Machine Learning Techniques in Pain Recognition

Md. Maruf Monwar

B.Sc., University of Rajshahi, Bangladesh, 1996
M.Sc., University of Rajshahi, Bangladesh, 1997

Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Science
in
Mathematical, Computer and Physical Sciences
(Computer Science)

The University of Northern British Columbia
December, 2006

Your file Votre référence
ISBN: 978-0-494-28422-3
Our file Notre référence
ISBN: 978-0-494-28422-3

Abstract

Facial expressions are a key index of emotion. To make use of the information afforded by facial expression for emotion science and clinical practice, reliable, valid, and efficient methods of measurement are critical. Enabling computer systems to recognize facial expressions and infer emotions from them is a challenging research topic. This thesis presends an appearance-based approach for pain recognition from video sequences. An automatic face detector is employed which uses skin color modeling to detect human face in the video sequence. The pain affected portions of the face are obtained by using a mask image. Facial features are processed by both holistically and locally. Two machine learning approaches – eigenimage and multilayer neural network are used for recognition. The first approach processes features holistically and projected onto a feature space, to produce the biometric template. Recognition in this approach is performed by projecting a new image onto the feature spaces spanned by the eigenimage and then classifying the painful face by comparing its position in the feature spaces with the positions of known individuals. Eigenface, eigeneye and eigenlip techniques are used for this approach. The multilayer neural network technique processes facial features locally. Two types of features, location features and shape features are computed and then used as inputs to the artificial neural network which uses standard error back-propagation algorithm for classification of painful and non-painful faces.

Table of Contents

List of Tables

List of Figures

Publications from the Thesis

[1] **Md. Maruf Monwar** and Dr. Siamak Rezaei, **Pain Recognition Using Artificial Neural Network**, in proceedings of the 6th IEEE International Symposium on Signal Processing and Information Technology 2006 (**ISSPIT 2006**), ISBN: 0-7803-9754-1, August 27-30, 2006, Vancouver, Canada, pp. 28-33.

[2] **Md. Maruf Monwar** and Dr. Siamak Rezaei, **Appearance-based Pain Recognition from Video Sequences**, in the proceedings of the 2006 International Joint Conference on Neural Networks (**IJCNN 2006**), ISBN: 0-7803-9490-9, July 16-21, 2006, Vancouver, Canada, pp. 2429-2434.

[3] Md. Maruf Monwar and Dr. Siamak Rezaei, **A Robust Technique for Pain Recognition from Video Sequences using Skin Color Modeling**, in the proceedings of the International MultiConference of Engineers and Computer Scientists 2006 (**IMECS 2006**), ISBN: 978-988-98671-3-3, June 20-22, 2006, Hong Kong, pp. 513-518.

[4] **Md. Maruf Monwar** , Padma Polash Paul, Md. Wahedul Islam and Dr. Siamak Rezaei, **A Real-Time Face Recognition Approach from Video Sequence Using Skin Color Model and Eigenface Method**, in the proceedings of the 19th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2006), ISBN: 1-4244-0038-4, May 7 – 10, 2006, Ottawa, Canada, pp. 2150-2154.

Acknowledgment

I am greatly indebted to my supervisor, Dr. Siamak Rezaei, for kindly providing suggestions and encouragement which helped me in all the time of research and writing of this thesis. His comments have been of greatest help at all times.

My gratitude also goes to my supervisory committee members for their suggestions and encouragement that led to substantial improvements of this thesis. In particular, I am grateful to Dr. Ken Prkachin for his valuable feedbacks.

I would like to express my thanks to Dr. Moustafa of Physics department of the University of Northern British Columbia, Canada for his important suggestions and feedback.

I thank the Dean of Graduate Studies Dr. Robert Tait, the secretary of the Dean of Graduate Studies Ms. Bethany Haffner for having always been so supportive.

Finally I would like to acknowledge my thanks to my family. My wife, Nahid Sultana provided me constant support which helped me to overcome the many difficulties and discouragement on the way to completing this dissertation. My parents Mirza Md. Monwar Hossain and Mrs. Sufia Monwar, my sister Sharmin Monwar supported me with their encouragement and comprehension. My daughter, Rushama Nahiyan Raiyan has been a constant source of joy and gave me the strength I needed to go through the preparation of this dissertation.

Chapter 1

Introduction

1.1 Overview

The face plays a crucial role in interpersonal communication. Seeing a face, we can recognize a person's identity, gender, age, expression, etc. This information is irreplaceable for the normal conduct of human communications. If machines could recognize such information from a human face, humans and machines might thereby communicate more smoothly, robustly, and harmoniously.

In recent years, a tremendous amount of research has been carried out for automatic recognition of facial expressions (such as, joy, anger, sadness, fear, disgust, surprise etc.) from video sequence and still there is significant potential for further research and development. This coupled with the vast array of commercial applications (e.g. in medical system and in psychological research) makes it an attractive area of research.

The facial expressions can be changed by several ways. Figure 1.1 shows the sources for facial expressions [1].

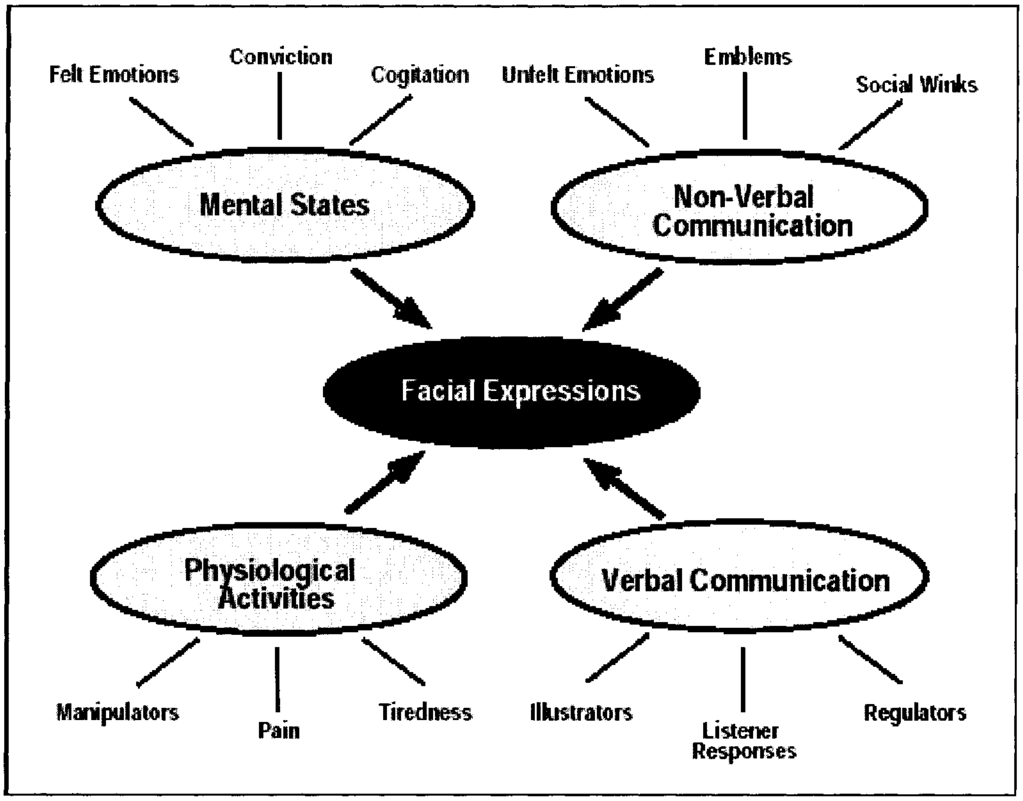


Fig. 1.1. Sources of facial expressions

In this thesis, we propose a method for automatically inferring pain in video sequences and treat the system as a special type of facial expression recognition system.

The pain recognition techniques can be subdivided in two categories – electroencephalography (EEG) signal-based and image-based.

In the first method, the current that flows during synaptic excitations of the dendrites of many pyramidal neurons in the cerebral cortex during pain are measured and used for pain or expression classification.

In the image-based method, the face images, collected either from static images or from video sequences are used for pain recognition. This approach can be further subdivided into appearance-based and model-based methods. The appearance-based approaches represent an object in terms of several object views. An image is considered as a high-dimensional vector, i.e. a point in a high-dimensional vector space. Many view-based approaches use statistical techniques to analyze the distribution of the object image vectors in the vector space, and derive an efficient and effective representation (feature space) according to different applications. Given a test image, the similarity between the stored prototypes and the test view is then carried out in the feature space. This image vector representation allows the use of learning techniques for the analysis and for the synthesis of images.

Facial features extraction methods can be categorized according to whether they focus on motion or deformation of faces and facial features, respectively whether they act locally or holistically. Holistic feature processing means the face is processed as a whole. Local feature processing means processing the features that are prone to change, with facial expressions.

Also facial features can be subdivided into transient and intransient. Intransient facial features are always present in the face, but may be deformed due to the facial expressions. Among these, the eyelids, eyebrows and the mouth are involved in the facial expressions. Tissue texture, facial hair as well as permanent furrows constitute other types of intransient facial features that influence the appearance of facial expressions. Transient facial features encompass different kind of wrinkles and bulges that occur with facial expressions. Especially the forefront and the frontal area regions surrounding the mouth and the eyes are prone to transient facial features. Opening and closing of eyes and the mouth may furthermore lead to iconic changes to texture that cannot be predicted from antecedent frames.

We will focus our research on developing a pain recognition scheme that does not depend on excessive geometry and computations like deformable templates. Instead, some linear appearance-based methods will be used. We will process the facial features both holistically and locally.

1.2 Scope of the Thesis

In this thesis, the performances of two machine learning approaches have been studied for automatic pain recognition from video sequences. The two approaches are eigenimage and multilayer neural network methods.

We have used a database of painful and neutral video files. In this database, there are 68 video files of 34 persons with different colors, ethnicities, ages and genders. In one file, the person displays a neutral facial expression and in the other file, the person displays pain. The individuals in the videos were all people who had shoulder problems and participated in an experiment in which pain was produced by manipulation of the affected shoulder.

We have used skin color modeling technique for face detection. After that, features are extracted from detected face portions. At last the pain recognition will be performed by a set of recognizers. Two approaches - eigenimage method and multi-layer neural network will be used to train the recognizers. This will allow us to compare the computational time and accuracy of the two methods. The eigenimage method will process the facial features holistically and the neural network-based pain recognizer will process the facial features locally. The simplified block diagram of our proposed system is shown in figure 1.2.

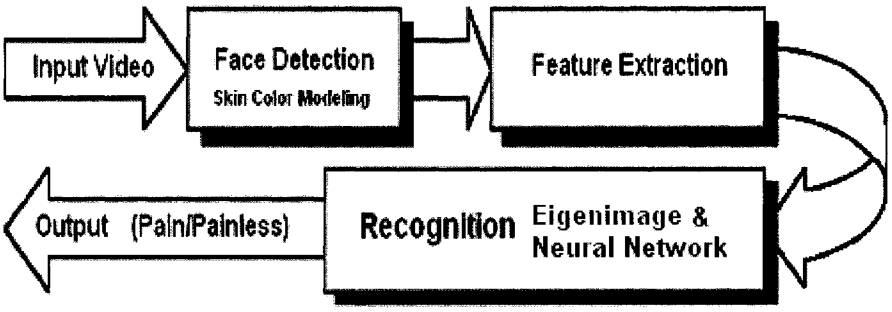


Fig. 1.2 Block diagram of the proposed pain recognition system

1.4 Outline of the Thesis

The rest of this thesis is organized as follows:

Chapter 2 introduces the literature for the machine recognition research. The basic differences between human recognition and machine recognition are reported in this chapter. Also various methods for machine recognition and pattern classifications are discussed.

Chapter 3 introduces some face detection basics and how the system learns to discriminate face and non-face examples from each other. In other words, how the face is detected in this research using skin color modeling technique is illustrated in this chapter.

Chapter 4 describes the two machine learning approaches. First, the Eigenimage method is described in detail. Then the neural network classifier is discussed.

Chapter 5 describes the simulation results for comparing the two machine learning methods for pain recognition. Speed performance and accuracy are compared between the Eigenimage method and neural network-based classifier.

Chapter 6 summarizes the conclusions of this thesis and gives some future directions for further research.

Chapter 2

Theoretical Background and Previous Work

2.1 Introduction

Recognition of facial expressions has been an interesting issue for both neuroscientists and computer engineers dealing with artificial intelligence (AI). A healthy human can detect and identify a face easily and then can recognize expression from that face, whereas for a computer to recognize expression, the face area should be detected first, and recognition comes next.

Hence, for a computer to recognize expressions from faces the photographs or video should be taken in a controlled environment; a uniform background and identical poses makes the problem easier to solve. These face images are called mug shots [2]. From these mug shots, canonical face images can be manually or automatically produced by some preprocessing techniques like cropping, rotating, histogram equalization and masking.

Image-based pain recognition system is similar to the facial expressions recognition system. Unlike most of the expression recognition systems, it does not classify joy, sadness, disgust, anger, fear and surprise expressions of faces, instead it recognizes pain in the face.

In this chapter we will look at the human vs. machine recognition of faces ad facial expression recognition and the research that has been done in this field by previous researchers.

2.2 Human Recognition of Facial Expressions

When building artificial facial expression recognition systems, scientists need to understand the architecture of the human facial expression recognition system. Focusing on the methodology of human expression recognition system may be useful to understand the basic system. However, the human expression recognition system utilizes more than just 2-dimensional data. The human facial expression recognition system uses data obtained from some or all of the senses. All these pieces of data are used either individually or collectively for storage and remembering of faces. In many cases, the surroundings also play an important role in human facial expression recognition system. It is hard for a machine recognition system to handle so much data and their combinations. However, it is also hard for a human to remember many faces due to storage limitations. A key potential advantage of a machine system is its memory capacity [3], whereas for a human facial expression recognition system the important feature is its parallel processing capacity.

Both holistic and feature information are important for the human facial expression recognition system. Studies suggest the possibility of global descriptions serving as a front end for better feature-based perception [3]. If there are dominant features present such as big ears and a small nose, holistic descriptions may not be used. Also, recent studies show that an inverted face (i.e. all the intensity values are subtracted from 255 to obtain the inverse image in the grey scale) is much harder to recognize than a normal face [4].

Eyes, mouth and face outline have been determined to be more important than the nose for perceiving and remembering faces and recognizing expressions. It has also been found that the eye, eyebrow region and the mouth region of the face are more useful than the other parts of the face for expression recognition [121].

For humans, expressions from photographic negatives of faces are difficult to recognize. But, there is not much study on why it is difficult to recognize expression from negative images of human faces. Also, a study on the direction of illumination [4] showed the importance of top lighting; it is easier for humans to recognize faces illuminated from top to bottom than the faces illuminated from bottom to top.

