiles of other persons from the training set. This means that for each input of the neural network we use different weight matrices which denote a different neural network. On Eq. (15), $x_q(n)$ denotes the q-th element of the n-th image profile. Each neural network has the topology depicted in Fig. 3, i.e., it is a feed-forward fully connected network. It is trained with the classical back-propagation algorithm

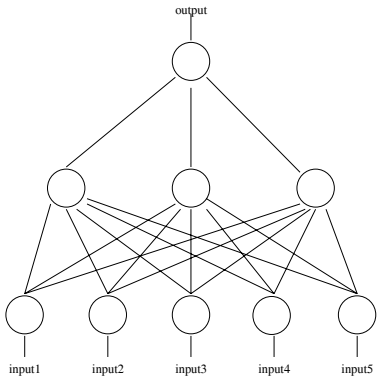


Fig. 3. Topology of the neural network employed.

[15]. Let $m = 0, 1, 2$ denote the layers of the neural network shown in Fig. 3. Let also $w_{kj}^{(m)}(n;\, i)$ be the synaptic weight of neuron k in layer m that is fed from neuron j in layer $m-1$. For $j = 0$, we have $y_0^{(m-1)}(n;\, i) = -1$ and $w_{k0}^{(m)}(n;\, i) = \theta_k^{(m)}(n;\, i)$, where $\theta_k^{(m)}(n;\, i)$ is the threshold applied to neuron k in layer m. The net internal activity level $v_k^{(m)}(n;\, i)$ of neuron k in layer m is given by:

$$v_k^{(m)}(n;\, i) = \sum_{j=0}^{2M+1} w_{kj}^{(m)}(n;\, i)\, y_j^{(m-1)}(n;\, i) \qquad (16)$$

with

$$y_j^{(0)}(n;\, i) = x_j(n;\, i), \quad j = 1, 2, \ldots, 2M+1. \qquad (17)$$

The output signal of neuron k in layer m is:

$$y_k^{(m)}(n;\, i) = \frac{1}{1 + \exp\left(-\beta\, v_k^{(m)}(n;\, i)\right)} \qquad (18)$$

where $\beta = 1.5$ or 2.5 for horizontal and vertical profiles, respectively. For the output neuron processing the pattern $\mathbf{x}(n;\, i)$, we define $o(n;\, i) = y_1^{(2)}(n;\, i)$. The error signal at the i-th element of the image profile is $e(n;\, i) = t(n;\, i) - o(n;\, i)$, where $t(n;\, i)$ is the desired response for the i-th element. The synaptic weights of the network in layer m are updated according to the generalized delta rule:

$$w_{kj}^{(m)}(n+1;\, i) = w_{kj}^{(m)}(n;\, i) + \alpha\, [w_{kj}^{(m)}(n;\, i) - w_{kj}^{(m)}(n-1;\, i)] + \eta\, \delta_k^{(m)}(n;\, i) y_j^{(m-1)}(n;\, i) \qquad (19)$$

where α is the momentum constant, η is the learning rate, and δ's are the local gradients. For output neurons (i.e., $k = 1$ and $m = 2$) we have:

$$\delta^{(2)}(n;\, i) = e^{(2)}(n;\, i)\, o(n;\, i)\, [1 - o(n;\, i)] \qquad (20)$$

while for neuron k in hidden layer m:

$$\delta_k^{(m)}(n;\, i) = y_k^{(m)}(n;\, i)\, \left[1 - y_k^{(m)}(n;\, i)\right] \cdot \sum_p \delta_p^{(m+1)}(n;\, i)\, w_{pk}^{(m+1)}(n;\, i). \qquad (21)$$

Both horizontal and vertical image profiles undergo a certain preprocessing before being fed to the network. First of all, they are smoothed by applying a running maximum filter of length 5 twice, so that the maxima become more prominent. Then, for each pattern $\mathbf{x}(n;\, i)$ the desired response or ground truth, $t(n;\, i) \in \{1, 0\}$, is coded considering whether the i-th element (i.e., a row index or column index) belongs to a face region or not. We consider as face region the area from the chin to the forehead in the vertical direction, and from the left to the right ear in the horizontal position. It is seen that the desired signal is a square wave signal with abrupt transitions. A branch of a Gaussian kernel is fit in each transition region so that more smooth transitions are provided to the neural network. A typical desired signal is depicted in Fig 4. The synaptic weights have

