Advanced Face Recognition System for Real Time Applications

Naga Venkateshwara Rao.Kollipara [#1], T.Sudheer Kumar [*2]

[#*]*ECE Department, St.Martins Engineering College, Dhulapally(v),Kompally,Secunderabad-500100,Telangana State ,India*

[2]Sudheerkumarece@smec.ac.in

[1]Nagaece@smec.ac.in

***Abstract*— A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database. They are several biometrics used for Person Recognition like Iris, Fingerprints, Face etc. Since Iris and Fingerprints are very short-distance biometrics, but most applications require a person to be at a medium distance from the camera. Face recognition can be applied without the subject's active participation, and indeed without the subject's knowledge. Thus face recognition systems are preferred. The proposed paper uses Viola Jones algorithm with resizing Haar feature for detecting the human faces, this face is further processed and finally it is recognized with the help of Gaussian Mixture Model. With so much of our everyday communication and commercial activities now taking place via the Internet, the threat from cybercrime is increasing, targeting citizens, businesses and governments at a rapidly growing rate. This paper aims at providing a robust solution for such scenarios through face recognition**

***Keywords*— Viola Jones algorithm, Haar feature, Gaussian Mixture Model, Person Recognition, Face recognition**

I. INTRODUCTION

Face recognition has been one of the most interesting and important research fields in the past two decades. The reasons come from the need of automatic recognitions and surveillance systems, the interest in human visual system on face recognition, and the design of human-computer interface, etc. These researches involve knowledge and researchers from disciplines such as neuroscience, psychology, computer vision, pattern recognition, image processing, and machine learning, etc. A bunch of papers have been published to overcome difference factors (such as illumination, expression, scale, pose, ……) and achieve better recognition rate, while there is still no robust technique against uncontrolled practical cases which may involve kinds of factors simultaneously.

The current evolution of computer technologies has envisaged an advanced machinery world, where human life is enhanced by artificial intelligence. Indeed, this trend has already prompted

Figure 1: Typical training images for face recognition.

. Computer vision, for example, aims to duplicate human vision. Traditionally, computer vision systems have been used in specific tasks such as performing tedious and repetitive visual tasks of assembly line inspection. Current development in this area is moving toward more generalized vision applications such as face recognition and video coding techniques.

Many of the current face recognition techniques assume the availability of frontal faces of similar sizes [2]. In reality, this assumption may not hold due to the varied nature of face appearance and environment conditions. Consider the pictures in figure 1 these pictures are typical test images used in face classification research.

The exclusion of the background in these images is necessary for reliable face classification techniques. Face recognition procedure is separated into three steps: Face Detection, Feature Extraction, and Face Recognition.

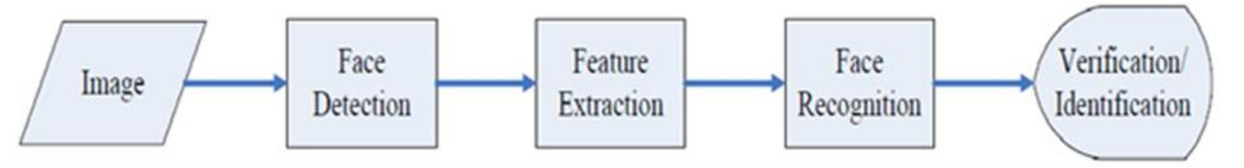

Figure2: Configuration of a general face recognition structure

A. *Face Detection:*

The main function of this step is to determine [1] whether human faces appear in a given image, and [2] where these faces are located at. The expected outputs of this step are patches containing each face in the input image. In order to make further face recognition system more robust and easy to design, face alignment are performed to justify the scales and orientations of these patches. Besides serving as the pre-processing for face recognition, face detection could be used for region-of-interest detection, retargeting, video and image classification, etc.

B. *Feature Extraction:*

After the face detection step, human-face patches are extracted from images. Directly using these patches for face recognition have some disadvantages, first, each patch usually contains over 1000 pixels, which are too large to build a robust recognition system1. Second, face patches may be taken from different camera alignments, with different face expressions, illuminations, and may suffer from occlusion and clutter[3]. To overcome these drawbacks, feature extractions are performed to do information packing, dimension reduction, salience extraction, and noise cleaning. After this step, a face patch is usually transformed into a vector with fixed dimension or a set of fiducially points and their corresponding locations. In some literatures, feature extraction is either included in face detection or face recognition.