In the next section, we will discuss the previous works on machine recognition of facial expressions. We also make comparison between human and machine recognition of facial expressions.

2.3 Machine Recognition of Facial Expressions

Although studies on human recognition of facial expressions were expected to be a reference on machine recognition of facial expressions, research on machine recognition of facial expressions has been developed independent of studies on human recognition of facial expressions. During the 1970's, typical pattern classification techniques, which use measurements between features in faces or face profiles, were used [5]. During the

1980's, work on face recognition remained nearly stable. Since the early 1990's, research interest in machine recognition of faces and facial expressions has grown tremendously. The reasons for that are:

1. an increase in emphasis on civilian/commercial research projects,
2. the studies on neural network classifiers with emphasis on real time,
3. computation and adaptation,
4. the availability of real time hardware and
5. the growing need for surveillance and robotics applications.

The basic question relevant for expression classification is what form the structural code (for encoding the facial features) should take to achieve face recognition. Two major approaches are used for machine identification of human faces; geometrical local feature based methods, and holistic template matching based systems. Also, combinations of these two methods, namely hybrid methods, are used. The first approach, the geometrical local feature based one, extracts and measures discrete local features (such as eye, nose, mouth, hair, etc.) for retrieving and identifying faces. Then, standard statistical pattern recognition techniques and/or neural network approaches are employed for matching faces using these measurements [6]. One of the well known geometrical-local feature based methods is the Elastic Bunch Graph Matching (EBGM) technique. The other approach, the holistic one, conceptually related to template matching, attempts to identify faces using global representations [7]. Holistic methods approach the face image as a whole and try to extract features from the whole face region. In this approach, as in the previous approach, the pattern classifiers are applied to classify the image after extracting the features. One of the methods to extract features in a holistic system is applying statistical methods such as Principal Component Analysis (PCA) to the whole image. PCA can also be applied to a face image locally; in that case the approach is not holistic.

Whichever method is used, the most important problem in face recognition is the problem of dimensionality. Appropriate methods should be applied to reduce the dimension of the studied space. Working on higher dimensions causes over fitting where the system starts to memorize. Also, computational complexity would be an important problem when working on large databases.

In the following sections, the main studies will be summarized. The recognition techniques are grouped as statistical and neural based approaches. The next section discusses statistical approaches, while section 2.3.2 discusses neural-based approaches.

2.3.1 Statistical Approaches

Statistical methods include template matching based systems where the training and test images are matched by measuring the correlation between them. Moreover, statistical methods include the projection based methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). In fact, projection based systems came out due to the shortcomings of the straightforward template matching based approaches; that is, trying to carry out the required classification task in a space of extremely high dimensionality.

Template Matching: Brunelli and Poggio [8] suggested that the optimal strategy for face image analysis is holistic and corresponds to template matching. In their study, they compared a geometric feature based technique with a template matching system. In the simplest form of template matching, the image (as 2-D intensity values) is compared with a single template representing the whole face using a distance metric.

Although recognition by matching raw images has been successful under limited circumstances, it suffers from the usual shortcomings of straightforward correlation-based approaches, such as sensitivity to face orientation, size, variable lighting conditions, and noise. The reason for this vulnerability of direct matching methods lies in their attempt to carry out the required classification in a space of extremely high dimensionality. In order to overcome the problem of dimensionality, the connectionist equivalent of data compression methods is employed first. However, it has been argued that the resulting feature dimensions do not necessarily retain the structure needed for classification, and that more general and powerful methods for feature extraction such as projection based systems are required. The basic idea behind projection based systems is to construct low dimensional projections of a high dimensional point cloud, by maximizing an objective function such as the deviation from normality.

2.3.1.1 Face Recognition by PCA

The Eigenface Method of Turk and Pentland [9] is one of the main methods applied in the literature which is based on the Karhunen-Loeve expansion. Their study is motivated by the earlier work of Sirowich and Kirby [10] [11]. It is based on the application of Principal Component Analysis to the human face. It treats the face images as 2-D data, and classifies the face images by projecting them to the eigenface space which is composed of eigenvectors obtained by the variance of the face images.

Eigenface recognition derives its name from the German prefix *eigen*, meaning own or individual. The Eigenface method of facial recognition is considered the first working facial recognition technology [12].

When the method was first proposed by Turk and Pentland [9], they worked on the image as a whole. Also, they used Nearest Mean classifier to classify the face images. By using the observation that the projection of a face image and non-face image are quite different, a method of detecting the face in an image is obtained. They applied the method on a database of 2500 face images of 16 subjects, digitized at all combinations of 3 head orientations, 3 head sizes and 3 lighting conditions. They conducted several experiments to test the robustness of their approach to illumination changes, variations in size, head orientation, and the differences between training and test conditions. They reported that the system was fairly robust to illumination changes, but degrades quickly as the scale changes [9]. This can be explained by the correlation between images obtained under different illumination conditions; the correlation between face images at different scales is rather low. The eigenface approach works well as long as the test image is similar to the training images used for obtaining the eigenfaces.

Later, derivations of the original PCA approach are proposed for different applications.

PCA and Image Compression: In their study, Moghaddam and Pentland [13] used the Eigenface Method for image coding of human faces for potential applications such as video telephony, database image compression and face recognition.

Face Detection and Recognition Using PCA: Lee et al. [14] proposed a method using PCA which detects the head of an individual in a complex background and then recognizes the person by comparing the characteristics of the face to those of known individuals.

PCA Performance on Large Databases: Lee et al. [15] proposed a method for generalizing the representational capacity of available face database.

PCA & Video: In a study by Crowley and Schwerdt [16], PCA was used for coding and compression for video streams of talking heads. They suggest that a typical video sequence of a talking head can often be coded in less than 16 dimensions.

Bayesian PCA: Another method, which is also studied throughout this thesis, is the Bayesian PCA method suggested by Moghaddam et al. [17] [18] [19]. By this system, the Eigenface method based on simple subspace-restricted norms is extended to use a probabilistic measure of similarity. Also, another difference from the standard Eigenface approach is that this method uses the image differences in the training and test stages. The difference of each image belonging to the same individual with each other is fed into the system as intrapersonal difference, and the difference of one image with an image from different class is fed into the system as extra-personal difference. Finally, when a test image comes, it is subtracted from each image in the database and each difference is fed into the system. For the biggest similarity (i.e. smallest difference) with one of the training images, the test image is decided to be in that class. The mathematical theory is mainly studied by B. Moghaddam, and A. Pentland [20].

Also, in [21] Moghaddam introduced his study on several techniques; Principal Component Analysis (PCA), Independent Component Analysis (ICA), and nonlinear Kernel PCA (KPCA). He examined and tested these systems using the Facial Recognition Technology (FERET) Database. He argued that the experimental results demonstrate the simplicity, computational economy and performance superiority of the Bayesian PCA method over other methods.

Finally, Liu and Wechsler [22] [23] worked on a Bayesian approach to face recognition.

PCA and Gabor Filters: Chung et al. [24] suggested the use of PCA and Gabor Filters (linear filters whose impulse responses are defined by harmonic function multiplied by a Gaussian function) together. Their method consists of two parts: In the first part, Gabor Filters are used to extract facial features from the original image on predefined fiducial points. In the second part, PCA is used to classify the facial features optimally. They suggested the use of combining these two methods in order to overcome the shortcomings of PCA. They argue that, when raw images are used as a matrix of PCA, the eigenspace cannot reflect the correlation of facial feature well, as original face images have deformation due to in-plane, in-depth rotation and illumination and contrast variation. Also they argued that, they have overcome these problems using Gabor Filters in extracting facial features.

2.3.1.2. Face Recognition by LDA

Etemad and Chellappa [25] proposed a method on applying of Linear/Fisher Discriminant Analysis for the face recognition process. LDA is carried out via scatter matrix analysis. The aim is to find the optimal projection which maximizes between class scatter of the face data and minimizes within class scatter of the face data. As in the case of PCA, where the eigenfaces are calculated by the eigenvalue analysis, the projections of LDA are calculated by the generalized eigenvalue equation.

Subspace LDA: An alternative method which combines PCA and LDA has also been studied [26] [27] [28] [29]. This method consists of two steps; the face image is projected into the eigenface space which is constructed by PCA, and then the eigenface space projected vectors are projected into the LDA classification space to construct a linear classifier. In this method, the choice of the number of eigenfaces used for the first step is critical; the choice enables the system to generate class separable features via LDA from the eigenface space representation. The generalization/over fitting problem can be solved in this manner. In these studies, a weighted distance metric guided by the LDA eigenvalues was also employed to improve the performance of the subspace LDA method.

2.3.1.3. Transformation-based Systems

There are three types of transformation-based system. One of them uses Discrete Cosine Transform, one uses a combination of Pseudo 2-dimentional Hidden Markov Models (HMMs) and Discrete Cosine Transform and the other uses Fourier Transform.

DCT: Podilchuk and Zang [30] proposed a method which finds the feature vectors using Discrete Cosine Transform (DCT). Their system tries to detect the critical areas of the face. The system is based on matching the image to a map of invariant facial attributes associated with specific areas of the face. They claim that this technique is quite robust, since it relies on global operations over a whole region of the face. A codebook of feature vectors or codewords is determined for each person from the training set. They examined recognition performance based on feature selection, number of features or codebook size, and feature dimensionality. For feature selection, they tried several block-based transformations and the K-means clustering algorithm [31] to generate the codewords for each codebook. They argued that the block-based DCT coefficients produce good low-dimensional feature vectors with high recognition performance. This brings the possibility of performing face recognition directly on a DCT-based compressed bitstream without having to decode the image.

DCT & HMMs: Eickeler et al. [32] suggested a system based on Pseudo 2-D Hidden Markov Models (HMMs) and coefficient of the 2-D DCT as the features. A major advantage of their approach is that the system works directly on JPEG-compressed face images, i.e. it uses the DCT-coefficients provided by the JPEG standard. Thus, it does not need any further decompressing of the image. Also Nefian and Hayes [33] used DCT and HMMs as their feature vectors in their research.

Fourier Transform (FT): Spies and Ricketts [34] describe a face recognition system based on an analysis of faces via their Fourier spectra. Recognition is achieved by finding the closest match between feature vectors containing the Fourier coefficients at selected frequencies. This technique is based on the Fourier spectra of facial images, thus it relies on a global transformation, i.e., every pixel in the image contributes to each value of its spectrum. The Fourier spectrum is a plot of the energy against spatial frequencies, where spatial frequencies relate to the spatial relations of intensities in the image. In the case of face recognition, this translates to distances between areas of particular brightness, such as the overall size of the head, or the distance of the eyes. Higher frequencies describe finer details and they claimed that these are less useful for identification of a person. They also suggested that, humans can recognize a face from a brief look without focusing on small details. They perform the recognition of faces by finding the Euclidian distance between a newly presented face and all the training faces. The distances are calculated between feature vectors with entries that are the Fourier transform values at specially chosen frequencies. They argue that, as few as 27 frequencies yield good results (98 %). Moreover, this small feature vector combined with the efficient Fast Fourier Transform (FFT) makes this system extremely fast.

2.3.1.4. Face Recognition by SVM

Phillips [35] applied Support Vector Machines (SVM) to face recognition. Face recognition is a K-class problem, where K is the number of known individuals; and SVM is a binary classification method. By reformulating the face recognition problem and reinterpreting the output of the SVM classifier, they developed a SVM-based face recognition algorithm. They formulated the face recognition problem in difference space, which models dissimilarities between two facial images. In difference space, they formulated the face recognition as a two class problem. The classes are; dissimilarities between faces of the same person and dissimilarities between faces of different people. By modifying the interpretation of the decision surface generated by SVM, they generated a similarity metric between faces, learned from examples of differences between faces [35].

2.3.1.5. Feature-based Approaches

Bobis et al. [36] developed a feature based face recognition system. They suggested that a face can be recognized by extracting the relative position and other parameters of distinctive features such as eyes, mouth, nose and chin. The system described the overall geometrical configuration of face features by a vector of numerical data representing position and size of main facial features. First, they extracted eyes coordinates. The interocular distance and eye position is used to determine the size and the position of the areas of search for face features. In these areas, binary thresholding is performed by modifying the threshold automatically to detect features. In order to find their coordinates, discontinuities are searched in the binary image. They claimed that their experimental results showed that their method is robust, valid for numerous kind of facial image in real scene, works in real time with low hardware requirements and the whole process is conducted automatically.

Cagnoni and Poggi [37] suggested a feature-based approach in which they applied the eigenface method to sub-images (eye, nose, and mouth). They also applied a rotation correction to the faces in order to obtain better results.

Guan and Szu [38] compared the performance of PCA and ICA on face images. They argued that ICA encodes face images with statistically independent variables, which are not necessarily associated with the orthogonal axis, while PCA is always associated with orthogonal eigenvectors. While PCA seeks directions in feature space that best represent the data in a sum-squared error sense, ICA seeks directions that are most independent from each other. They also argued that both these pixel-based algorithms have the major drawback that they weight the whole face equally and therefore lack the local geometry information. Hence, Guanand Szu suggested that approaching the face recognition problem with ICA or PCA applied on local features.

Martinez [39] proposed a different approach based on identifying frontal faces. Their approach divides a face image into *n* different regions, analyzes each region with PCA, and then uses a Bayesian approach to find the best possible global match between a probe and database image. The relationship between the n parts is modeled by using Hidden Markov Models (HMMs).

2.3.2 Neural Network Approaches

Neural Network approaches have been used in face and facial expression recognition generally in a geometrical local feature based manner, but there are also some methods where neural networks are applied holistically. In the next section we will discuss three types of neural network-based recognition of faces and facial expressions.

Feature based Backpropagation NN: Temdee et al. [40] presented a frontal view face recognition method by using fractal codes which are determined by a fractal encoding method from the edge pattern of the face region (using eyebrows, eyes and nose). In their recognition system, the obtained fractal codes are fed as inputs to a Backpropagation Neural Network for identifying an individual. They tested their system performance on the ORL (Olivetti Research Laboratory) face database. They reported their performance as 85 % correct recognition rate in the ORL face database.

Dynamic Link Architectures (DLA): Lades et al. [41] presented an object recognition system based on Dynamic Link Architectures, which is an extension of the Artificial Neural Networks. The DLA uses correlations in the fine-scale cellular signals to group neurons dynamically into higher order entities. These entities can be used to code high-level objects, such as a 2-D face image. The face images are represented by sparse graphs, whose vertices are labeled by a multi-resolution description in terms of local power spectrum, and whose edges are labeled by geometrical distance vectors. Face recognition can be formulated as elastic graph matching, which is performed in this study by stochastic optimization of a matching cost function.