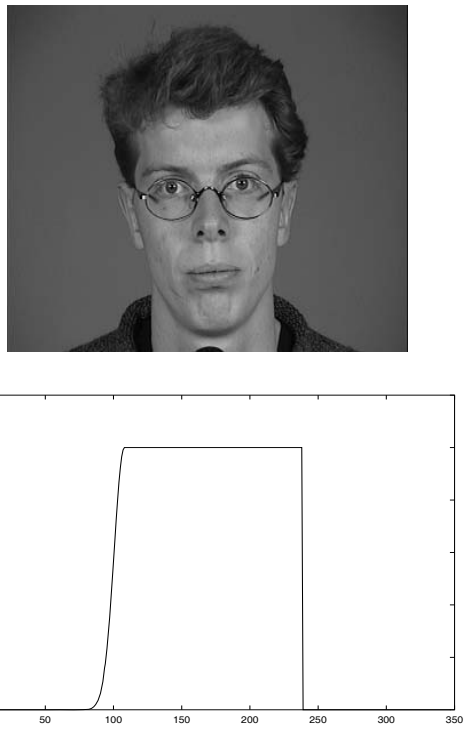


Fig. 4. Example of a desired response when profiles are being preprocessed.

been initialized randomly in the interval $[-1.5, 1.5]$. The constants α, η are set to 0.9 each. As stopping criterion,

we have used the condition the average mean squared error between the output of each neural network and the desired target becomes less than 0.07.

Moreover, we can augment the image profiles extracted from the frontal face images of the database with "synthetic" ones that are produced by adding Gaussian noise of zero mean and unit variance to the original image profiles.

4. EXPERIMENTAL RESULTS

The proposed algorithms have been applied to the European ACTS project M2VTS database [10]. The database includes the video-sequences of 37 different persons in four different shots. A training set is built from the frontal face images of the 37 persons in three shots. The algorithms are trained on this set. Frontal face images of the 37 persons from the fourth shot are used as test images. Rotations between the four available shots by leaving one shot out are also tested.

Two quantitative figures of merit have been used in the assessment of the performance of each algorithm namely the *false acceptance rate* (FAR) and the *false rejection rate* (FRR) during the test phase. The false acceptance rate is the ratio of non-face examples that have been wrongly classified as faces, while the false rejection rate is the ratio of face examples that have been failed to be detected, i.e., they have been rejected as non-faces. Receiver operating characteristic (ROC) curves (i.e., plots of FRR versus FAR) for a detection algorithm are provided whenever a tunable parameter (e.g., a threshold) is employed in the decision taking procedure.

4.1. SVM-based face detection

The pattern extraction algorithm yields roughly $1-10$ face patterns when each frontal face image is processed at several quartet cell resolutions. Accordingly, for each shot 200 face patterns result on average. When three shots are considered, a training set of 600 face patterns is formed. The following kernels have been employed during the training phase:

- Linear with $C = 1000$;
- Polynomial $K(\boldsymbol{\chi}, \boldsymbol{\psi}) = (s\,\boldsymbol{\chi}^T\boldsymbol{\psi} + c)^d$ with $c = 1$, $d = 3, 4, 5$ and 10;
- Radial Basis Function (RBF) $K(\boldsymbol{\chi}, \boldsymbol{\psi}) = \exp(-\gamma||\boldsymbol{\chi} - \boldsymbol{\psi}||^2)$ with $\gamma = 1$ and 5;
- Sigmoidal $K(\boldsymbol{\chi}, \boldsymbol{\psi}) = \tanh(s\,\boldsymbol{\chi}^T\boldsymbol{\psi} + c)$ with $c = 1$ and $s = 0.005$.

Table 1 summarizes the FAR and FRR obtained for all the kernels and the four combinations of test and training sets. Bootstrapping techniques are employed in SVMs with linear and polynomial kernel functions with $d = 5$. The corresponding rates obtained with and without bootstrapping are tabulated in Table 2. Table 3 illustrates the number of support vector obtained out of 600 feature vectors for the four training/test set combinations for several kernels and the number of iterations during each training. Table 4 summarizes the number of support vectors with and without bootstrapping for the linear and polynomial kernel, with $d = 5$, functions.

Table 1. False acceptance and false rejection rates for several kernels and test sets.