C. *Face Recognition:*

After formulizing the representation of each face, the last step is to recognize. In order to achieve automatic recognition, a face database is required to build. For each person, several images are taken and their features are extracted and stored in the database. Then when an input face image comes in, we perform face detection and feature extraction, and compare its feature to each face class stored in the database. There have been many researches and algorithms proposed to deal with this classification problem, and we'll discuss them in later sections. There are two general applications of face recognition, one is called identification and another one is called verification[4]. Face identification means given a face image, we want the system to tell who he / she is or the most probable identification; while in face verification, given a face image and a guess of the identification, we want the system to tell true or false about the guess. In fig. 2, we show an example of how these three steps work on an input image.

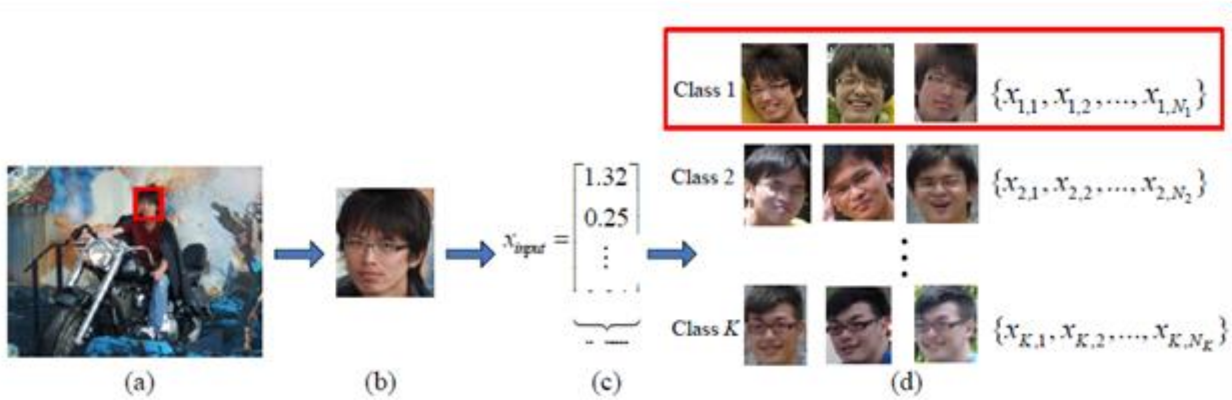

Figure 3: (a) The input image and the result of face detection (the red rectangle) (b) The extracted face patch (c) The feature vector after feature extraction (d) Comparing the input vector with the stored vectors in the database by classification techniques

The existing system uses Viola-Jones algorithm, Active Shape Model algorithm and Pose and Illumination normalization in order to detect faces and PCA to recognize. In my report faces are detected with Viola-Jones algorithm but with Haar feature resize and only Illumination normalization using Histogram Equalization and face is recognized using Gaussian Mixture Model.

II. PROPOSED FACE RECOGNITION SYSTEM

The block diagram for the face recognition system is shown below.

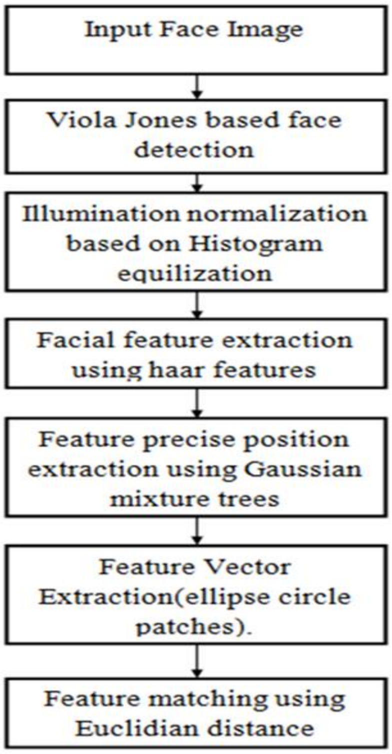

Figure 4: Block Diagram of Face Recognition System

A. Proposed method Algorithm

1. Capture the input image from the camera.
2. Give the image to face detection module using viola jones algorithm.
   - 2.1 get the image
   - 2.2 initialize the haar features
   - 2.3 run the haar features on the image in the first run.
   - 2.4 Resize the haar features then again run them on the image.
   - 2.5 Whenever a face is detected by the haar features, haar features return the starting point, height and width and the sclae of the haar features.

The face detection procedure classifies images based on the value of simple features. There are many motivations for using features rather than the pixels directly. The most common reason is that features can act to en-code ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. There is also a second critical motivation for using features which is the feature-based system operates much faster than a pixel-based system. More specifically, we use three kinds of features.

The value of a two-rectangle feature is the difference between the sum of the pixels within two rectangular regions[5]. The regions have the same size and shape and are horizontally or vertically adjacent (see Figure 4). A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a centre rectangle. Finally a four-rectangle feature computes the difference between diagonal pairs of rectangles. One of the main contributions to reduce time for face detection is internal integral image which allows very fast feature evaluation. In order to compute these features very rapidly at many scales we introduce the integral image representation for images.

The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Harr-like features can be computed at any scale or location in constant time.

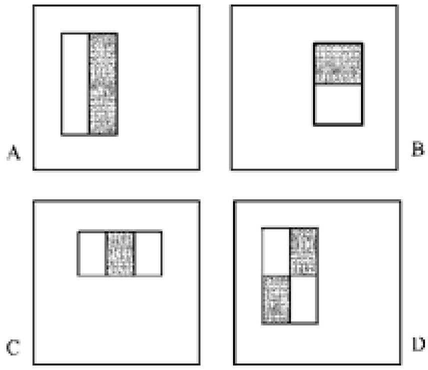

Figure 5: Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four- rectangle feature.

In normal viola jones algorithm the image is resized to several levels which is very complex because two standard scaling algorithms were used for image resizing which are Bilinear and Bi-cubic interpolation. Filters like these work by interpolating pixel colour values, introducing a continuous transition into the output even where the original material has discrete transitions. Although this is desirable for continuous-tone images, some algorithms reduce contrast (sharp edges) in a way that may be undesirable for line art[6].Nearest-neighbour interpolation preserves these sharp edges, but it increases aliasing (or jaggies; where diagonal lines and curves appear pixelated). Several approaches have been developed that attempt to optimize for bitmap art by interpolating areas of continuous tone, preserve the sharpness of horizontal and vertical lines and smooth all other curves. So image resizing consists of complex operations. But removing this and using the haar feature resize would reduce the complexity in which only three operations are involved.

B. Integral Image

Rectangle features can be computed very rapidly using an intermediate representation for the image which we call the integral image. The integral image at location x, y contains the sum of the pixels above and to the left of x, y, inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \tag{1}$$

Where ii (x, y) is the integral image and i (x, y) is the original image using the following pair of recurrences:

$s(x , y) = s(x , y - 1) + i (x , y)$ (i)
$ii (x , y) = ii (x - 1, y) + s(x , y)$ (ii)

(Where s(x, y) is the cumulative row sum, $s(x, -1) = 0$, and $ii (-1, y) = 0$) the integral image can be computed in one pass over the original image. Using the integral image any rectangular sum can be computed in four array references (see Fig. 3). Clearly the difference between two rectangular sums can be computed in eight references[7]. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

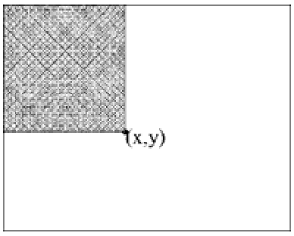

Figure 6: The value of the integral image at point (x , y) is the sum of all the pixels above and to the left.

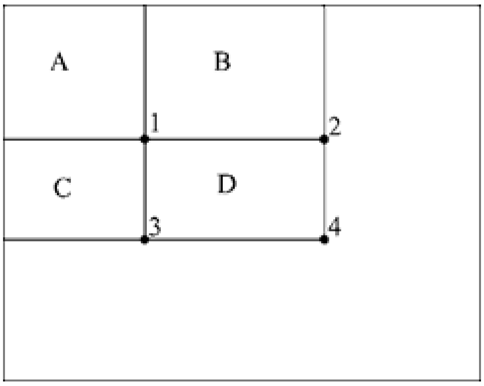

Figure 7: The sum of the pixels within rectangle D can be computed with four array references.

The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D. The sum within D can be computed as 4 + 1 − (2 + 3).

$$f * g = \int \int (f' * g'). \tag{2}$$

In the case of linear operations (e.g. f*g), any invertible linear operation can be applied to f or g if its inverse is applied to the result. For example in the case of convolution, if the derivative operator is applied both to the image and the kernel the result must then be double integrated:

$$(f'') * \left(\int \int g \right) = f * g. \tag{3}$$

Convolution can be significantly accelerated if the derivatives of f and g are sparse (or can be made so). A similar insight is that an invertible linear operation can be applied to f if its inverse is applied to g:

$$i \cdot r = \left(\int \int i \right) \cdot r''. \tag{4}$$

Viewed in this framework computation of the rectangle sum can be expressed as a dot product, i •r , where i is the image and r is the box car image (with value 1 within the rectangle of interest and 0 outside).The integral image is in fact the double integral of the image (first along rows and then along columns). The second derivative of the rectangle (first in row and then in column) yields four delta functions at the corners of the rectangle. Evaluation of the second dot product is accomplished with four array accesses. The appearance of each facial feature is assumed independent of the other features and is modelled discriminatively by a feature/non-feature classifier trained using a variation of the AdaBoost algorithm and using the "Haar-like" image features proposed. A collection of labelled consumer photographs was used to fit the parameters of the model and train the feature classifiers.

III. MIXTURE MODEL

In statistics, a mixture model is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population.

However, while problems associated with "mixture distributions" relate to deriving the properties of the overall population from those of the sub-populations, "mixture models" are used to make statistical inferences about the properties of the sub-populations given only observations on the pooled population, without sub-population identity information. Some ways of implementing mixture models involve steps that attribute postulated sub-population-identities to individual observations (or weights towards such sub-populations), in which case these can be regarded as types of unsupervised learning or clustering procedures. However not all inference procedures involve such steps [8].

Mixture models should not be confused with models for compositional data, i.e., data whose components are constrained to sum to a constant value (1, 100%, etc.). However, compositional models can be thought of as mixture models, where members of the population are sampled at random. Conversely, mixture models can be thought of as compositional models, where the total size of the population has been normalized to 1[12].

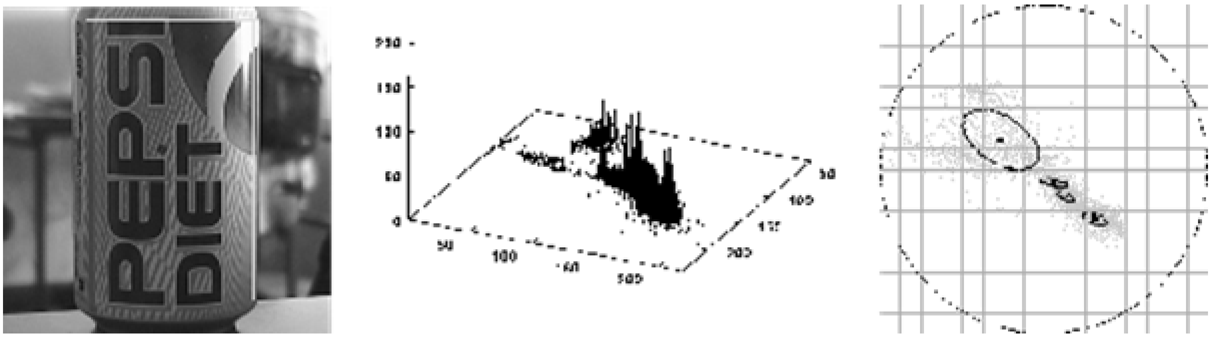

Figure 8: Shows a multi-colored object's histogram and GMM

Left: a multi-colored objects (a PEPSI can). Centre: its color histogram in HS-space. It can be noted that such a histogram representation is only viable when a large amount of data is available due to being non-parametric. Right: its Gaussian mixture model. The mixture components are shown as elliptical contours of equal probability Color mixture models of a multi-colored object (person model) and the context (scene model). The first row shows the data used to build the foreground (person) and the background (laboratory) models[9]. The second row illustrates the probability density estimated from mixture models for the object foreground and scene background. The rightmost image is the combined posterior density in the HS color space. Here the ``bright" regions represent foreground whilst the ``dark" regions give the background. The ``grey" areas are regions of uncertainty.

Figure 9: Shows the probability density from mixture model for different background

Now the more accurate features are extracted and separated as shown in figure 10, for eye two sections from both edges and one from center, similarly with nose and mouth. From these features, feature vector is calculated.

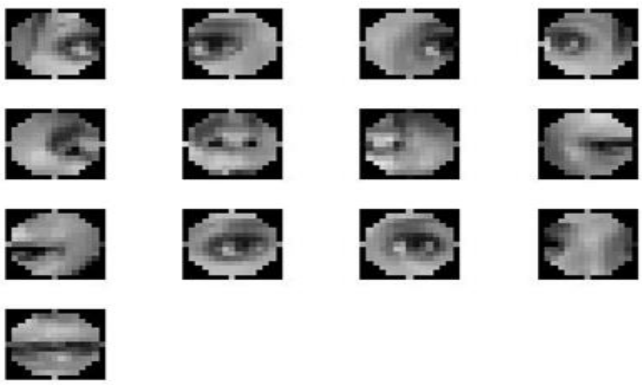

Figure 10: Shows how GMM points out features

A. Face Recognition Using Euclidian Distance

Very often, especially when measuring the distance in the plane, we use the formula for the Euclidean distance. According to the Euclidean distance formula, the distance between two points in the plane with coordinates (x, y) and (a, b) is given by

$$\text{dist}((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2} \quad (5)$$

As an example, the (Euclidean) distance between points (2, -1) and (-2, 2) is found to be

$$\text{dist}((2, -1), (-2, 2)) = \sqrt{(2 - (-2))^2 + ((-1) - 2)^2} \quad (6)$$

$$\text{dist}((2, -1), (-2, 2)) = \sqrt{(2 - (-2))^2 + ((-1) - 2)^2} \quad (7)$$

$= \sqrt{(2 + 2)^2 + (-1 - 2)^2}$
$= \sqrt{(4)^2 + (-3)^2}$
$= \sqrt{16 + 9}$
$= \sqrt{25}$
$= 5.$

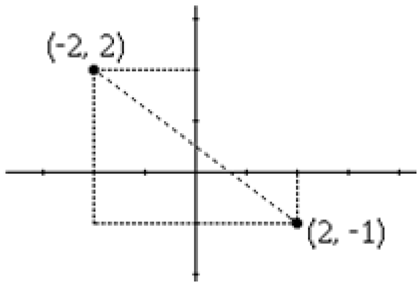

Figure 11: Shows the graphical representation of Pythagorean Theorem

The horizontal distance between the points is 4 and the vertical distance is 3. Let's introduce one more point (-2, -1). With this small addition we get a right-angled triangle with legs 3 and 4. By the Pythagorean theorem, the square of the hypotenuse is $(\text{hypotenuse})^2 = 3^2 + 4^2$. Which gives the length of the hypotenuse as 5, same as the distance between the two points according to the distance formula[10]. This is of course always the case: the straight line segment whose length is taken to be the distance between its endpoints always serves as a hypotenuse of a right triangle (in fact, of infinitely many of them. We just chose the most convenient one.) How good is the (Euclidean) distance formula for measuring real distances? This depends on the circumstances. In the plane - since the Earth is round, this means within relatively small areas of Earth's surface - it is pretty good, provided the distance is exactly what you want to estimate. If the question is, How fast you can get from one point to another while moving at a given speed, the Euclidean formula may not be very useful providing the answer. Indeed, in a city just to take one example, it is often impossible to move from one point straight to another.

There are buildings, streets busy with traffic, fences and what not, to be accounted for. In a city, one often finds that the taxicab distance formula $\text{dist}((x, y), (a, b)) = |x - a| + |y - b|$ is more useful. In mathematics, the Euclidean distance is most fundamental[11].

B. Software Required:

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyse data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spread sheets or traditional programming languages, such as C/C++ or Java. You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

IV. EXPERIMENTAL RESULTS

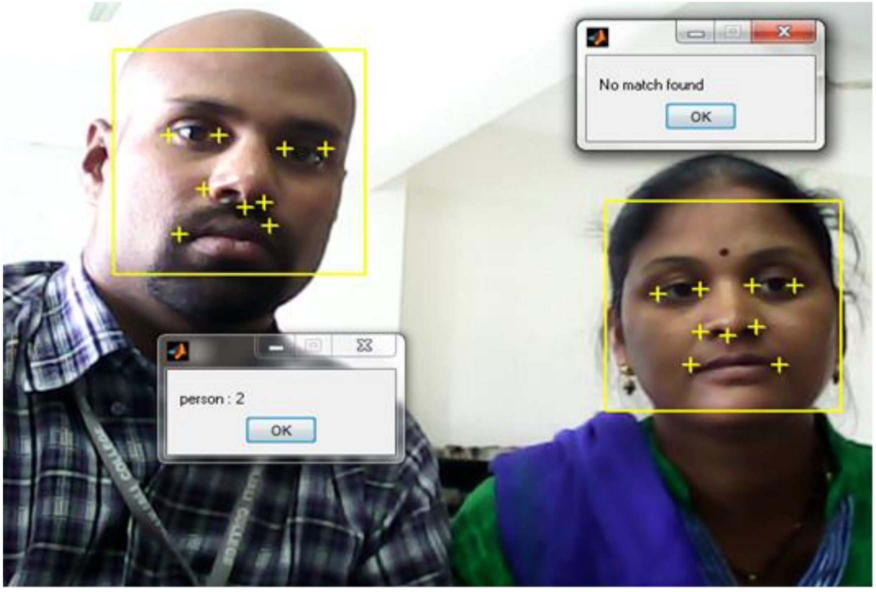

Figure 12: Shows the result of face recognition system

The above figure shows the output of the face recognition system. This system can detect any number of faces in an image and recognize the faces whose features are stored in the data base.

V. CONCLUSION

Face recognition is one of the several techniques for recognizing people. There are several methods that can be used for that purpose. A face recognition system is developed and implemented using Viola Jones algorithm with Haar feature resizing and faces are recognized with the help of Gaussian Mixture Model and Euclidean Distance. With this face recognition system, it achieved 89% recognition rate.

REFERENCES

[1]. Paul Viola and Michael, Rapid Object Detection using a Boosted Cascade of Simple Features, Accepted Conference on Computer Vision and Pattern Recognition 2001.

[2]. Rabia Jafri and Hamid R. Arabnia, A Survey of Face Recognition Techniques, Journal of Information Processing Systems, Vol.5, No.2, June 2009

[3]. Sinjini Mitra, Gaussian Mixture Models for Human Face Recognition under Illumination Variations, Applied Mathematics, 2012, 3, 2071-2079

[4]. Y. Amit, D. Geman, and B. Jedynak, Efficient focusing and face detection, in Face Recognition: From Theory to Application, Springer-Verlag, Berlin/New York, 1998.

[5]. P. J. L. Van Beek, M. J. T. Reinders, B. Sankur, and J. C. A. Van Der Lubbe, Semantic segmentation of videophone image sequences, in Proc. of SPIE Int. Conf. on Visual Communications and Image Processing, 1992, pp. 1182–1193
.
[6]. S. Ben-Yacoub, B. Fasel, and J. L¨uttin, Fast face detection using MLP and FFT, in Proceedings Second International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA), 1999.

[7] D. E. Benn, M. S. Nixon, and J. N. Carter, Extending concentricity analysis by deformable templates for improved eye extraction, in Proceedings Second International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA), 1999.

[8]. H. Bessho, Y. Iwai, and M. Yachida, Detecting human face and recognizing facial expressions using potential net, in Proceedings of the 15th International Conference on Pattern Recognition, 2000, Vol. I, p. 3A.

[9]. R. Brunelli and T. Poggio, Face recognition: Feature versus templates, IEEE Trans. Pattern Anal. Mach. Intell. 15, 1993, 1042–1052.

[10]. G. Burel and D. Carel, Detection and localization of faces on digital images, Pattern Recog. Lett. 15, 1994,963–967
.

[11]. M. C. Burl, T. K. Leung, and P. Perona, Face localization via shape statistics, in Int. Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, June 1995.

[12]. M. C. Burl and P. Perona, Recognition of planar object classes, in IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition, 6, 1996.