Elastic Bunch Graph Matching (EBGM): Wiskott et al. [42] presented a geometrical local feature based system for face recognition from single images out of a large database containing one image per person, which is known as Elastic Bunch Graph Matching (EBGM). In this system, faces are represented by labeled graphs, based on a Gabor Wavelet Transform (GWT). Image graphs of new faces are extracted by an Elastic Graph Matching process and can be compared by a simple similarity function. In this system, phase information is used for accurate node positioning and object-adapted graphs are used to handle large rotations in depth. The image graph extraction is based on the bunch graph, which is constructed from a small set of sample image graphs. In contrast to many neural-network systems, no extensive training for new faces or new object classes is required. Only a small number of typical examples have to be inspected to build up a bunch graph, and individuals can then be recognized after storing a single image.

The system inhibits most of the variance caused by position, size, expression and pose changes by extracting concise face descriptors in the form of image graphs. In these image graphs, some predetermined points on the face (eyes, nose, mouth, etc.) are described by sets of wavelet components (jets). The image graph extraction is based on the bunch graph, which is constructed from a small set of image graphs.

2.3.3 Hybrid Approaches

There are some other approaches which use both statistical pattern recognition techniques and Neural Network systems.

Er et al. [43] worked on the use of Radial Basis Function (RBF) Neural Networks on the data extracted by discriminant eigenfeatures. They used a hybrid learning algorithm to decrease the dimension of the search space in the gradient method, which is crucial on optimization of high dimension problem. First, they tried to extract the face features by both the PCA and LDA methods. Next, they presented a hybrid learning algorithm to train the RBF Neural Networks, so the dimension of the search space is significantly decreased in the gradient method.

Thomaz et al. [44] also developed a system by combining PCA and RBF neural network. Their system is a face recognition system consisting of a PCA stage which inputs the projections of a face image over the principal components into a RBF network acting as a classifier. Their main concern is to analyze how different network designs perform in a PCA+RBF face recognition system. They used a forward selection algorithm, and a Gaussian mixture model. According to the results of their experiments, the Gaussian mixture model optimization achieves the best performance even using less neurons than the forward selection algorithm. Their results also show that the Gaussian mixture model design is less sensitive to the choice of the training set.

2.3.4 Other Issues

Besides statistical and neural network approaches, there are some other approaches, e.g., range data, infrared scanning and profile images methods, used in facial expressions recognition system development. In the next section, those two methods will be discussed.

In the ***range data method***, range images are used for recognition. In this method data is obtained by scanning the individual with a laser scanner system. This system also has the depth information so the system processes 3-dimensional data to classify face images [5].

The ***infrared scanning method*** scans the face image by an infrared light source. Yoshitomi et al. [45] used thermal sensors to detect temperature distribution of a face. In this method, the front-view face in input image is normalized in terms of location and size, followed by measuring the temperature distribution, the locally averaged temperature and the shape factors of the face. The measured temperature distribution and the locally averaged temperature are separately used as input data to feed a Neural Network, while the values of shape factors are used for supervised classification. By integrating information from the Neural Network and supervised classification, the face is identified. One disadvantage of visible ray image analysis is that the accuracy of face identification is strongly influenced by lighting conditions including variation of shadow, reflection and darkness.

The ***profile images approach*** was first introduced by Liposcak and Loncaric [46]. This method is based on the representation of the original and morphological derived profile shapes. Their aim is to use the profile outline that bounds the face and the hair. They take a grey-level profile image, threshold it to produce a binary image, representing the face region. They normalize the area and orientation of this shape using dilation and erosion. Then, they simulate hair growth and haircut and produce two new profile silhouettes. From these three profile shapes they obtain the feature vectors. After normalizing the vector components, they use the Euclidean distance measure for measuring the similarity of the feature vectors derived from different profiles.

2.4 Previous Research on Expression Recognition

Since the early 1970s, Paul Ekman and his colleagues [47] have performed extensive studies of human facial expressions. They found evidence to support universality in facial expressions. These ‘‘universal facial expressions’’ are those representing happiness, sadness, anger, fear, surprise, and disgust. They studied facial expressions in different cultures, including preliterate cultures, and found much commonality in the expression and recognition of emotions on the face. However, they observed some differences in expressions as well, and proposed that facial expressions are governed by ‘‘display rules’’ in different social contexts. For example, Japanese subjects and American subjects showed similar facial expressions while viewing the same stimulus film. However, in the presence of authorities, the Japanese viewers were more reluctant to show their real expressions. Babies seem to exhibit a wide range of facial expressions without being taught, thus suggesting that these expressions are innate [48].

Ekman and Friesen [49] developed the Facial Action Coding System (FACS) in 1976 to code facial expressions where movements on the face are described by a set of action units (AUs). There are 46 action units and expression is defined as one of these action units, which is a contraction or relaxation of one or more muscles. For example, it can be used to distinguish the two types of smiles as follows:

- insincere and voluntary Pan American smile: contraction of zygomatic major (a muscle of facial expression which draws the angle of the mouth superiorly and posteriorly) alone
- sincere and involuntary Duchenne smile: contraction of zygomatic major and inferior part of orbicularis oculi (arises from the nasal part of the frontal bone,

from the frontal process of the maxilla in front of the lacrimal groove, and from the anterior surface and borders of a short fibrous band, the medial palpebral ligament).

Each AU has some related muscular basis. This system of coding facial expressions is done manually by following a set of prescribed rules. The inputs are still images of facial expressions, often at the peak of the expression.

A constraint in the development of FACS was that it deals with what is clearly visible in the face, ignoring invisible changes (e.g. certain changes in muscle tonus), and discarding visible changes too subtle for reliable distinction.

FACS excludes visible changes in muscle tonus which do not entail movement; changes in skin coloration are usually not visible on black and white records. Also excluded from FACS are: facial sweating, tears, rashes, pimples and permanent facial characteristics.

The user of FACS must learn the mechanics -- the muscular basis -- of facial movement, not just the consequence of movement or a description of a static landmark. FACS emphasizes patterns of movement, the changing nature of facial appearance. Distinctive actions are described: the movements of the skin, the temporary changes in shape and location of the features, and the gathering, pouching, bulging and wrinkling of the skin.

Although the labeling of expressions currently requires trained experts, researchers have had some success in using computers to automatically identify FACS codes, and thus quickly identify emotions.

In spite of its high time-consumeness and these constraints, it is the most popular standard currently used to systematically categorize the physical expression of emotions, and it has proven useful both to psychologists and to animators. This system of coding facial expressions is done manually by following a set of prescribed rules. The inputs are still images of facial expressions, often at the peak of the expression.

Ekman's work inspired many researchers to analyze facial expressions by means of image and video processing. By tracking facial features and measuring the amount of facial movement, they attempted to categorize different facial expressions. Recent work on facial expression analysis and recognition [50-61] has used these ''basic expressions'' or a subset of them. In [62], Pantic and Rothkranz provide an in depth review of many of the researches done in automatic facial expression recognition in recent years.

The work in computer-assisted quantification of facial expressions did not start until the 1990s. Mase [56] used optical flow (OF) to recognize facial expressions. He was one of the first to use image processing techniques to recognize facial expressions. Lanitis et al. [53] used a flexible shape and appearance model for image coding, person identification, pose recovery, gender recognition, and facial expression recognition. Black and Yacoob [50] used local parameterized models of image motion to recover non-rigid motion. Once recovered, these parameters were used as inputs to a rule-based classifier to recognize the six basic facial expressions. Yacoob and Davis [63] computed optical flow and used similar rules to classify the six facial expressions. Rosenblum et al. [60] also computed optical flow of regions on the face, then applied a radial basis function network to classify expressions. Essa and Pentland [52] used an optical flow region-based method to recognize expressions. Donato et al. [51] tested different features for recognizing facial AUs and inferring the facial expression in the frame. Otsuka and Ohya [59] first computed optical flow, then computed the 2D Fourier transform coefficients, which were used as feature vectors for a hidden Markov model (HMM) to classify expressions. The trained system was able to recognize one of the six expressions near real-time (about 10 Hz). Furthermore, they used the tracked motions to control the facial expression of an animated Kabuki system [64]. A similar approach, using different features, was used by Lien [54]. Nefian and Hayes [57] proposed an embedded HMM approach for face recognition that uses an efficient set of observation vectors based on the DCT coefficients. Martinez [55] introduced an indexing approach based on the identification of frontal face images under different illumination conditions, facial expressions, and occlusions. A Bayesian approach was used to find the best match between the local observations and the learned local features model and an HMM was employed to achieve good recognition even when the new conditions did not correspond to the conditions previously encountered during the learning phase. Oliver et al. [58] used lower face tracking to extract mouth shape features and used them as inputs to an HMM based facial expression recognition system (recognizing neutral, happy, sad, and an open mouth). Hok-chun Lo and Ronald Chung used Eigenface, first introduced by M. Turk and A. Pentland in 1991 and later modified by many researchers, for facial expression recognition in 2003.

In 2004, Jeffrey Cohn of University of Pittsburgh and T. Kanade of Carnegie Mellon University [123] developed an automated facial image analysis system for automatic recognition of facial action units and quantitative analysis of their dynamics, such as timing. In their system, the changes in both permanent (e.g., brows) and transient (e.g., furrows) facial features are automatically detected and tracked offline throughout the image sequence. Using FACS,. they grouped facial features into separate collections of feature parameters. These parameters include feature displacement, velocity, and appearance. The extracted facial feature and head motion trajectories are fed to a classifier for action unit recognition. In addition to recognizing action units, the system quantified the timing of facial actions and head gesture for studies of timing of facial actions.

All of these methods are similar in that they first extract some features from the images, then these features are used as inputs into a classification system, and the outcome is one of the pre-selected emotion categories. They differ mainly in the extracted features of the video images and in the classifiers used to distinguish different emotions.

In this study, we also followed this strategy. First, we extract features from the video images and then fed them into classifiers. We did not use FACS in our research. In stead, we have use PCA and the local face features for our feature extraction. As a recognizer, we used Eigenimage and the back-propagated neural networks.

2.5 Conclusion

In this chapter, we have discussed the various methods of facial expressions recognition. Also, we have discussed the previous researches on this field. Some of these works are good for all types of images and some of them are not good for low intensity images. Some researchers recognized expressions by using some action units of the faces.

In our research, we have not used any action units but have considered the eye and lip portions of a face because these regions of face are most sensitive to pain. For learning and recognition, we have used eigenimage method and neural network approaches due to their simplicity and accuracy.

One other important issue of any facial expression recognition system is the face detection. In the next section, we will discuss various face detection techniques and illustrate our face detection system using skin color modeling.

Chapter 3

Face Detection

3.1 Introduction

Computer vision, in general, aims to duplicate (or in some cases compensate) human vision, and traditionally, has been used in performing routine, repetitive tasks, such as classification in massive assembly lines. Today, research on computer vision is spreading enormously so that it is almost impossible to itemize all of its subtopics. Despite this fact, one can list several relevant applications, such as face processing (i.e. face, expression, and gesture recognition), computer human interaction, crowd surveillance, and content-based image retrieval. All of these applications require face detection, which can be simply viewed as a preprocessing step, for obtaining the "object". In other words, many of the techniques that are proposed for these applications assume that the location of the face is pre-identified and available for the next step.

Face detection is one of the tasks which human vision can do effortlessly. However, for computer vision, this task is not that easy. A general definition of the problem can be stated as follows: Identify all of the regions that contain a face, in a still image or image sequence, independent of any three dimensional transformation of the face and lighting condition of the scene. There are several methods issued for this problem and they can be broadly classified in two main classes, which are *feature-based*, and *image-based* approaches. Previous research has shown that both feature-based and image-based approaches perform effectively while detecting upright frontal faces, whereas feature-based approaches show a better performance for the detection scenarios especially in simple scenes.

Face detection is the problem of determining whether a sub-window of an image contains a face. From the point of view of learning, any variation which increases the complexity of decision boundary between face and non-face classes will also increase the difficulty of the problem. For example, adding tilted faces into the training set increases the variability of the set, and may increase the complexity of the decision boundary. Such complexity may cause the classification to be harder. There are many sources introducing variability when dealing with the face. They can be summarized as follows:

Image plane variation is the first simple variation type that one may encounter. Image transformations, such as rotation, translation, scaling and mirroring may introduce such kind of variations. Utilization of image pyramids with a sliding detector window is one common way to deal with such transformations for the input image. Variations in the global brightness, contrast level can also be expressed in the same category. Typical examples for such variations can be seen in Figure 3.1.

Pose variations can also be listed under image plane variations aspects. However, changes in the orientation of the face itself on the image can have larger impacts on its appearance. Rotation in depth and perspective transformation may also cause distortion. The common way to deal with pose variation is to isolate pose types (i.e. frontal, profile, rotated). Some examples for such pose variations are shown in Figure 3.1.

Lighting variations may dramatically change face appearance in the image. Such variations are the most difficult type to deal with due to the fact that pixel intensities are directly affected in a nonlinear way by changing illumination intensity or direction. For example, when using skin color as a feature for face detection, varying color temperature [65] of the light source may cause skin color filtering to fail. Some examples for lighting variations are shown in Figure 3.1

Background variations are another challenging factor for face detection in cluttered scenes. Discriminating windows including a face from those of non-face is more difficult when no constraints exist on background. Most of the examples shown in figure 3.1 have complex backgrounds which makes the face detection problem harder.

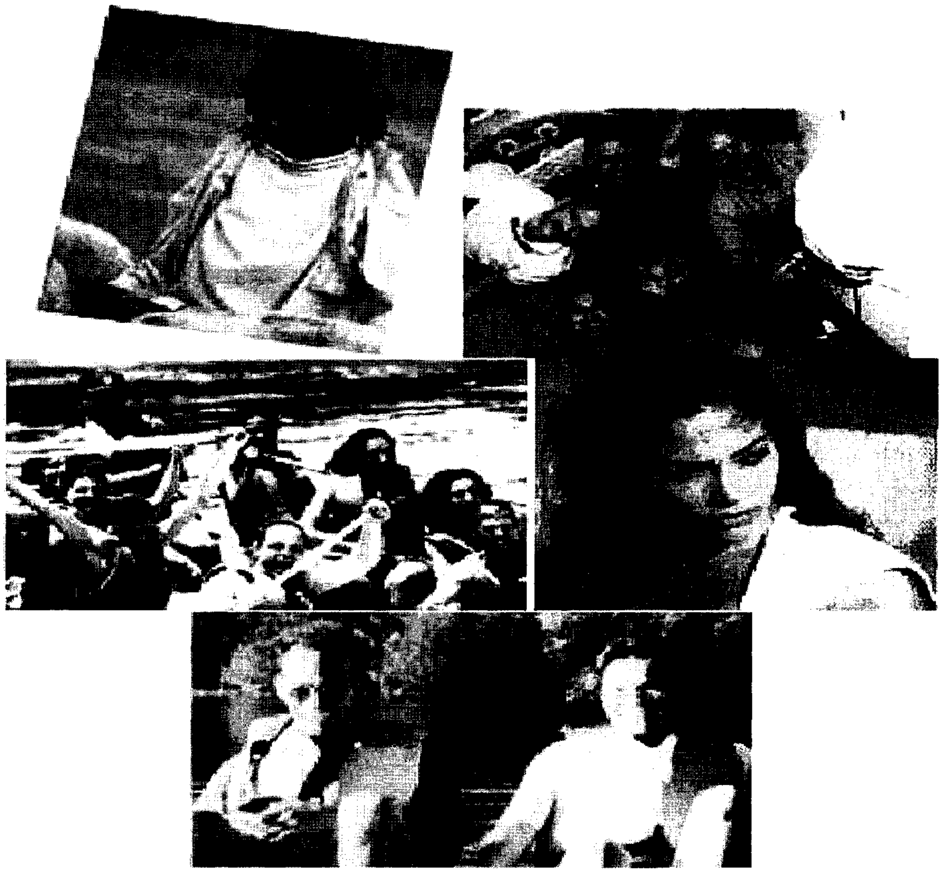


Fig. 3.1. Examples of several variations of background

In the rest of this chapter, we will give a general overview of face detection approaches in section 3.2 and then in section 3.3, we will discuss our face detection approach.

3.2 Background of Face Detection

Over the past ten years, there has been a great deal of research concerning important aspects of face detection. Using generalized face shape rules, motion, and color information many segmentation schemes have been presented [66] [67] [68]. The use of probabilistic [69] and neural network methods [70] has made face detection possible in cluttered scenes and variable scales. Face detection research can be heuristically classified in two main categories: feature-based approaches and image-based approaches.

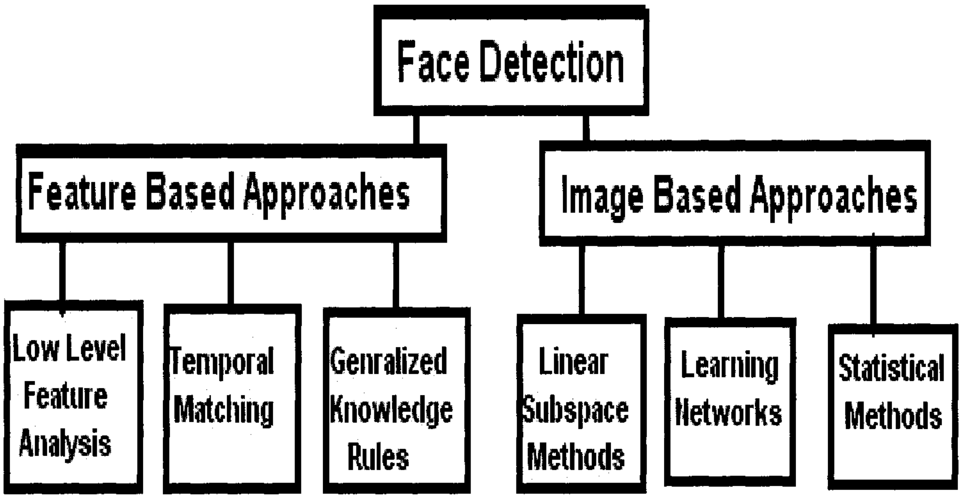


Fig. 3.2. Classification of face detection methods

According to the taxonomy in figure 3.2, feature-based methods make explicit use of face knowledge and follow the classical detection methodology, in which low level features that are used prior to analysis mostly rely on heuristics or advance templates. The apparent properties of the face, such as skin color and face geometry, are used at different levels of the system. Since features are the main ingredients, these techniques are named as the feature-based approach. These approaches [71] have embodied the majority of interest in face detection research starting as early as the 1970s. Taking advantage of current advances in pattern recognition theory, image-based approaches address face detection as a general pattern recognition problem. Partly due to well known work by K. Sung and T. Poggio [72], these approaches have attracted much attention in recent years, and have demonstrated remarkable results. According to the image-based methods, face detection is a two class (face, non-face) object recognition problem which uses pure image (intensity) representations instead of abstract feature representations.

3.2.1 Feature-Based Approaches

Most feature-based approaches share similar consecutive steps. Usually, the first step is to make pixel level eliminations by utilizing low level feature(s), e.g. skin color filtering, edge detection. Due to the low level properties, the result that is generated in the first step is ambiguous. In the second step, visual features which are not eliminated in the first step are organized within a global face knowledge or geometry. Using this feature analysis, feature ambiguities are reduced and the locations of face and facial features are determined. The final step may involve the use of templates or active shape models.

In the next section, we discuss the three feature-based approaches – low level feature analysis, template matching and generalized knowledge rules.

3.2.1.1 Low Level Feature Analysis

There are three low level features – edges, skin color and motion, that are used in face detection process. In the following section we will discuss these three low level features

Edges: As a useful primitive feature in computer vision, edge representation was applied to early face detection system by Sakai et al. [73]. Later, based on this work, a hierarchical framework was proposed by Craw et al. [74] to trace the human head line. This approach included a line follower which is implemented with a curvature constraint. Some more recent examples of edge-based techniques can be found in references [75-78].

Edge detection is the important step in edge-based techniques. For detecting edges, various types of edge detector operators are used. The Sobel operator is the most common filter among others for detecting edges [76] [79]. Also, a variety of 1st and 2nd derivatives (Laplacian) of Gaussians have been used in some approaches [73] [80]. A large scale Laplacian was used to obtain lines [73], and steer-able and multi-scale-orientation filters are preferred in [80].

Skin Color: Human skin color has been used and proven to be effective feature for face detection, and related applications. Although skin color differs among individuals, several studies have shown that the major difference exists in the intensity rather than the chrominance. Several color spaces have been used to label skin pixels including RGB [81] [82], NRGB (normalized RGB) [83-85], HSV (or HSI) [86-87], YCrCb [88], CIE-XYZ [89], CIE-LUV [66]. Although, the effectiveness of the different color spaces is arguable, common point of all above works is the removal of the intensity component.

Color segmentation can basically be performed using appropriate skin color thresholds where skin color is modeled through histograms or charts [90-91] [87]. More complex methods make use of statistical measures that model face variation within a wide user spectrum [83-84] [92-93]. For instance, Oliver et al. [84] and Yang et. al. [93] employ a Gaussian distribution to represent a skin color cluster, consisting of thousands of skin color samples, taken from the different human races. Even though color information seems to be an efficient tool for identifying facial areas, the skin color models may fail when the spectrum (correlated color temperature) of light source varies significantly.

We have also studied skin color information to utilize a skin color filter in the preprocessing step in face detection. However, in general, the skin color filters are constructed by using fixed boundaries (thresholds) for sample pixel distributions in color space. Illumination and camera parameters are omitted. Hence, the exhaustiveness in the variations for a sample pixel set may create a performance for the resulting skin color filter. The response of two skin color filters for the same color image can be seen in Figure 3.3. Note that the HSI skin color filter with fixed thresholds is unsuccessful in determining skin color pixels. On the other side, the NRGB skin color filter that is using adjustable thresholds is successful in determining skin color pixels by adding false alarms. Although, it may be more deeply experimented, we may state that a varying threshold skin color filter which includes self adaptation to image illumination properties (e.g. CCT) may result in more effective skin color filtering results.

In our research, we have used this method for face detection. More details of our implementation is given in section 3.3.

Motion: Motion information is a convenient way of locating moving objects when a video sequence is provided. It is possible to narrow face searching area utilizing this information. The simplest way to achieve motion information is frame difference analysis. Accumulated frame difference is improved frame difference analysis which is used by many reported face detection research studies [68, 94]. Besides face region, Luthon et. al. [95], also employ frame difference to locate facial features, such as eyes. Another way of measuring visual motion is through the estimation of moving image contours. Compared to frame difference, results generated from moving contours are always more reliable, especially when the motion is insignificant [96].

3.2.1.2 Template Matching

Given an input image, the correlation values in predetermined standard regions, such as face contour, eyes, nose and mouth are calculated independently. Although, this approach has the advantage of simplicity, it has been insufficient for face detection since it can not handle variations in scale, rotations pose and shape. Multi resolution, multi scale, sub-templates and deformable templates have been proposed to achieve scale and shape invariance template matching [97-98].

There are many studies which have been done on template matching. In [97], Miao et al. proposed a hierarchical template matching method for face detection. Kwon et al. [98] proposed a detection method based on *snakes* and templates. Lanitis et al.[99] established a detection method utilizing both shape and intensity information.

3. 2.1.3 Generalized Knowledge Rules

In generalized knowledge-based approaches, the algorithms are developed based on heuristics about face appearance. Although, it is simple to create heuristics for describing the face, the major difficulty is in translating these heuristics into classification rules in an efficient way. If these rules are over detailed, they may come up with missed detections; on the other hand, if they are more general they may introduce much false detection. In spite of this, some heuristics can be used at an acceptable rate in frontal faces on uncluttered backgrounds. Yang and Huang [67] used a hierarchical knowledge-based method to detect faces. Their system consists of three level rules going from general to detail. This method does not report a high detection rate, their ideas for mosaic (multi-resolution), and multiple level rules have been used in more recent methods.

3.2.2 Image-Based Approaches

In contrast to feature-based approaches, image-based approaches utilize example image representations, instead of abstract representations consisting of features. In general, image-based approaches rely on machine learning and statistical analysis. Face detection is a two class (face, non-face) classification problem, which relies on learned characteristics generally in the form of distributions. The specific need for face knowledge is avoided by formulating the problem as a learning paradigm to discriminate a face pattern from a non-face pattern.

Most of the image-based approaches apply a window scanning technique for detecting faces. The window scanning algorithm employs an exhaustive search of the input image for possible face locations at all scales, but there are variations in the implementation of this algorithm for almost all the image based systems. Typically, the size of the scanning window, the sub sampling rate, the step size, and the number of iterations vary depending on the method proposed and the need for a computationally efficient system.

In the following section, we will discuss three image-based approaches for face detection. These approaches are linear subspace methods, learning networks and statistical approaches.

3.2.2.1 Linear Subspace Methods

In the late 1980s, Sirovich and Kirby [100] developed a technique using PCA (Principal Component Analysis)to efficiently represent human faces. Given a set of face images, the proposed technique obtains the principal components of the distribution of faces, expressed in terms of eigenvectors (of the covariance matrix of the distribution). Then, each individual face in the set can be approximated by a linear combination of the largest eigenvectors (*eigenfaces*) corresponding to the largest eigenvalues, using appropriate weights.

Later, Turk and Pentland [94] improved this technique for face recognition. Their method takes advantage of the distinct nature of the weights of eigenfaces for individual face representation. Since, face reconstruction, by using its principal components, is an approximation, a residual error is defined in the algorithm as a preliminary measure of "faceness". This residual error which they termed "distance-from-face-space" (DFFS), gives an indication of face existence through the observation of global minima in the distance map.

Pentland et al. [101] later proposed a facial feature detector using DFFS, generated from eigenfeatures (*eigeneye, eigennose, eigenmouth*), which are obtained from various facial feature templates in a training set. The performance of the eye locations was reported to be 94% with 6% false positive rate in a database of 7562 frontal face images in front of a plain background.

More recently, Moghaddam and Pentland have further developed this technique within a probabilistic framework [102]. Unlike the usual PCA framework, they did not discard the orthogonal complement of face space. This leads to uniform density assumption of the face space. Hence, they developed a maximum likelihood detector which takes into account both face space and its orthogonal complement to handle arbitrary densities. They report a detection rate of 95% on a set of 7000 face images for detecting the left eye. Compared to the DFFS detector, the results were significantly better. On a task of multi scale head detection of 2000 face images from the FERET [32] database which includes mug shot faces in front of a uniform background, the detection rate was 97%.

3.2.2.2 Learning Networks

Since face detection can be understood as a two class pattern recognition problem, several neural network-based approaches have been introduced for solution. A review of neural network-based face detection methods can be found in Viennet et al. [103].

The first advanced neural network approach which reported significant results on a large, complex dataset was introduced by Rowley et al. [70]. The system incorporates face knowledge in a rationally connected neural network as shown in figure 3.3. The neural network is designed to look at windows of 20 x 20 pixels. There is one hidden layer with 26 units, where 4 hidden units connected to 10 x 10 pixel sub regions, 16 units connected to 5 x 5 sub regions, and 6 units connected to 20 x 5 pixels via input units. The input window is pre-processed through lighting correction (a best fit linear function is subtracted) and histogram equalization. This pre-processing method was adopted from Sung and Poggio's system. A major problem which arises with window scanning techniques is the overlapping detections. Rowley et al. [104] deals with this problem using the following two heuristics:

Thresholding: the number of detections in a small neighborhood surrounding the current location is counted, in the output pyramid which includes both location and scale and if this count turns out to be above a certain threshold, a face is assumed to be present at this location.

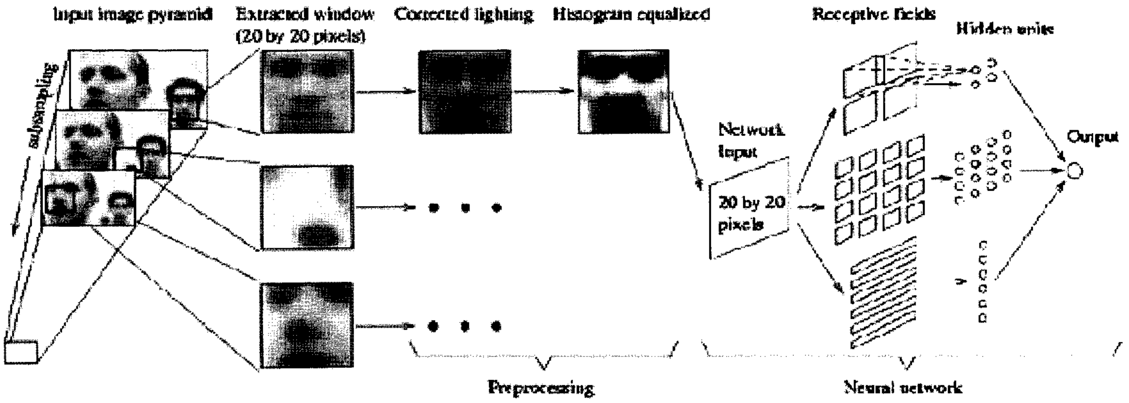


Fig 3.3. The face detection system by Rowley et al.

Overlap elimination: when a region is classified as a face by using heuristic thresholding, then overlapping detections are likely to be false positives and removed.

In Lin et al. [105], a fully automatic face recognition system is proposed based on probabilistic decision-based neural networks (PDBNN). A PDBNN is a classification neural network with a hierarchical modular structure. Instead of converting input images to a raw vector, they preferred to use features based on intensity and edge information.

Sparse Network of Winnows (SNoW), which is a new learning architecture in the visual domain, is applied to face detection by Roth et al. [106]. Similar to the previously mentioned methods, Roth et al. also use the bootstrap method of Sung and Poggio for generating training samples and preprocess all images with histogram equalization. Moreover, the window scanning technique is used in multi-scales during the evaluation stage similar to the previously mentioned methods. This method is one of the underlying methods used in this thesis, hence it will be examined in detail in the next chapter.

3.2.2.3 Statistical Approaches

There are several statistical approaches for face detection. Some of the proposed systems are based on information theory [107], a support vector machine [108] and Bayes [69] decision rule.

Colmenarez and Huang [107] proposed a system based on Kullback relative information (Kullback divergence). This divergence is a nonnegative measure between two probability density functions for a random process *Xn*. During training, for each pair of pixels in the training set, a joint-histogram is used to create probability functions for the classes of faces and non-face. Since pixel values are highly dependent on neighboring pixel values, *Xn* is treated as a first order Markov process and the pixel values in the gray-level images are re-quantized to four levels. The authors used a large set of 11 x 11 images of faces and non-face for training, and the training procedure results in a set of look-up tables with likelihood ratios. In order to further improve performance, pairs of pixels which contribute poorly to the overall divergences are dropped from the look-up tables and not used in the face detection system. This technique is further improved by including error bootstrapping, and later the technique was also incorporated in a real-time face tracking system [107].

Another major approach is Support Vector Machines (SVM) which can be considered as a new paradigm to train polynomial functions, or neural network classifiers. While most training classifiers (e.g. Bayesian, neural networks) are based on minimizing of training error *empirical risk*, SVM is based on another principle called *structural risk minimization*, which aims to minimize an upper bound on the expected generalization error. The SVM classifier is a linear classifier and its optimal hyper-plane is defined by a weighted combination of a set of training (support) vectors, which is chosen to minimize the expected classification error of the preciously unseen test patterns. Osuna et al. [109] developed an efficient method to train a SVM for large scale problems, and applied it to face detection. SVMs are also applied to the problem in the wavelet domain to detect pedestrians and faces [108]. Kumar and Poggio [110] recently incorporated Osuna et al.'s SVM algorithm in a system for real-time tracking and analysis of faces. They apply the SVM algorithm on segmented skin regions of the input images to avoid exhaustive scanning.

As another approach, Schneiderman and Kanade [111, 69] described two face detectors based on Bayes decision rule, presented as a likelihood ratio test as

$$\frac{P(image|object)}{P(image|non-object)} > \frac{P(non-object)}{P(object)}$$

If the likelihood ratio (left side) of above equation is greater than the right side, then it is decided that an object (a face) is present at the current location. The advantage of this approach is the optimality of the Bayes decision rule [112], if the representations are accurate. By the help of this approach, a view-based detector is developed with a frontal view detector and a right profile detector (to detect left profile images, the right profile detector is applied to mirror reversed images). Some examples of outputs which are processed using wavelets can be seen in figure 3.4.

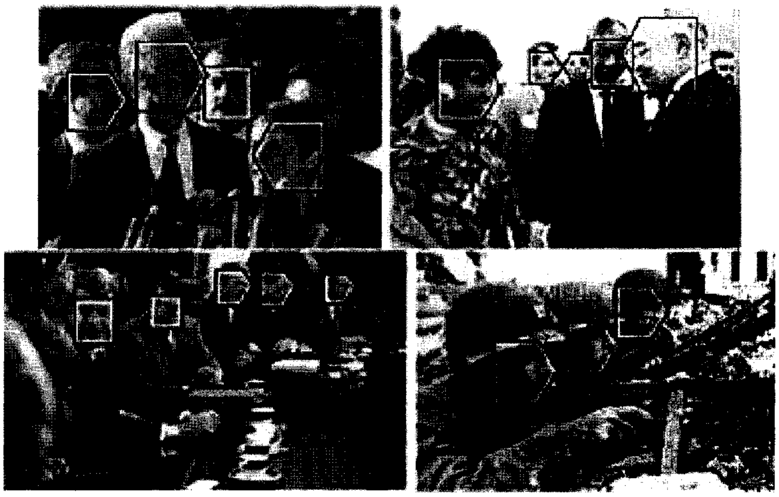


Fig. 3.4. Face detection examples from Schneiderman and Kanade

3.3 Skin Color Modeling for Face Detection

It would be fair to say that the most popular algorithm for face localization is the use of color information, whereby estimating areas with skin color is often the first vital step of such a strategy. Hence, skin color classification has become an important task. Much of the research in skin color based face localization and detection is based on RGB, YCbCr and HSI color spaces. These three color spaces are described in the following.

RGB Color Space: The RGB color space consists of the three additive primaries: red (R), green (G) and blue (B). Spectral components of these colors combine additively to produce a resultant color.

The RGB model (figure 3.5) is represented by a 3-dimensional cube with red green and blue at the corners on each axis (Figure 1). Black is at the origin. White is at the opposite end of the cube. The gray scale follows the line from black to white. In a 24-bit color graphics system with 8 bits per color channel, red is (255, 0, 0). On the color cube, it is (1, 0, 0). The RGB model simplifies the design of computer graphics systems but is not ideal for all applications. The red, green and blue color components are highly correlated. This makes it difficult to execute some image processing algorithms. Many processing techniques, such as histogram equalization, work on the intensity component of an image only.

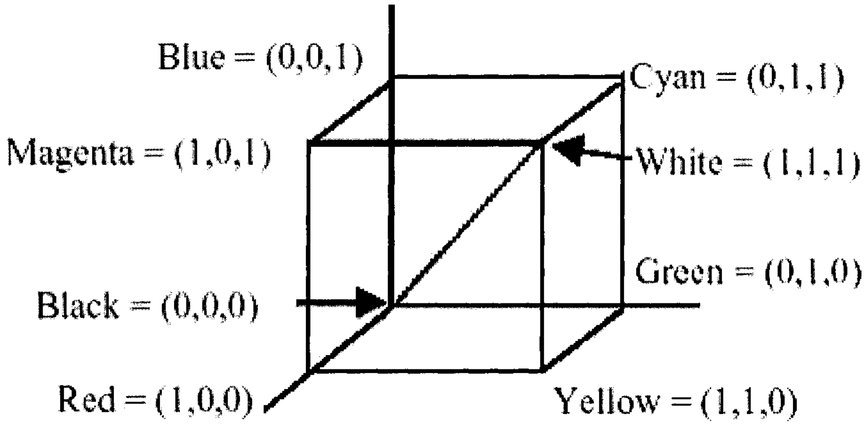


Fig. 3.5. RGB color cube

YCbCr Color Space: YCbCr color space has been defined in response to increasing demands for digital algorithms in handling video information, and has since become a widely used model in a digital video. Here Y is the luma component and Cb and Cr the blue and red chroma components.

It belongs to the family of television transmission color spaces. The family includes others such as YUV and YIQ. YCbCr is a digital color system, while YUV and YIQ are analog spaces for the respective PAL and NTSC systems. These color spaces separate RGB (Red-Green-Blue) into luminance and chrominance information and are useful in compression applications however the specification of colors is somewhat unintuitive.

The 601 recommendation specifies 8 bit (i.e. 0 to 255) coding of YCbCr, whereby the luminance component Y has an excursion of 219 and an offset of +16. This coding places black at code 16 and white at code 235. In doing so, it reserves the extremes of the range for signal processing footroom and headroom. On the other hand, the chrominance components Cb and Cr have excursions of +112 and offset of +128, producing a range from 16 to 240 inclusively.

HSI Color Space: Since hue (H), saturation (S) and intensity (I) are three properties used to describe color, it seems logical that there should be a corresponding HSI color model. When using the HSI color space, one does not need to know what percentage of blue or green is required to produce a color. One simply adjusts the hue to get the color that one wishes. To change a deep red to pink, one adjusts the saturation. To make it darker or lighter, one alters the intensity.

Many applications use the HSI color model. Machine vision uses HSI color space in identifying the color of different objects. Image processing applications such as histogram operations, intensity transformations and convolutions operate only on an intensity image. These operations are performed with much ease on an image in the HSI color space.

For HSI being modeled with cylindrical coordinates, see figure 3.6. The hue (*H*) is represented as the angle 0, varying from 0^{o} to 360^{o}. Saturation (*S*) corresponds to the radius, varying from 0 to 1. Intensity (*I*) varies along the *z* axis with 0 being black and 1 being white.

When *S* = 0, color is a gray value of intensity 1. When *S* = 1, color is on the boundary of top cone base. The greater the saturation, the farther the color is from white/gray/black (depending on the intensity). Adjusting the hue will vary the color from red at 0o, through green at 120o, blue at 240o, and back to red at 360o. When *I* = 0, the color is black and therefore *H* is undefined. When *S* = 0, the color is grayscale. *H* is also undefined in this case.

By adjusting *I*, a color can be made darker or lighter. By maintaining *S* = 1 and adjusting *I*, shades of that color are created.

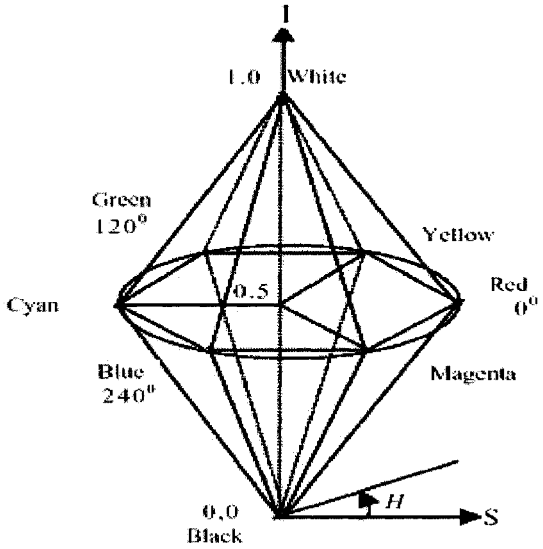


Fig 3.6. Double cone model of HSI color space

We have used RGB color space for face detection using skin color information in our research. The details of this process are given in the following section.

3.3.1 Skin Color Based Face Detection in RGB Color Space:

Crowley and Coutaz [113] argue that one of the simplest algorithms for detecting skin pixels is to use a skin color algorithm. The perceived human color varies as a function of the relative direction to the illumination. The pixels for skin region can be detected using a normalized color histogram, and can be further normalized for changes in intensity on dividing by luminance. Thus an [R, G, B] vector is converted into an [*r, g*] vector of normalized color which provides a fast means of skin detection. This gives the skin color region which localizes the face. As in [113], the output is a face detected image which is from the skin region. This algorithm fails when there are some more skin region like legs, arms, etc.

3.3.1.1 Building a Skin Color Model:

We have used almost the same technique in our implementation since the common RGB representation of color images is not suitable for characterizing skin-color. In the RGB (red, green and blue) space, the triple component (r, g, b) represents not only color but also luminance. Luminance may vary across a person's face due to the ambient lighting and is not a reliable measure in separating skin from non-skin region [114]. Luminance can be removed from the color representation in the chromatic color space. Chromatic colors [115], also known as "pure" colors in the absence of luminance, are defined by a normalization process shown below:

$$r = R/(R+G+B)$$

$$b = B/(R+G+B)$$

Color green is redundant after the normalization because r + g + b = 1. If two points $P_1[r_1,g_1,b_1]$ and $P_2[r_2,g_2,b_2]$, are proportional, i.e.,

$$\frac{r_1}{r_2} = \frac{g_1}{g_2} = \frac{b_1}{b_2}$$

then, P_1 and P_2 have the same color but different brightness.

Chromatic colors have been effectively used to segment color images in many applications [116]. It is also well suited in this case to segment skin regions from non-skin regions. The color distribution of skin colors of different people was found to be clustered in a small area of the chromatic color space. Skin colors of different people are very close, but they differ mainly in intensities [117]. With this finding, we could proceed to develop a skin-color model in the chromatic color space.

A total of 68 skin samples from 68 color images taken from the same number of videos (neutral and painful) were used to determine the color distribution of human skin in chromatic color space and generate the statistical skin-color model. Our samples were taken from persons of different ethnicities: Asian, Caucasian and African and from different ages and genders with varying illumination conditions. As the skin samples were extracted from color images, the skin samples were filtered using a low-pass filter to reduce the effect of noise in the samples. The used low pass filter is as follows:

$$1/9 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Fig. 3.7(a) and 3.7(b) illustrate the training process, in which a skin-color region is selected and its RGB representation is stored. It was verified, using training data, that skin-colors are clustered in color space, as illustrated in Fig. 3.8(a). Although skin colors of different people appear to vary over a wide range, they differ much less in color than in brightness. In other words, skin colors of different people are very close, but they differ mainly in intensities [117]. With this finding, we could proceed to develop a skin-color model in the chromatic color space.

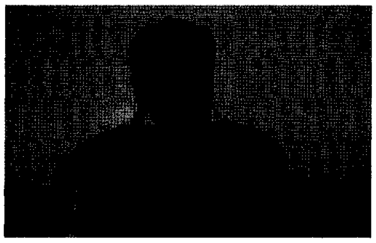


Fig. 3.7(a). Selected skin region in RGB image

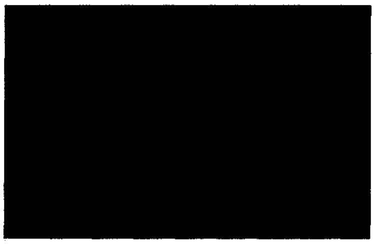


Fig. 3.7(b). Selected skin in Chromatic color

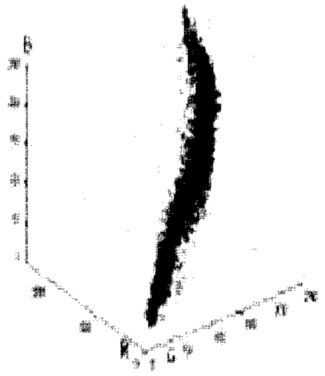


Fig. 3.8(a). Cluster in color space (RGB)

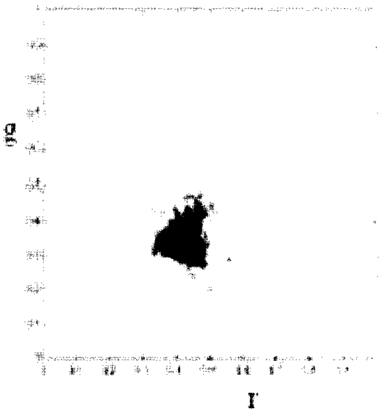


Fig. 3.8(b). Cluster in chromatic space (r,g)

The color histogram revealed that the distributions of skin-color of different people are clustered in the chromatic color space and a skin color distribution can be represented by a Gaussian model N(m, C), where:

Mean, $m = E\{x\}$ [where $x = (r \ \ b)^T$] and

Covariance, $\sum = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix}$

The Gaussian model which was generated from the training data is illustrated in fig 3.9.

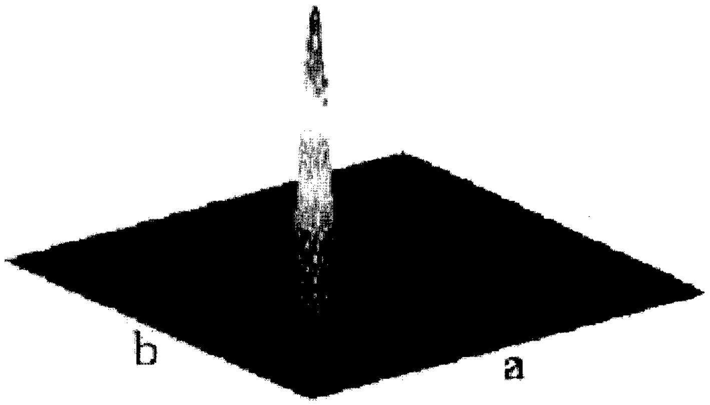


Fig. 3.9. Gaussian model

With this Gaussian fitted skin color model, we can now obtain the likelihood of skin for any pixel of an image. If a pixel, having been transformed from RGB color space to chromatic color space, has a chromatic pair value of (r, b), the likelihood of skin for this pixel can then be computed as follows:

Likelihood = $P(r,b) = \exp[-0.5(x-m)^T C^{-1}(x-m)]$, [where, $x = (r,b)^T$]

Hence, this skin color model can transform a color image into a gray scale image such that the gray value at each pixel shows the likelihood of the pixel belonging to the skin. With appropriate thresholding, the gray scale images can then be further transformed to binary images showing skin regions and non-skin regions.

3.3.1.2 Skin Region Segmentation:

Our main goal in this segmentation process was to remove the background of the image from skin regions using the previously discussed skin color model. At first, the input image is converted to chromatic color space. Then using the Gaussian model, a grayscale image of skin likelihood pixels is constructed.

Each skin pixel has a set of constant values for each r, g and b component. Also every pixel in the normalized image has three values and they are normalized-red, normalized-green and normalized-blue. The segmentation process extracts these normalized components and constructs two images (fig. 3.11(c) and fig. 3.11(d)). Each of these images is converted into black and white image by applying different threshold for normalized input image (fig. 3.11(e) and fig. 3.11(f)) such that r = 0.41-0.50 and g = 0.21- 0.30. Finally, we perform an 'AND' operation between these two black and white images where white pixels are skin and blacks are non skin pixel. In this approach, due to noise and distortion in input image, color information of some skin pixels acts like non skin region and generate non contiguous skin color region. To solve this problem, first a morphological closing operator is used to obtain skin-color blobs (fig. 3.11(g)). A median filter was also used to eliminate spurious pixels (fig. 3.11(h)). Boundaries of skin-color regions are determined using a region growing algorithm in the binary image. Regions with size less than 1% of image size are eliminated [116]. At the end of the segmentation process, black and white skin regions of images are multiplied by the original RGB image and we then get the skin region (fig. 3.11(i)) of the face. Fig. 3.10 illustrates a simple block diagram for the segmentation process and fig. 3.11 shows an example of segmentation and face location detection process performed on a painful image.

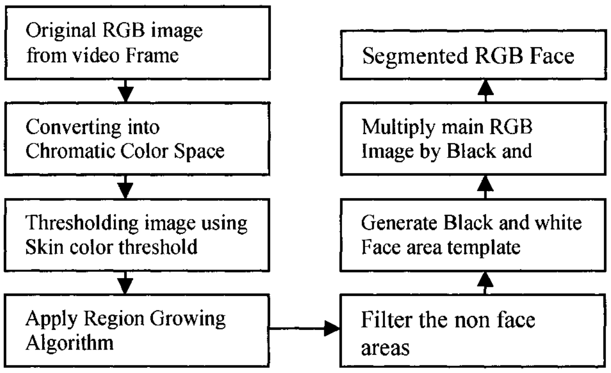

Original RGB image from video Frame
Converting into Chromatic Color Space
Thresholding image using Skin color threshold
Apply Region Growing Algorithm
Filter the non face areas
Generate Black and white Face area template
Multiply main RGB Image by Black and
Segmented RGB Face

Fig. 3.10. Block diagram for face segmentation

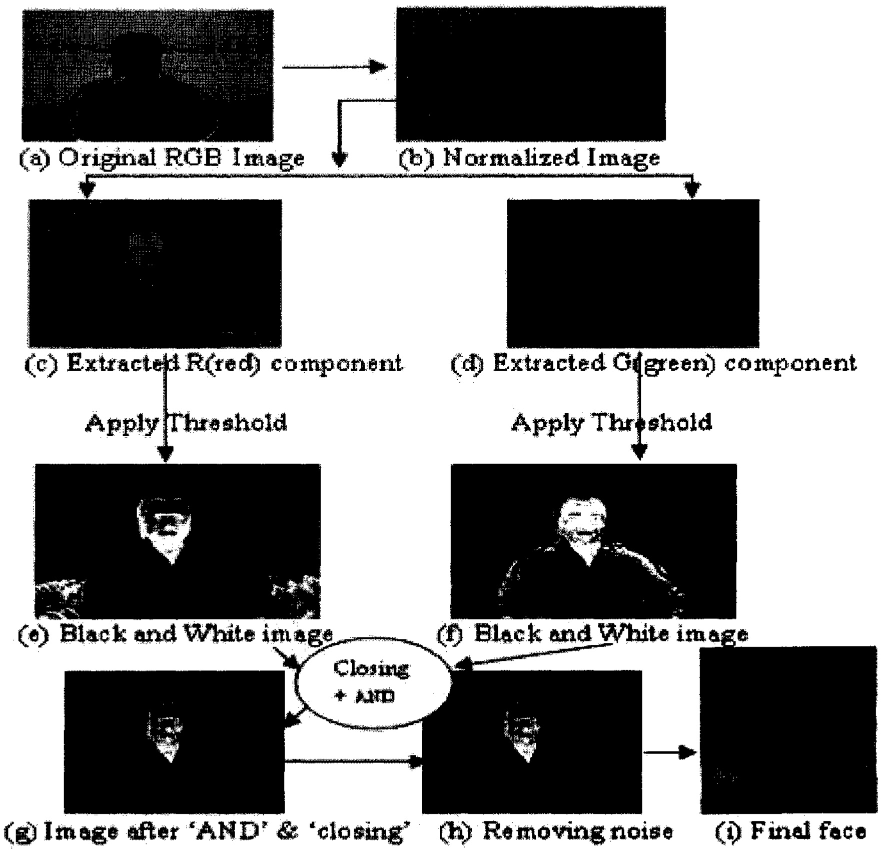

(a) Original RGB Image
(b) Normalized Image
(c) Extracted R(red) component
(d) Extracted G(green) component
Apply Threshold
Apply Threshold
(e) Black and White image
(f) Black and White image
Closing + AND
(g) Image after 'AND' & 'closing'
(h) Removing noise
(i) Final face

Fig. 3.11. Segmentation and approximate face location process

3.3.1.3 Face Detection:

To reduce some search space for eye template matching, bounding rectangles of all connected areas from the black-white template are taken into consideration and the center of the face areas is calculated. This is the mass point of the template area. Now the calculation of the height and width of the bounding rectangle can be performed. If the height-width proportions satisfy for a face-like shape, we keep those areas, otherwise we remove them. Thus the template with the approximate face area is multiplied by the original image and we get the face. Then to consider only the meaningful portions of the face we use a mask image. A bitwise 'AND' operation is used to apply the mask image with the original face image. Features in the image which coincide with the white areas on the mask image will be displayed.

The original video frame, the obtained gray level image, the mask image and the resultant image are shown in fig. 3.12.

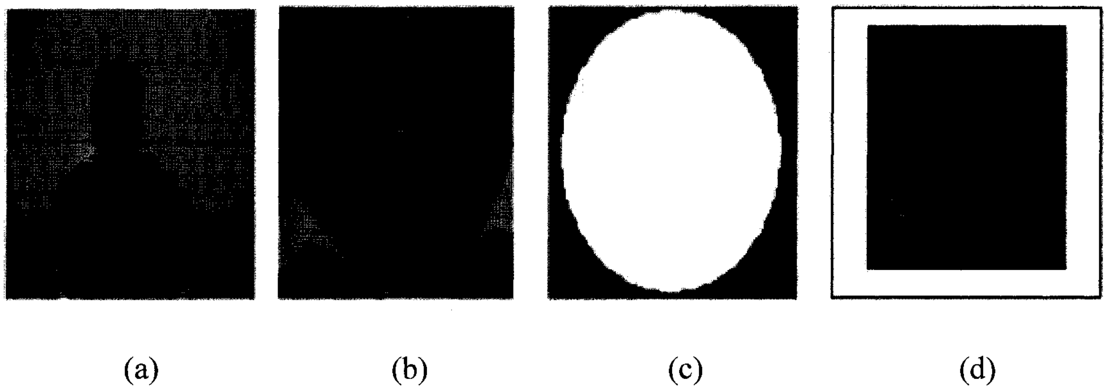


Fig. 3.12. (a) Original video frame (b) gray level image (c) mask image and (d) resultant image

3.4 Conclusion

Face detection is the first step of an image-based pain recognition system. In this chapter, we have given an introduction to face detection techniques and have discussed our face detection approach. In the next chapter, we will discuss the pain recognition algorithms which will use the output of the face detection algorithm.

Chapter 4

Machine Learning Approaches for Pain Recognition

4.1 Introduction

The face plays a crucial role in interpersonal communication. Seeing a face, we can recognize a person's identity, gender, age, expression, etc. This information is irreplaceable for the normal conduct of human communications. If machines could recognize such information from a human face, humans and machines might thereby communicate more smoothly, robustly, and harmoniously. In the previous chapters, we have looked at the various face detection and facial expression techniques. Also we have distinguished between feature-based and image-based expression recognition techniques. In our work, we will use image –based pain recognition method.

Analysis of the face is the main task in image-based pain recognition method because, during pain, distinct changes occur in the face region. Like humans, who recognize pain of a person by seeing the face of that person, machines can also detect pain (and all other expressions) of a person by the analysis of one's facial image.

In our research, we have developed two image-based pain recognition systems using two machine learning techniques for training and recognizing pain from the input videos. The approaches are – the Eigenimage method and the multilayer neural network method.

The rest of the chapter will describe these two approaches. Section 4.2 will describe the eigenimage-based pain recognition system while section 4.3 will describe the multilayer neural network –based pain recognition system.

4.2 Eigenimage-based Pain Recognition

The Eigenimage approach is a principal component analysis method, in which a small set of characteristic pictures is used to describe the variation between the images. In this method, the goal is to find out the eigenvectors (eigenimages) of the covariance matrix of the distribution, spanned by a training set of images. Every image is represented by a linear combination of these eigenvectors. Evaluations of these eigenvectors are quite difficult for typical image sizes but, an approximation can be made. Recognition is performed by first projecting a new image into the subspace spanned by the eigenimages and then classifying the image by comparing its position in the image with the positions of the known individuals. The general block diagram for eigenimage-based pain recognition is given in figure 4.1.

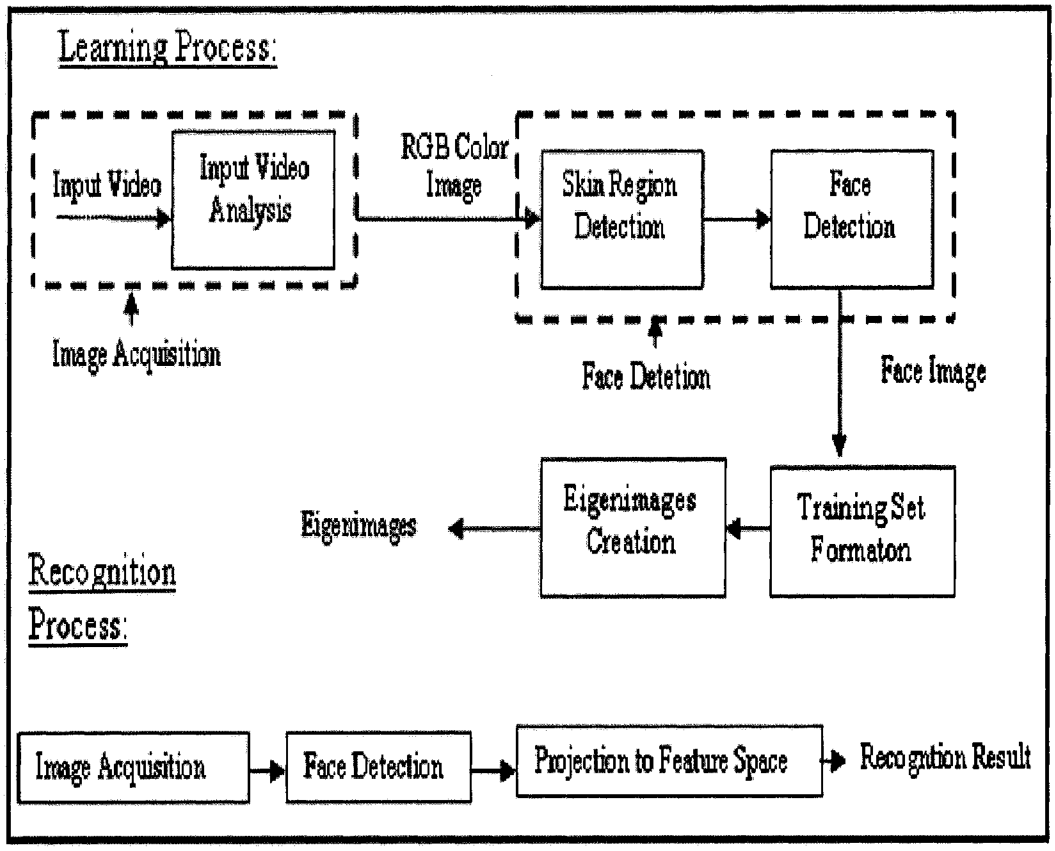


Fig. 4.1. Block diagram of eigenimage-based pain recognition system

We have used the whole faces, eye regions and lip regions as our image set. So three set of eigenimages are produced in our work. They are eigenface, Eigeneye and Eigenlip. The reason for choosing eye and lip region is because these regions are mostly affected by pain or we can say that most changes occur in the eye and lip regions of the face during pain.

We have used a total of 68 videos for our pain recognition system. Half of these videos are of neutral mood and half are of painful mood. Figure 4.2 shows the sample training images for producing eigenfaces.

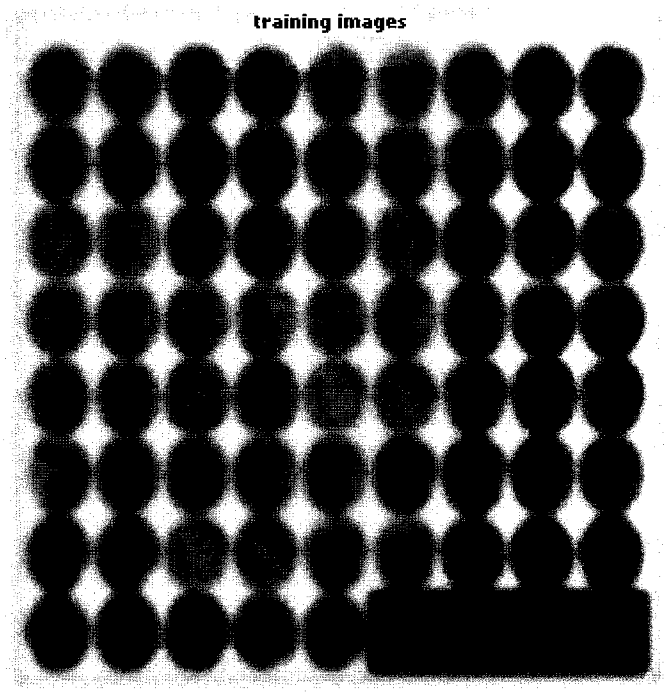


Fig. 4.2. Training images for Eigenfaces

4.2.1 Calculating Eigenfaces

Let a face image I (x, y) be a two-dimensional N X N array of (8-bit) intensity values. Such an image may also be considered as a vector of dimension N^2, so that a typical image of size N x N becomes a vector of dimension N^2 or, equivalently, a point in N^2-dimensional space. Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the PCA method is to find the vectors which best account for the distribution of face images within the entire image space. These vectors define the subspace of face images called "face space". Each vector is of length N^2, describes an N X N image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, they are referred to as Eigenfaces [94].

Steps for Eigenfaces calculation:

1. The first step is to obtain a set S with M face images. Each image is transformed into a vector of size N and placed into the set.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \cdots\cdots\cdots, \Gamma_M\}$$

2. The second step is to obtain the mean image Ψ.

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

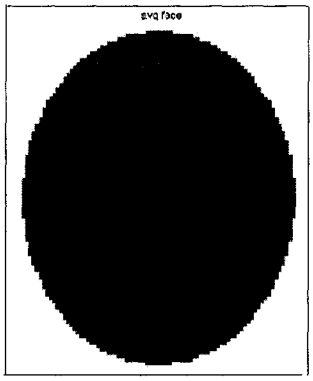


Fig. 4.3. Average Image

3. Then we find the difference Φ between the input image and the mean image

$$\Phi_i = \Gamma_i - \Psi$$

4. Next we seek a set of M orthonormal vectors, u_n, which best describes the distribution of the data. The k_{th} vector, u_k, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}\left(u_k^T \Phi_n\right)^2$$

5. λ_k is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & \text{If } l=k \\ 0 & \text{Otherwise} \end{cases}$$

where u_k and λ_k are the eigenvectors and Eigenvalues of the covariance matrix C

6. The covariance matrix C has been obtained in the following manner

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \text{ [where, } A=\{\Phi_1,\Phi_2,\cdots\cdots,\Phi_n\} \text{]}$$

7. To find eigenvectors from the covariance matrix is a huge computational task. Since M is far less than N^2 by N^2, we can construct the M by M matrix,

$$L = A^T A, \quad [\text{ where } L_{mn} = \Phi_m^2 \Phi_n]$$

8. We find the M Eigenvectors, v_l of L.

9. These vectors (v_l) determine linear combinations of the M training set face images to form the eigenfaces u_l (figure 4.4)

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k , \quad l = 1,2,3, \ldots\ldots\ldots, M$$

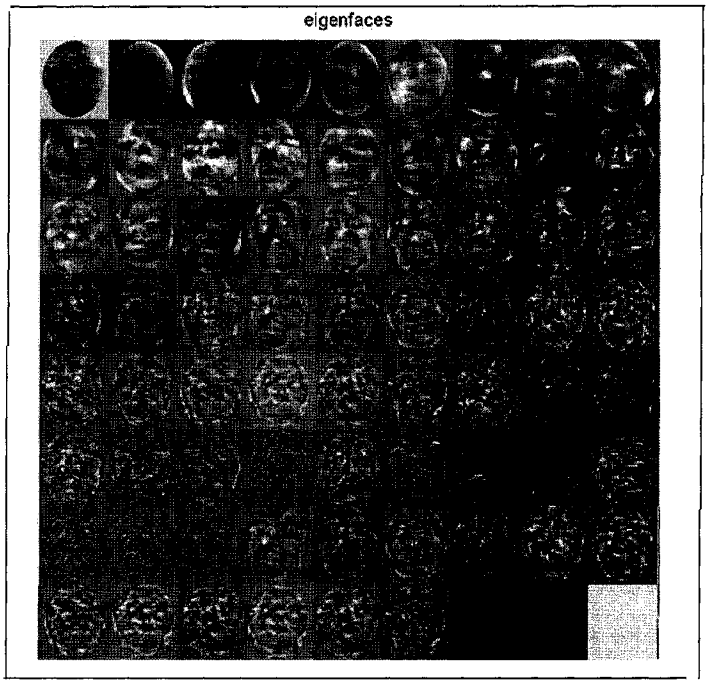


Fig. 4.4. Eigenfaces for recognition

After computing the Eigenvectors and Eigenvalues on the covariance matrix of the training images, these M eigenvectors are sorted in order of descending Eigenvalues and chosen to represent Eigenspace.

Finally, we project each of the original images into Eigenspace. This gives a vector of weights representing the contribution of each Eigenface to the reconstruction of the given image.

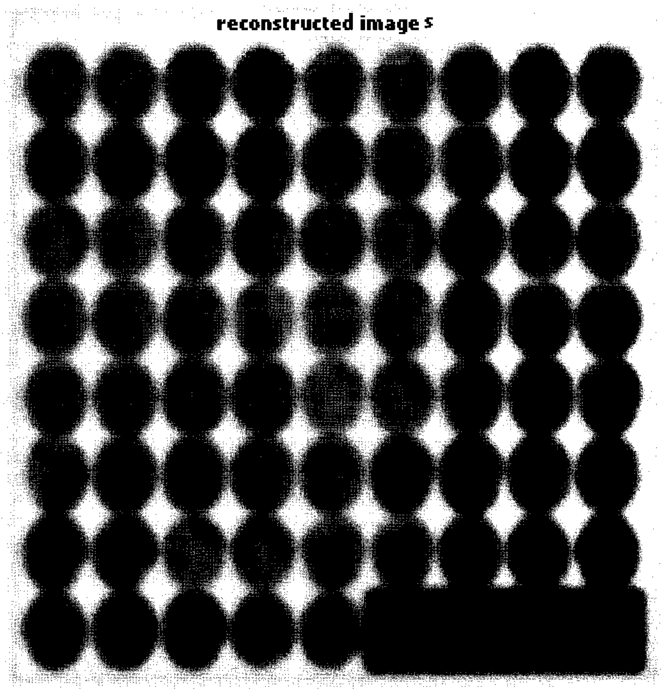


Fig. 4.5. Reconstructed training images for eigenfaces

4.2.2 Recognition using Eigenfaces

Once eigenspace has been defined, we can project any image into eigenspace by a simple matrix multiplication:

$$w_k = \mathrm{u}_{\mathrm{k}}^{\mathrm{T}}(\Gamma - \Psi) \qquad \Omega^{\mathrm{T}} = \left[\omega_1, \omega_2, \cdots\cdots\cdots, \omega_{\mathrm{M}}\right]$$

where, u_k is the kth eigenvector and ω_k is the kth weight in the vector Ω^T $=[\omega_1,\omega_2,\omega_3,\ldots\omega_M]$. The M weights represent the contribution of each respective Eigenfaces. The vector Ω, is taken as the 'image-key' for an image projected into Eigenspace. We compare any two 'image-keys' by a simple Euclidean distance measure,

$$\frac{\| \Phi - \Phi_f \|}{\| \Phi_f \|} \leq \varepsilon_k .$$

An acceptance (the two images match) or a rejection (the two images do not match) is determined by applying a threshold. Any comparison producing a distance below the threshold is a match.

The steps for recognition process are as follows:

1. When an unknown face is found, project it into eigenspace.
2. Measure the Euclidean distance between the unknown face's position in eigenspace and all the know faces' positions in eigenspace.
3. Select the face closest in eigenspace to the unknown face as the match.

4.2.3 Rebuilding an Image using Eigenfaces

A face image can be approximately reconstructed (rebuilt) by using its feature vector and the eigenfaces as

$$\Gamma' = \Psi + \Phi_f$$

where

$$\Phi_f = \sum_{i=1}^{M'} w_i u_i \quad \text{is the projected image.}$$

We see that the face image under consideration is rebuilt just by adding each eigenface with a contribution of w_i to the average of the training set images. The degree of the fit or the "rebuild error ratio" can be expressed by means of the Euclidean distance between the original and the reconstructed face image as

$$\text{Rebuild Error Ratio, RER} = \frac{\| \Gamma' - \Gamma \|}{\| \Gamma \|}$$

It has been observed that, the rebuild error ratio increases as the training set members differ heavily from each other. This is due to the addition of the average face image. When the members differ from each other, especially in image background, the average face image becomes messier and this increases the rebuilding error ratio. There are four possibilities for an input image and its pattern vector:

1. Near a face space and near a face class,
2. Near a face space but not near a known face class,
3. Distant from a face space and near a face class,
4. Distant from a face space and not near a known face class.

In the first case, an individual is recognized and identified. In the second case, an unknown individual is presented. The last two cases indicate that the image is not a face image. Case three typically shows up as a false classification. It is possible to avoid this false classification in this system as

$$\frac{\| \Phi - \Phi_f \|}{\| \Phi_f \|} \leq \theta_k$$

where θ_k is a user defined threshold for the faceness of the input face images belonging to kth face class.

If the image is found to be an unknown face, we could decide whether or not we want to add the image to our training set for future recognitions. Figure 4.6 and 4.7 depict the images before and after recognition respectively.

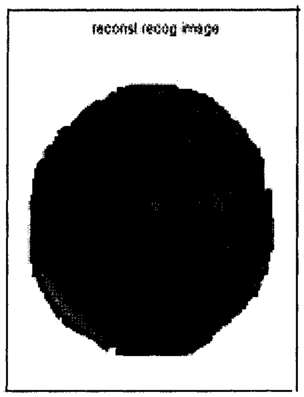


Fig 4.6. Image to recognize.

The image after the reconstruction process is

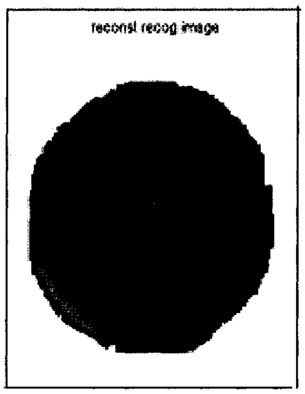
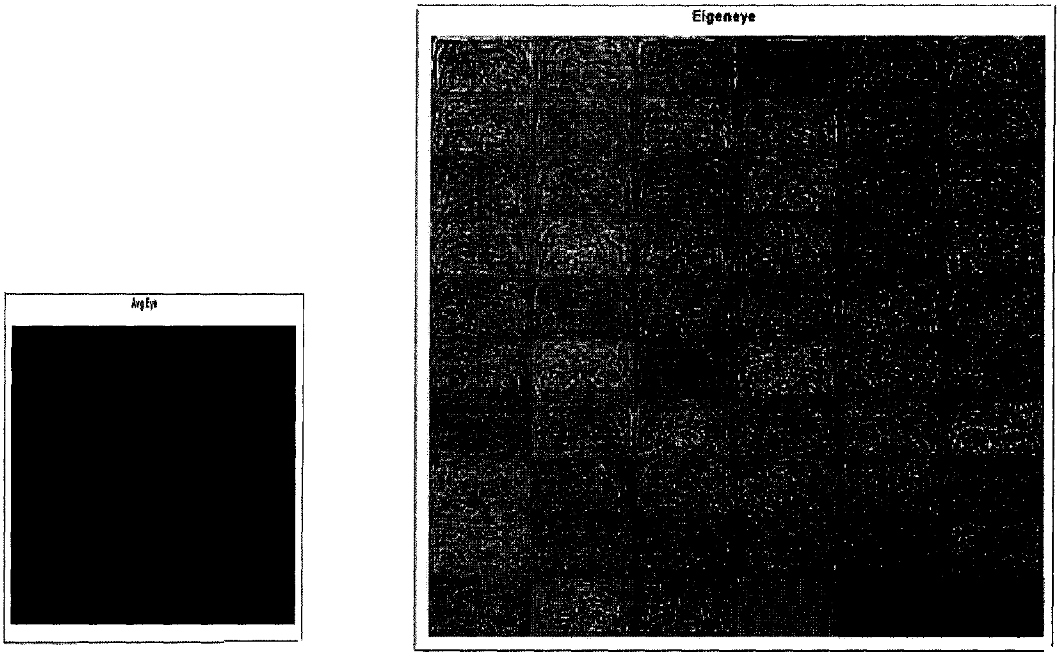


Fig 4.7. Image after the reconstruction process.

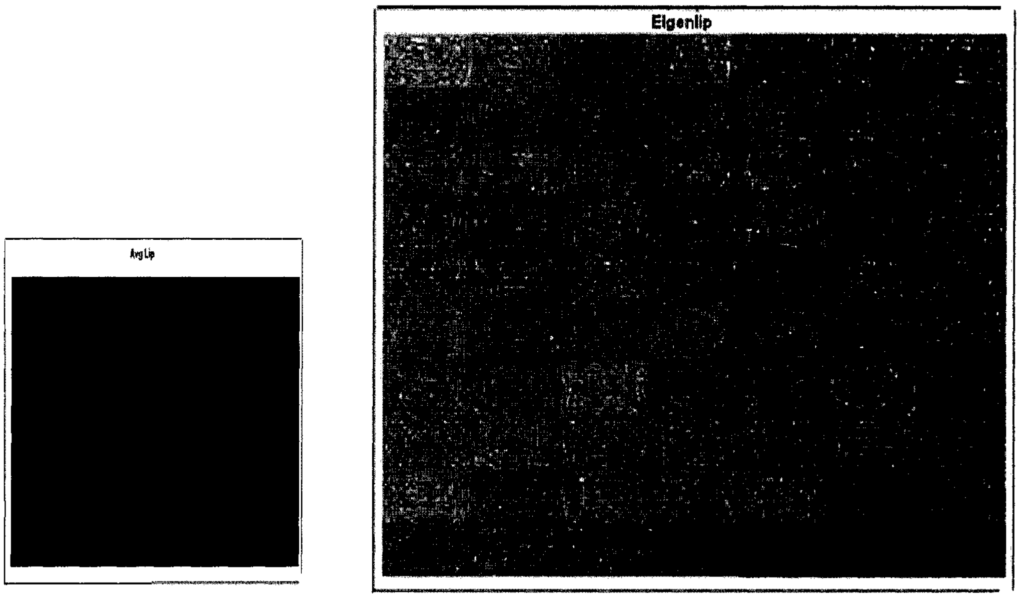
4.2.4 Eigeneye and Eigenlip Methods:

The method for eigeneyes and eigenlips are similar to the eigenface method except that in these methods, instead of using the whole face, only the segmented eye or lip portions of the face images are used. The average eye and lip image and eigeneyes and eigenlips are shown in figure 4.8 and figure 4.9.



(a) (b)

Fig. 4.8. (a) Average eye (b) Eigeneyes



(a) (b)

Fig. 4.9. (a) Average lip (b) Eigenlips

We have discussed an eigenimage –based pain recognition system in the previous section. First we have described the eigenface technique and then eigeneye and eigenlip methods were also illustrated. In the next section, we will discuss the multilayer neural network-based pain recognition method.

4.3 Multilayer Neural Network-based Pain Recognition

Neural network is another machine learning technique that we have used in our pain recognition system. The general block diagram of neural network-based pain recognition is shown in figure 4.10.

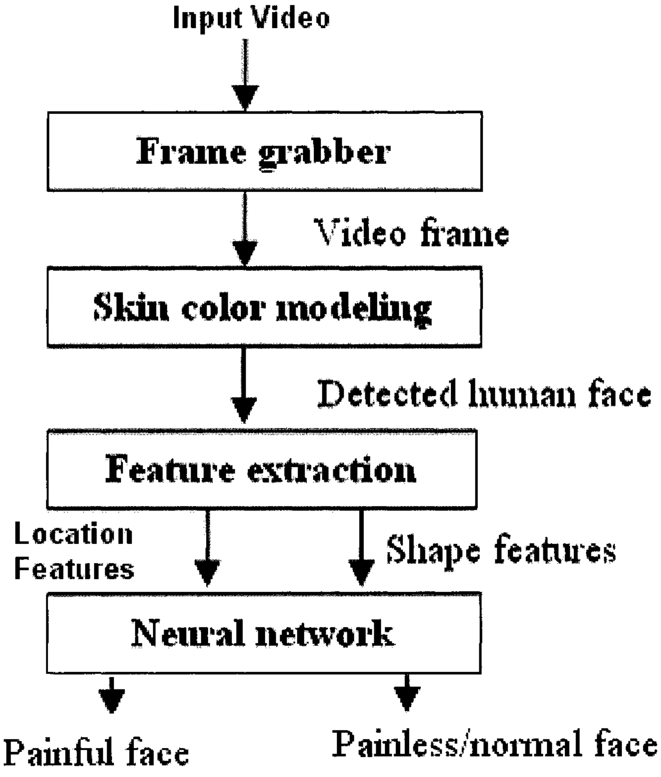


Fig. 4.10. Block diagram of neural network-based pain recognition system

The following sections will give an overview and will illustrate our implementation. Section 4.3.1 will give us some neural network basics, section 4.3.2 will illustrate the feature extraction process and section 4.3.3 will describe the learning and recognition process.

4.3.1 Neural Network Basics:

Neural networks can be used in many ways for any pattern recognition problem. In our implementation, we have used a multilayer error back propagation algorithm.

4.3.1.1 Artificial Neural Networks

An artificial neural network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANNs as well.

With their remarkable ability to derive meaning from complicated data, ANNs can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real-time operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault tolerance via redundant information coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Therefore, the important components of neural networks are “unit” and “connection.” Neural networks can be categorized in two ways: how the units are connected and the type of information processing in the unit. From the point of how the units are connected, the neural networks are categorized as layered network and mutually connected network

A layered network is a network, which has a layered structure of units with layers ordered from the input layer to the output layer. A unit in a layer is only connected to the units in the next higher layer. Radial Basis Function (RBF) networks are layered networks with three layers. Multilayer perceptrons in general fall in this category.

A mutually connected network is a network, which allows connections between any two units for both directions. Hopfield networks and Boltzmann machines explained in the following use networks of this type.

From the point of view of the information processing in the unit, neural networks are categorized as follows:

- Hopfield networks
- Boltzmann machine
- Backpropagation

A Hopfield network [118] uses a mutually connected network with symmetrical weights. Hopfield networks are used for associative memories and solving optimization problems.

A "Boltzmann machine" [119] is essentially a stochastic version of the Hopfield network. A Boltzmann machine can also learn the weights of states of the environment and can simulate the environment later.

Backpropagation is a learning algorithm (procedure) proposed for multilayer networks. We will explain this algorithm in section 4.3.4.

In the rest of this chapter, we use the word "neural network" to mean a multilayer perceptron without any confusion since we do not handle the other types of the networks hereafter.

4.3.1.2 Structure of Multilayer Perceptrons

A multilayer perceptron is a network of units, where a unit receives outputs of other units or the input from the environment as its inputs, and where a unit outputs to other units or to the environment.

A unit has more than or equal to one input and a single output. The weighted sum of the inputs is combined with a bias and then it is operated on by a sigmoid function to produce the output. Let n be the number of inputs to the unit, o be the output, x_i's be the inputs, w_i's be the weights, b be the bias, and σ be the sigmoid function. The output of the unit is given as follows.

(i) $$O = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

Here n is a positive integer, o, x_i's, w_i's, and b are real numbers, and the sigmoid function, σ is a one-dimensional non-linear monotonic differentiable function. Even though any function can serve as a sigmoid function for a unit as long as it is one-dimensional, non-linear, monotonic, and differentiable, the following function, given by equation (i), is used in our implementation of neural networks and throughout the experiments described in this thesis and a graph of that function is given by figure 4.11.

(ii) $$\sigma(x) = \frac{1}{1+e^{-x}}$$

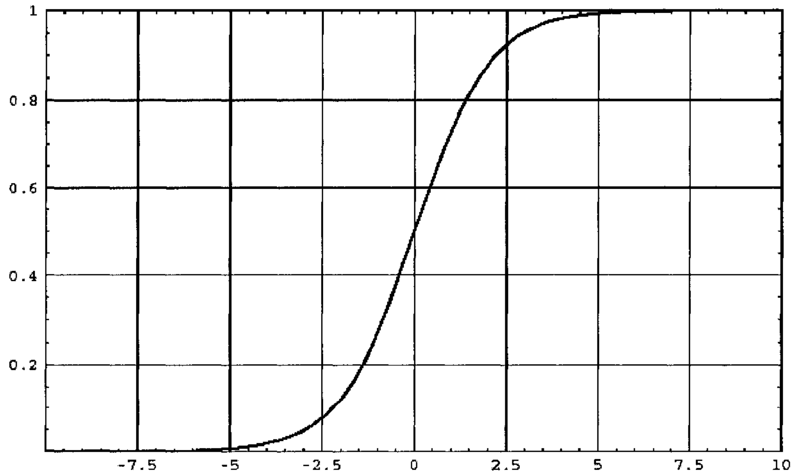


Fig. 4.11. Sigmoid function

A unit, which receives the input from the environment, is called an "input unit." A unit, which outputs to the environment, is called an "output unit." A multilayer perceptron must have at least one input unit and one output unit. Here the "environment" means the outside of the network. We decided to allow any connection in the network. The only restriction is that the network is connected to the environment (i.e. outside) in both ways. We do not assume any layer structure for a multilayer perceptron. Units in the network can have different sigmoid functions.

This kind of the networks is actually a layered network. The input layer is the layer, which consists entirely of the input units. The output layer is the layer, which consists entirely of the output units. Hidden layers are the layers which consists entirely of the units without connections to the environment. A hidden layer is sometimes called a middle layer.

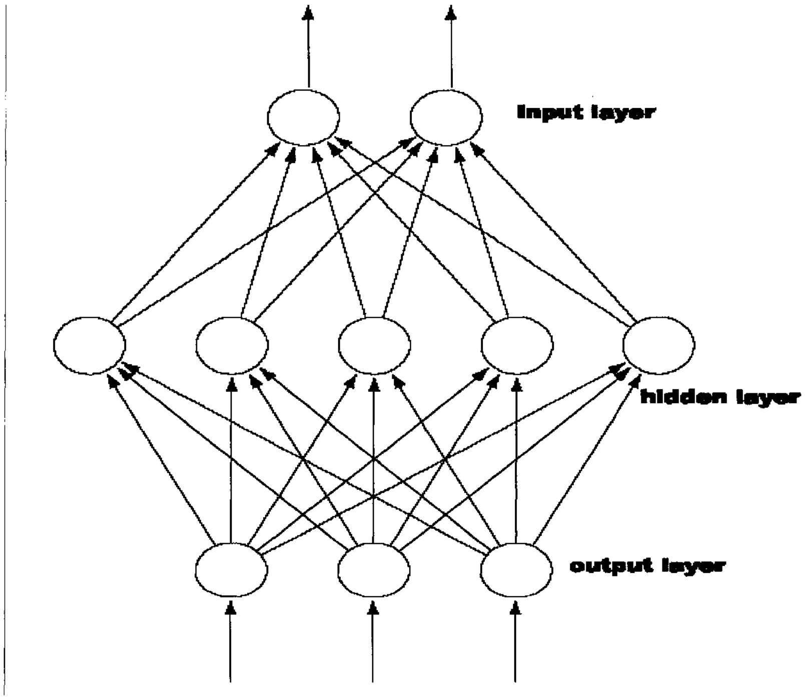


Fig. 4.12. Structure of a Multilayer Perceptron

Multilayer perceptrons will be called feed-forward ANNs when they allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

4.3.1.3 Back Propagation for Multilayer Perceptron

The backpropagation technique consists of the backpropagation of the errors by the environment through the network from the output units to the input units, and weight and bias updates. The purpose of backpropagation is to adjust the internal state (weights and biases) of the multilayer perceptron so that the multilayer perceptron produces the desired output for the specified input.

In order to realize this, the following error function, which is sometimes called "energy function", *E* is defined for desired input and output pairs *{(Ip,Op)$_p$}*, where n is the function defined by the multilayer perceptron.

(iii)

$$E = \frac{1}{2}\sum_{P}(O_P - n(I_P))^2$$

For each pair, the multilayer perceptron is made to propagate the input *Ip* forward, then the squared distance between the output of the network *n(Ip)* and the desired output is calculated. The squared distances are summed for all the pairs and divided by 2 to produce the error function. If the error function is 0, it means that the multilayer network produces exactly the desired output for each input.

The backpropagation algorithm is essentially a gradient descent procedure with respect to this error function. The weights (and the biases) are therefore updated as follows, where α is some positive constant, called the "learning constant."

$$(iv) \qquad \Delta\omega = -\alpha\frac{\delta E}{\delta\omega}$$

Furthermore a momentum term by the past weight update value $\tilde{\Delta} w$ is added as follows to avoid oscillation for a practical purpose of making rapid learning possible [120],

$$(\mathrm{v}) \qquad \Delta\omega = -\alpha\frac{\delta E}{\delta\omega} + \beta\tilde{\Delta}\omega$$

where β is a positive constant less than 1.

The backpropagation algorithm is used for calculation of δ, and the value is assigned to each unit. In backpropagation, this δ is propagated backward from the unit to those units, which output to that unit. Actually backpropagation is an embodiment of repeated applications of the chain rule for partial derivatives.

This algorithm is completely localized to each unit. A weight update can be calculated from δ and the output of the unit involved. This is the reason why backpropagation is applicable not only to single-layer, but also to multilayer perceptrons and such networks even without layer structures.

4.3.2 Feature Extraction

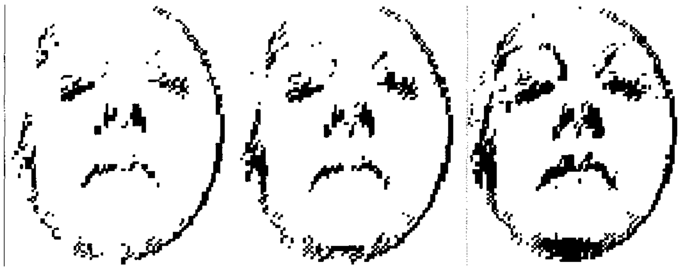
After detecting the human face from video frames, we need to detect reliable facial features. We observe that most facial feature changes that are caused by pain are in the areas of eyes, brows and mouth. In this thesis, two types of facial features in these areas are extracted: location features and shape features. The idea for extracting features presented here is similar to that taken by Yang et al. [121] and Ying-li Tian et al. [122]. It is an attempt to make the feature extraction robust to the available video sequences for this field of research.

4.3.2.1 Location Features Extraction

In this system, six location features are extracted for pain recognition. They are two eye centers, two eyebrow inner endpoints and two corners of the mouths.

4.3.2.1.1 Eye Corners and Eyebrow Inner Endpoints

To find the eye centers and eyebrow inner endpoints inside the detected frontal or near frontal face, we have developed an algorithm that searches for two pairs of dark regions which correspond to the eyes and the brows by using certain geometric constraints such as position inside the face, size and symmetry to the facial symmetry axis. Similar to reference [121], the algorithm employs an iterative thresholding method to find these dark regions. Figure 4.13 shows the iterative thresholding method to find eyes and brows. Generally, after four iterations, all the eyes and brows are found. If satisfactory results are not found after 15 iterations, we think the eyes or the brows are occluded. Unlike the work of Yang *et al.* to find one pair of dark regions for the eyes only, we find two pairs of parallel dark regions for both the eyes and eyebrows. By doing this, not only more features are obtained, but also the accuracy of the extracted features is improved. Figure 4.13 illustrates this process. Figure 4.13(a), 4.13(b) and 4.13(c) are the binary images after applying threshold value 45, 55 and 65 respectively to the original image. As shown in figure 4.13(a), the right brow and the left eye is wrongly extracted as the two eyes in Yang's approach. Figure 4.13(c) shows the correct positions are extracted for all the eyes and eyebrows in our method. Then the eye centers and eyebrow inner endpoints can be easily determined.



(a) (b) (c)

Fig. 4.13. Iterative thresholding of the face to find eyes and brows.

4.3.2.1.2 Mouth Corners

After finding the positions of the eyes, the location of the mouth is first predicted. Then the vertical position of the line between the lips is found using an integral projection of the mouth region proposed by Yang *et al.* [121]. Finally, the horizontal borders of the line between the lips are found using an integral projection over an edge-image of the mouth. After Yang *et al.* , the following steps are use to track the corners of the mouth:

1) