2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece, □
Proceedings, Companion Volume, pp. 367-378

Robust Face Detection Based on Convolutional Neural Networks

M. Delakis and C. Garcia

Department of Computer Science, University of Crete
P.O. Box 2208, 71409 Heraklion, Greece
`{delakis, cgarcia}@csd.uoc.gr`

Abstract. Automatic face detection in digital video is becoming a very important research topic, due to its wide range of applications, such as security access control, model-based video coding or content-based video indexing. In this paper, we present a connectionist approach for detecting and precisely localizing semi-frontal human faces in complex images, making no assumption on the content or the lighting conditions of the scene, neither on the size, the orientation, and the appearance of the faces. Unlike other systems depending on a hand-crafted feature detection stage, followed by a feature classification stage, we propose a convolutional neural network architecture designed to recognize strongly variable face patterns directly from pixel images with no preprocessing, by automatically synthesizing its own set of feature extractors from a large training set of faces. Moreover, the use of receptive fields, shared weights and spatial subsampling in such a neural model provides some degrees of invariance to translation, rotation, scale, and deformation of the face patterns. We present in details the optimized design of our architecture and our learning strategy. Then, we present the process of face detection using this architecture. Finally, we provide experimental results to demonstrate the robustness of our approach and its capability to precisely detect extremely variable faces in uncontrolled environment.

1 Introduction

Human face processing is becoming a very important research topic, due to its wide range of applications, like security access control, model-based video coding or content-based video indexing. Face recognition and expression analysis algorithms have received most of the attention in the academic literature in comparison to face detection. In recent years, considerable progress has been made on the problem of face recognition, especially under stable conditions such as small variations in lighting, facial expression and pose. An interesting survey may be found in [1]. Most automatic face recognition and expression analysis algorithms have either assumed that the face have been cropped from the image or used "mugshot" images with uniform background so that the face is detected in a trivial way. However, the task of face detection is not trivial in complex scenes. Face patterns can present significant variations due to differences in facial appearance, expression and orientation.

Some techniques have been developed recently for detecting faces in “non-mugshot" images. These methods can be roughly divided into three broad categories: local facial features detection, template matching and image invariants. In the first case, low level computer vision algorithms are used to detect facial features such as eyes, mouth, nose and chin and statistical models of human face are used like in [3, 7, 13] among others. In the second case, several correlation templates are used to detect local sub-features. These features can be considered as rigid in appearance (view-based eigenspaces [8]) or deformable (deformable templates [12, 5]). The main drawback of these approaches is that either little global constraints are applied on the face template or extracted features are strongly influenced by noise or change in face expression or viewpoint. In the last case, image-invariant schemes assume that there are certain spatial image relationships, like brightness distribution, common and possibly unique to all face patterns, even under different imaging conditions [10]. They proved not to be robust in non-constrained scenes.

The use of skin color information can be an important cue for constraining the search space. In [4], Garcia and Tziritas proposed a fast method for detecting faces using skin color filtering and probabilistic classification of face texture based on statistical measures extracted from a wavelet packet decomposition. In [5], Garcia at al. extended this method for precise localization of facial feature by using a deformable face template.

In the general case of gray-level images, instead of detecting faces by following a set of human-designed rules, approaches based on neural networks like in [11, 9] have proven to give the best results. In this paper, we present a novel neural network based approach for detecting and precisely localizing semi-frontal human faces in complex images, making no assumption on the content or the lighting conditions of the scene, neither on the size, the orientation, and the appearance of the faces.

Unlike other systems depending on a hand-crafted feature detection stage, followed by a feature classification stage, we propose a convolutional neural network architecture designed to recognize strongly variable face patterns directly from pixel images with no preprocessing, by automatically synthesizing its own set of feature extractors from a large training set of faces. The use of receptive fields, shared weights and spatial subsampling in such a neural model provides some degrees of invariance to translation, rotation, scale, and deformation of the face patterns.

We first present the optimized design of our architecture and our learning strategy. Then, we present the process of face detection using this architecture. Finally, we provide experimental results and a comparison to the methods we described in [5] to demonstrate the robustness of our approach and its capability to precisely detect extremely variable faces in uncontrolled environment.

2 The Proposed Approach

2.1 Neural Network Architecture

The problem of finding face patterns is very difficult due to the large variety of distortions we have to take into account. These distortions include different facial expressions, environmental conditions, perspective of view etc. After any trial of manually enumerating every possible situation, we can easily conclude that this procedure is endless. Therefore, we need a machine-learning approach such as a neural network system. The standard, unstructured, fully connected topologies have the disadvantage of requiring a large amount of training data because they do not have any other way of encoding all the possible variations of the pattern. Instead of this, we can encode prior knowledge about the nature of the problem directly to the structure of the network. Such knowledge is relevant to the locality of the features, the invariance to translation, orientation, distortions, etc. One class of neural networks that are able to encode these notions are the convolutional neural networks [6].

The convolutional neural network we use is shown in Fig.1. The network consists of six layers, the first four of them acting as so-called feature maps [6]. Layer C1 performs a convolution on the input image with an adaptive mask, followed by an additive bias. Note that the receptive fields of neighboring neurons overlap, as in a pure convolutional procedure. This mask (weights) is shared through all the neurons of the same feature map, so there are actually only four neurons in this layer. These neurons extract the same kind of features, independently of their precise location. The size of the masks we use is 5x5, so the network has a total of 104 weights in the first layer. The size of the feature maps of this layer is 28 columns x 32 lines, supposing that the input dimensionality is 32x36.

Layer S2 performs local subsampling of the corresponding outputs of the previous layer. More precisely, a local averaging over four outputs is performed followed by a multiplication by a trainable coefficient and an addition with a trainable bias. In this way, we reduce the dimensionality of the feature maps by a factor of four. Finally, the linear output of this procedure is passed through a sigmoid function (in our case, the hyperbolic tangent function). After subsampling, the exact location and specific condition of every feature becomes less important, which gives a strong degree of robustness to our network. The size of these feature maps is 14x16 while the number of weights used in this layer is 8.

Layers S1 and C2 are partially connected, as described in table 1. In this way, the ability to combine different kinds of features so as to compose new ones is added to the network. The procedure on the layers C2 and S2 is exactly the same as on the layers C1 and S1, with the exception that in layer C2 we use a 3x3 mask for convolution. There are 14 feature maps in these layers with a total of 168 weights and output dimensionality 6x7 each, on the S2 layer.

Table 1. Each column corresponds to one feature map of the C2 layer and each row to one feature map of the S1 layer. The connections are marked with an X.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	X	X							X	X	X			
2			X	X					X			X	X	
3					X	X				X		X		X
4							X	X			X		X	X

In the N1 and N2 layers, the actual classification is performed, after feature extraction and input dimensionality reduction is done. In layer N1, we have 14 neurons, each of them connected only to the corresponding feature map of the S2 layer. The single neuron of the output layer N2 is fully connected to all the neurons of layer N1. These final layers contain a total of 617 weights.

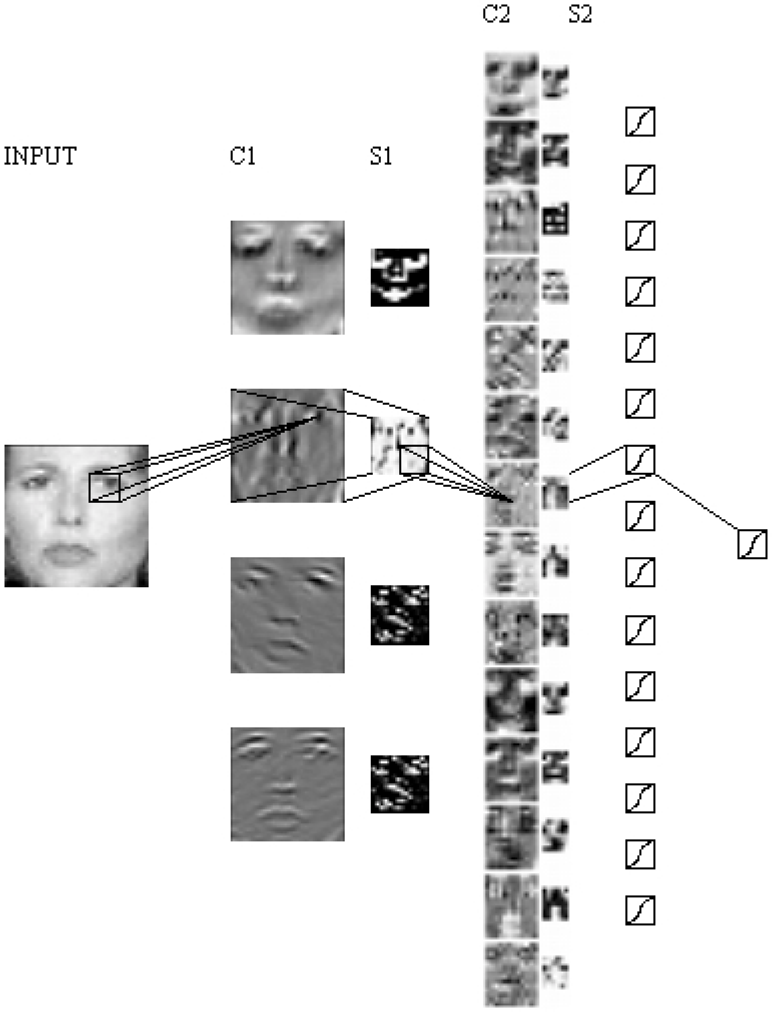

Fig. 1. The structure of the network we use. The contents of the feature maps show the different kind of features that have been actually detected from a real world example

Consequently, the proposed topology results in a global sum of only 897 trainable parameters, despite the 127,093 connections it uses. This structured topology is very promising to solve two problems at once: the problem of robustness, due to its nature, and the problem of good generalization, due to the relatively small amount of weights it uses.

2.2 Training Methodology

As a training set we use a large collection of images obtained from various sources over the Internet. This collection resulted in a database containing highly variable examples, which is not the usual case for the majority of the face databases we checked. As input dimensionality most of the neural network based approaches in the literature [11, 9] use as input a 20x20 window considering that it is the minimal resolution that can be used without loosing critical information from the face pattern. Usually, this window is the very central part of the face, excluding the border of the face and any background information. In our approach, we preferred to add to the input window the border of the face and some portions of the background. This is due to the way convolutional neural networks operate: they need the critical mass of information to be in the center of the input plane. On the other hand, by adding the border and some background, we give the network some additional information, which can help in classifying the face pattern more effectively. Note that the borders and the background must have a great variation, thus not introducing a serious bias to the network (for example, the background must not be always black). The final choice for the input size is 32x36 in order to roughly preserve the original aspect ratio of the pattern.

During the extraction of the faces, we did not perform any normalization, such as histogram equalization or brightness correction [11, 9]. In addition, we did not normalize the face examples so that the eyes, the mouth and other parts of the faces always remain exactly on the same position [11, 9]. Technically, this cannot be done without loosing the original aspect ratio of the face, which will introduce a bias to the network (all the examples will be size-normalized in this way). In addition, we mentioned that this network topology is quite robust to varying scale and position, so we need to boost this robustness by giving examples that are not normalized. Fig. 2 presents some of the examples we use.