Test Set	Kernel	FA %	FR %
4	Linear	1.11	6.66
	Polynomial (d=3)	1.11	4.44
	Polynomial (d=5)	0	0
	RBF (γ=1)	0	0
	RBF (γ=5)	1.11	0
	Sigmoidal ($s = 0.005$)	1.11	2.22
3	Linear	3.62	1.21
	Polynomial (d=3)	3.62	1.21
	Polynomial (d=5)	3.61	5.81
	RBF (γ=1)	3.62	2.40
	RBF (γ=5)	3.61	3.61
	Sigmoidal ($s = 0.005$)	3.61	3.61
2	Linear	5	5
	Polynomial (d=3)	3	8
	Polynomial (d=5)	3	8
	RBF (γ=1)	1	5
	RBF (γ=5)	2	0
	Sigmoidal ($s = 0.005$)	2	0
1	Linear	1.88	1.88
	Polynomial (d=3)	1.88	1.88
	Polynomial (d=5)	1.88	1.88
	RBF (γ=1)	3.77	1.88
	RBF (γ=5)	2.83	0.94
	Sigmoidal ($s = 0.005$)	2.83	0.94

Table 2. False acceptance and false rejection rates (in %) achieved by linear SVMs with and without bootstrapping.

Test Set	Without Bootstrapping		With Bootstrapping	
	FA	FR	FA	FR
2	5.0	5.0	1.0	4.0
1	1.88	1.88	0.0	2.0

Table 3. Number of support vectors and of iterations for several kernels and test sets.

Test Set	Kernel	Number of Support Vectors	Number of Iterations
4	Linear	30	2274
	Polynomial (d=3)	36	1988
	Polynomial (d=5)	36	2171
	RBF (γ=1)	52	447
	RBF (γ=5)	192	216
	Sigmoidal ($s = 0.005$)	71	61
3	Linear	26	5670
	Polynomial (d=3)	35	1059
	Polynomial (d=5)	33	1286
	RBF (γ=1)	45	225
	RBF (γ=5)	189	209
	Sigmoidal ($s = 0.005$)	42	73
2	Linear	29	2474
	Polynomial (d=3)	36	2492
	Polynomial (d=5)	37	2082
	RBF (γ=1)	49	533
	RBF (γ=5)	179	251
	Sigmoidal ($s = 0.005$)	74	91
1	Linear	29	2570
	Polynomial (d=3)	37	204
	Polynomial (d=5)	38	1266
	RBF (γ=1)	52	341
	RBF (γ=5)	186	252
	Sigmoidal ($s = 0.005$)	70	105

Table 4. Number of support vectors acquired by linear and polynomial SVMs with and without bootstrapping.

Test Set	Without Bootstrapping		With Bootstrapping	
	Linear	Polynomial	Linear	Polynomial
4	30	36	35	38
3	26	33	36	36
2	29	38	30	39
1	29	39	36	46

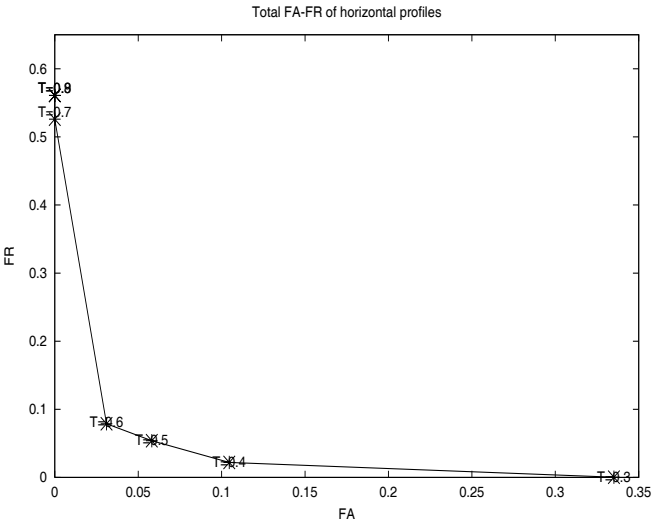


Fig. 5. Receiver Operating Characteristic curve when the horizontal image profiles are only considered.

4.2. Back-propagation neural network-based face detection

Experimental results are reported when the fourth shot has been used as test set. The neural network output is a signal taking values in the interval $[0, 1]$. The output is first smoothed by applying a running maximum filter of length 5, twice. To quantize the output as either 0 or 1 a threshold T is employed, so that when the output is greater than the threshold, the binary output is 1 (i.e., face pattern) and zero otherwise. Tests have been performed in the range $0.3 \leq T \leq 0.9$. In this case, the FAR and FRR values depend on the implicit parameter T. Accordingly, we may create ROC curves. The ROC curve when face detection is performed on the horizontal profiles only is depicted in Fig. 5. The corresponding curve, when the vertical profiles are only used, is shown in Fig. 6. The *equal error rate* (EER) is 5.01% for the ROC of Fig. 5 and 5.95% for the ROC of Fig. 6. Decisions taken on either the horizontal or the vertical profile independently can be combined using "AND" and "OR" rules. The false acceptance and false rejection rates are then given by [16]

$$FA_{AND} = fa_1 fa_2 \quad (22)$$

$$FR_{AND} = fr_1 + fr_2 - fr_1 fr_2 \quad (23)$$

$$FA_{OR} = fa_1 + fa_2 - fa_1 fa_2 \quad (24)$$

$$FR_{OR} = fr_1 fr_2 \quad (25)$$

where fa_1 and fr_1 are the FAR and FRR measured on the horizontal profile and fa_2 and fr_2 are the FAR and FRR measured on the vertical. The ROC curves are depicted in Figs. 7- 8, with EER 1.71% and 5.01% respectivly. Comparing the results with the ones of the previous method we conclude in Tables 5.

Table 5. Comparative FA-FR results of the two methods.

	Methods	FA %	FR %
SVM	Polynomial (d=3)	1.11	4.44
	Sigmoidal (s=0.005)	1.11	2.22
N.A.	AND combination	1.38	3.33
	OR combination	1.16	0.19

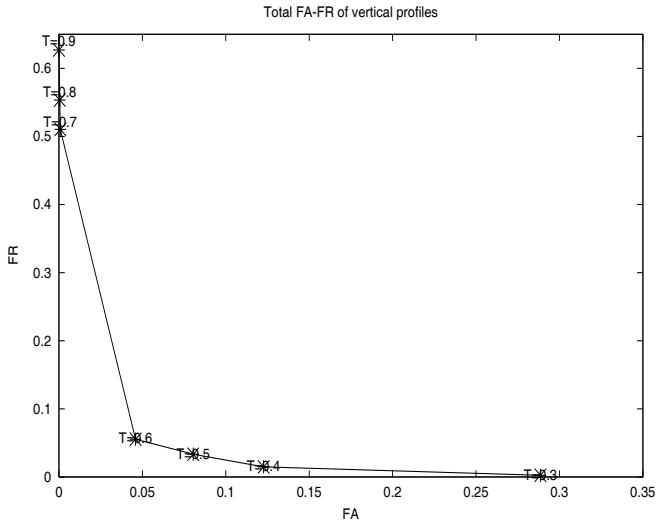


Fig. 6. Receiver Operating Characteristic curve when the vertical image profiles are only considered.

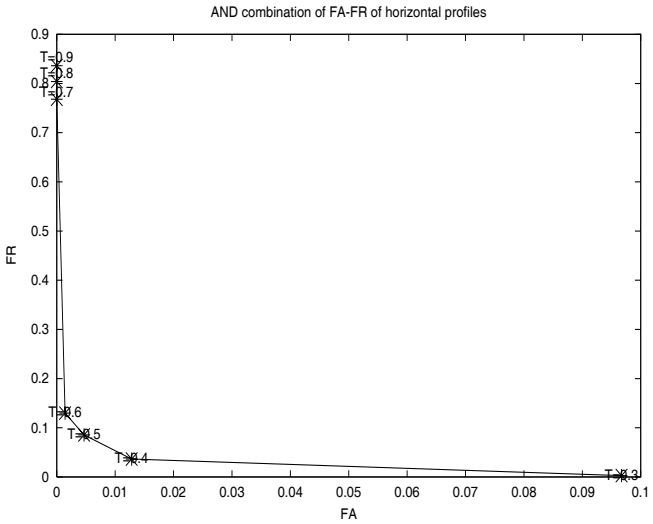


Fig. 7. Receiver Operating Characteristic curve when decisions taken independently on the horizontal and vertical image profiles are combined with an "AND" rule.

5. CONCLUSIONS

In this paper, two methods for detecting faces in frontal views have been described and their performance has been thoroughly measured with respect to the false acceptance and false rejection rates. These techniques are example-based and offer great flexibility in contrast to the knowledge-based approaches. They can replace the explicitly-defined knowledge for facial regions and facial features in mosaic-based face detection algorithms.

6. REFERENCES

[1] M.-H. Yang, N. Ahuja and D. Kriegman, "A survey on face detection methods, 1999. Available at `http://vision.ai.uiuc.edu/mhyang/papers/`.

[2] B. Moghaddam, and A. Pentland, "Probabilistic visual learning for object recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, July 1997.

[3] K.-K. Sung, and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, January 1998.

[4] M.-H. Yang, N. Ahuja, and D. Kriegman, "Face detection using a mixture of factor analyzers," in *Proc. of the 1999 IEEE Int. Conf. on Image Processing*, vol. 3, pp. 612-616, 1999.

[5] H.A. Rowley, S. Balujao, and T. Kanade, "Neural network-based face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–37, January 1998.

[6] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An application to face detection," in *Proc. of the IEEE Computer Society Computer Vision and Pattern Recognition Conf.*, pp. 130–136, 1997.

[7] C. Papageorgiou, M. Oren, and F. Girosi, "A general framework for object detection," in *Proc. of the Fifth Int. Conf. on Computer Vision*, pp. 555-562, 1998.

[8] G. Yang, and T.S. Huang, "Human face detection in a complex background," *Pattern Recognition*, vol. 27, no. 1, pp. 53–63, 1994.

[9] C. Kotropoulos, and I. Pitas, "Rule-based face detection in frontal views," in *Proc. of the 1997 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2537–2540, 1997.

[10] S. Pigeon, and L. Vandendorpe, "The M2VTS multimodal face database," in *Lecture Notes in Computer Science: Audio- and Video- based Biometric Person Authentication* (J. Bigün, G. Chollet and G. Borgefors, Eds.), vol. 1206, pp. 403–409, 1997.

[11] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.

[12] R. Fletcher, *Practical Methods of Optimization*, 2/e. Chichester, U.K.: J. Wiley & Sons, 1987.

[13] T. Joachims, "Making Large-Scale SVM Learning Practical," in B. Schölkopf, C.J.C. Burges and A.J. Smola, Eds. *Advances in Kernel Methods: Support Vector Learning*, pp. 41-56, Cambridge, MA: The MIT Press, 1998.

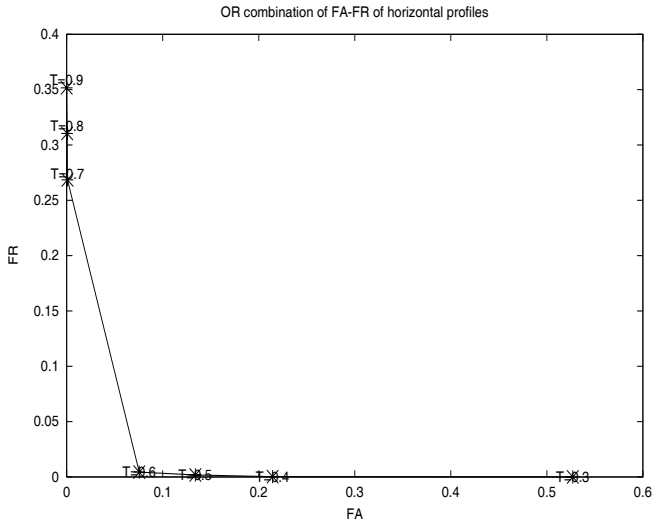


Fig. 8. Receiver Operating Characteristic curve when decisions taken independently on the horizontal and vertical image profiles are combined with an "OR" rule.

[14] V.N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer Verlag, 1995.

[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewoods Cliffs, N.J.: Prentice Hall, 1999.

[16] S. Pigeon, and L. Vandendorpe, "Image-based multimodal face authentication," *Signal Processing*, vol. 69, no. 1, pp. 59–79, 1998.

[17] M. Pantic, and L.J.M.Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, December 2000.

[18] M. Pantic, and L.J.M.Rothkrantz, "Expert System for Automatic Analysis of Facial Expression," *Image and Vision Computing J.*, vol 18, no. 11, pp. 881–905, 2000.

[19] J. Yang, and A. Waibel, "A Real-Time Face Tracker," *Workshop Applications of Computer Vision*, pp. 142–147, 1996.