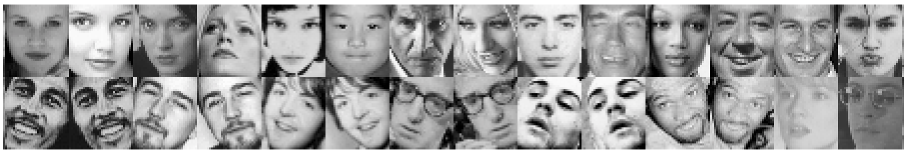

Fig. 2. Some examples of the extracted 2146 faces. In the second line there are some examples of the rotation and the contrast reduction transformation.

Rotation and gray level variance in real-world faces are taken into consideration by applying rotations (±20 degrees) and, then, contrast reduction to all the examples (including the rotated ones). The latter is important for obtaining better performance in bad-quality images, due to the fact that most of the original training examples are of very good quality. Another solution to this problem could be to normalize the gray levels of the test images, applying histogram equalization. This may be dangerous as it may introduce unexpected false alarms from textured surfaces. Some examples of transformations are shown in Fig. 2. Finally, after applying the transformations, the size of the training set reached the number of 12,876 examples.

For training the network, we used the backpropagation algorithm modified for use on convolutional networks as described in [6]. Desired responses are set to –1 for non-faces and to +1 for faces. Randomly cropped stimuli from images not containing faces could be used as false (non-face) examples. In general, we believe that this method is not the optimal one because false examples as close as possible to the boundary of the target class are needed. As an alternative solution, we trained the network using such false examples, just for getting the false alarms it produces. Next, the randomly selected false examples were replaced by the false alarms for the actual training of the network. In addition, we applied to the new set of false examples all the transformations we discussed above, for avoiding any bias that could be introduced. As a result, we produced approximately 6,000 false examples.

For producing more false examples, close to the class boundaries, we followed a bootstrapping procedure. Bootstrapping is a widely used solution for such classification problems due to the inability to predict all the possible false examples that a machine-learning algorithm may need. Note that, during the bootstrapping procedure, we were gradually decreasing the threshold values for grabbing false alarms. In this way, only the most suitable and helping false alarms were used. Also note that in every bootstrapping iteration the same network is re-trained, instead of building a new one. The false alarms we get from one network are not likely to occur again on a second network (without the same initialization).

In Table 2 the results of the training procedure are reported. For the state of the art the results of a minimized architecture with only one layer of feature maps (4 feature maps followed by 4 partially connected neurons and finally one output neuron) are also presented. As it is expected, the minimized topology produces much more false alarms (the training set size in the second case is much larger). The errors on the two validation sets, which are used to stop the training at the best-performing point, show that it is better to use the network having two layers of feature maps.

Table 2. The errors of the two network topologies on the training and on the validation set, after learning. An error is encountered when the answer has not the same sign with the desired output value.

	Training set	Validation set
Network of Fig. 1	900/27997 (3.21%)	17/800 (2.12%)
Minimized topology	1475/42770 (3.44%)	46/800 (5.75%)

2.3 Finding Faces Using the Neural Filter

Our system acts like a filter receiving a 32x36 image and generates an output ranging from -1 to 1, signifying the presence or absence of a face, respectively. In order to detect faces of different sizes, the input image is repeatedly subsampled via a factor of 1.2, resulting in a pyramid of images. Each image of the pyramid is filtered by the convolutional neural network with the fixed input size 32x36. In [11, 9], the filter is applied at every pixel of each image of the pyramid, given that it has very small invariance in position and scale.

In our case, a better invariance in position and scale allows us to filter the input with a step of 4 in both directions, resulting in speeding up the process significantly. This search may be seen as a rough localization, where the positive answers of the network correspond to candidate faces. First, candidate faces in each scale are mapped back to the input image scale. They are then iteratively grouped according to their proximity in image and scale spaces. More precisely, every candidate face *i* is represented by a vector (x_i, y_i, h_i, w_i, o_i) where (x_i,y_i) is the face center coordinate, (h_i,w_i) are the height and width of the face and o_i is the network answer. The candidate faces are stored in a list sorted in a decreasing order according to the scores o_i. The grouping algorithm is described in Table 3.

After applying this grouping algorithm, the N found clusters correspond to N face candidates (X_n,Y_n,W_n,H_n,O_n) which will serve as a basis for the next stage of the algorithm for face localization and false alarm dismissal.

Table 3. The grouping algorithm.

```
for each not yet assigned list element i with score $o_i$
    n=n+1
    Assign list element i to a new cluster $C_n$ with center ($X_n$,$Y_n$,$W_n$,$H_n$,$O_n$)=
     (xi,yi,wi,hi,oi)
    for each not yet assigned list element j
            if list element j is such that ($x_j$,$y_j$) belongs to
            the rectangle which top left and bottom right points are respectively
            [$X_n$-$W_n$/4, $Y_n$-$H_n$/4] and [$X_n$+$W_n$/4, $Y_n$+$H_n$/4]
            assign list element j to cluster $C_n$ and update cluster center:
                        $X_n=\sum_{k \text{ in cluster}} o_k x_k / \sum_{k \text{ in cluster}} o_k$
                        $Y_n=\sum_{k \text{ in cluster}} o_k y_k / \sum_{k \text{ in cluster}} o_k$
                        $H_n=\sum_{k \text{ in cluster}} o_k h_k / \sum_{k \text{ in cluster}} o_k$
                        $W_n=\sum_{k \text{ in cluster}} o_k w_k / \sum_{k \text{ in cluster}} o_k$
                        $O_n=\text{MAX}_{k \text{ in cluster}} o_k$
            end if
    end for
end for
N=n
```

A fine search is performed in an area around each rough face candidate center in image-scale space. A search space centered around the face candidate position is defined in image-scale space for precise localization of the candidate face. It corresponds to a small pyramid centered at the face candidate position covering 6 scales varying from 0.7 to 1.3 of the scale corresponding to the face candidate. For every scale, the presence of a face is evaluated on a grid of 6 pixels around the corresponding rough face candidate center position. Usually true faces will give high responses in 2 or 3 consecutive scales, but non-faces not so often. According to this phenomenon, we count the number of filter responses greater than a threshold in the fine search space. The selection among the candidate faces is performed according to the decision parameters THR_FACE and NOK, the threshold for a good response and the threshold for a sufficient number of good responses, respectively. One example illustrating the different phases of the face detection process is shown in Fig. 3.

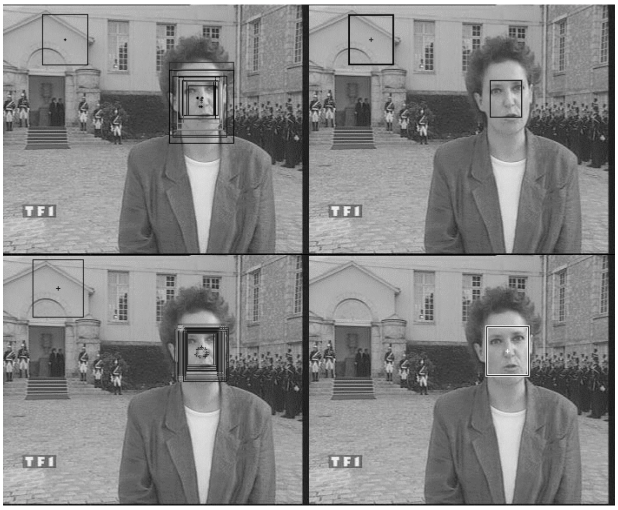

Fig. 3. An example of the face detection process. Each image presents the different steps of the algorithm. The first image shows the candidate faces detected in the rough search phase. The second image shows the candidate face clusters. The third image shows the candidate faces found in the fine search and the last image shows the final results with NOK=4 and THR_FACE=0.25.

3 Experimental Results

The proposed method has been evaluated using the same test data set as in [1]. This test data set contains images that have been extracted as key-frames from various MPEG videos and especially from the test videos used in the DiVAN project evaluation phase [11]. The video material has been kindly provided by the Institut National Audiovisuel, France and by ERT Television, Greece. The test data set contains 100 images, most of them being extracted from advertisements, news, movies and external shots. This set of 100 images contains 124 faces (minimal size being 24x20 pixels) and ten images which do not contain faces. They cover most of the cases that the algorithm has to deal with, i.e., large variability in size, illumination, facial expression and orientation. Moreover, most of the backgrounds are extremely textured. The faces contain in this test set are independent from the one contained in the training set.

In Fig. 4., we present some results of the proposed face detection scheme for 15 images of the test data set. These examples include images with multiple faces of different sizes and different poses. Some false alarms are presented as well.

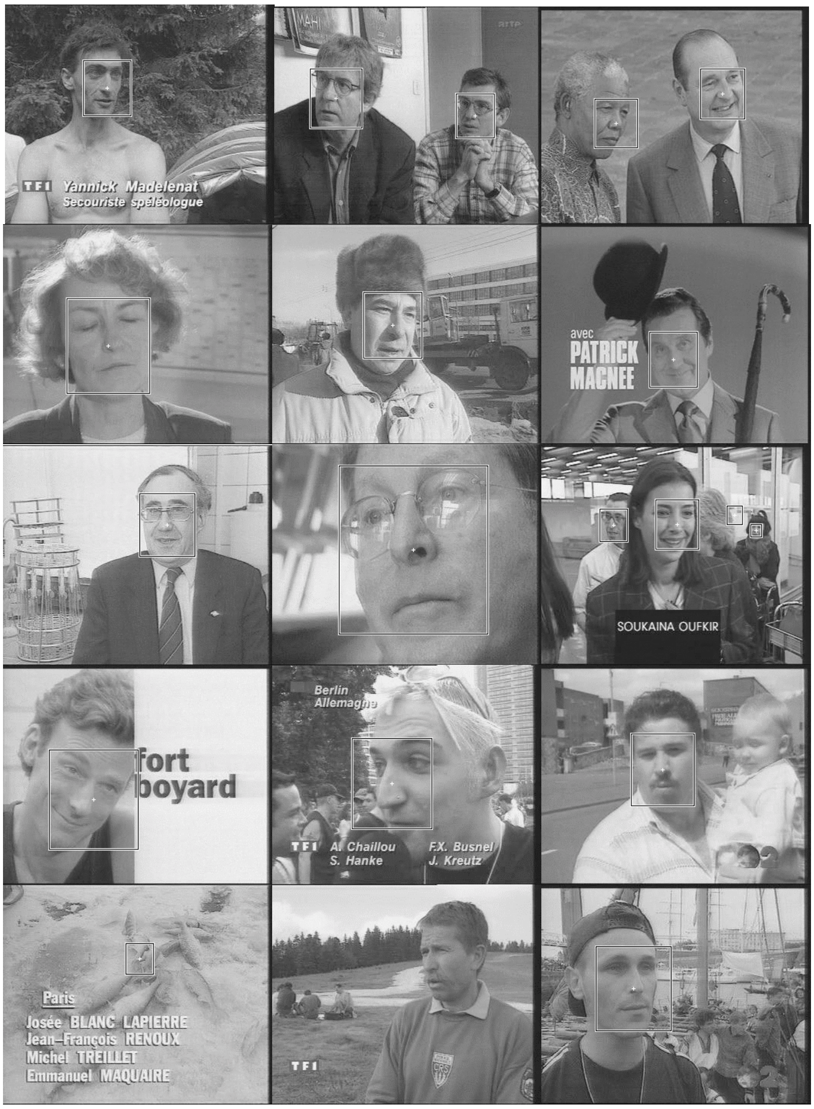

Fig. 4. Some results of the proposed method.

On this test set we obtained a good detection rate of 95.1% with 3 false alarms for NOK=4 and THR_FACE=0.25. It should be noted that the number of false alarms is very small even for a small value of THR_FACE. This may illustrate the capacities of the convolutional network architecture to highly separate face examples from non-face examples. As a comparison, the approach proposed in [4] gave 94.23% of good detection rate with 20 false alarms when 104 faces (of size greater than 80x48 pixels which is the minimal size for this approach) are considered. Considering this subset of 104 faces, the approach proposed in this paper gave 99.0% (one missed face) of good detection rate and one false alarm. The system in [9] resulted in 85.57% of good detection and 15 false alarms. An interactive demonstration of our system is available on the Web at http://csd.uoc.gr/~cgarcia/FaceDetectDemo, allowing anyone to submit images and to see the detection results for pictures submitted by other people.

4 Conclusions

Our experiments have shown that using convolutional neural networks for face detection is a very promising approach. The robustness of the system to varying poses, lighting conditions, facial expressions and image qualities was evaluated using a large set of non-trivial images. In addition, stability of responses in consecutive scales and a precise localization of faces were noticed. Using this network and the grouping strategy, we are allowed to quickly scan the input image and then to reject the small number of sparse false alarms. This is an advantage of our system, compared to other connectionist approaches, which require a dense scanning of the input image in all scales and positions [11, 9]. Finally, the relatively small number of training examples we used proves that our topology generalizes in a fine way and can be scaled easily to a greater training set, if needed.

As an extension of this work, we believe that the face detector should handle higher-level semantic information about facial features in order to solve some problems of occlusions, non-trivial poses etc. Another interesting point could be the use of video image sequences instead of still images for training. In such an environment, the fact that a face is a 3D object projected to the 2D space could be learned and help the network to be more robust in varying pose.

References

1. R. Chellappa, C.L. Wilson, S. Sirohey. Human and Machine Recognition of faces: A survey. In Proceedings of IEEE, 83(5), 705-740, 1995.
2. DiVAN: Distributed audio-Visual Archives Network (European Esprit Project EP 24956). http://divan.intranet.gr/info, 1997.
3. A. Eleftheradis and A. Jacquin Model-assisted coding of video teleconferencing sequences at low bit rates. In Proceedings of IEEE Int. Symp. Circuits and Systems, pp. 3.177-3.180, 1994.
4. C. Garcia and G. Tziritas. Face Detection Using Quantized Skin Color Region Merging and Wavelet Packet Analysis. IEEE Transactions On Multimedia,1(3), pp. 264-277, September 1999.

5. C. Garcia, G. Simandiris and G. Tziritas. A Feature-based Face Detector using Wavelet Frames. In: Proceedings of International Workshop on Very Low Bit Coding, pp. 71-76, Athens, Ooctober 2001.
6. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. Intelligent Signal Processing, pp. 306-351, 2001.
7. S.-H. Jeng, H. Y. M. Yao, C. C. Han, M. Y. Chern and Y. T. Liu. Facial Feature Detection Using Geometrical Face Model: An Efficient Approach. Pattern Recognition, 31(3), pp. 273-282, 1998.
8. A. Pentland, R.W. Picard, S. Sclaroff. Photobook: Content-Based Manipulation of Image Databases. in: Proc. of the SPIE Storage and Retrieval and Video Databases II, 1994.
9. H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 20, pp. 23-28, 1998.
10. P. Sinha. Object Recognition via Image Invariants: A Case Study. Investigative Ophthalmology and Visual Science, 35, pp. 1.735-1.740, 1994.
11. K. K. Sung and T. Poggio, “Example-based learning for view-based human face detection,” IEEE Trans. Pattern Analalysis Machine Intelligence., vol. 20, pp. 39–51, 1998.
12. L. Wiskott, JM. Fellous, N. Kruger, C. Von der Malsburg. Face Recognition by Elastic Bunch Graph Matching. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(7), pp. 775-779, 1997.
13. K. C. Yow, C. Cipolla. Feature-based human face detection. Image and Vision Computing, 15, pp. 713-735, 1997.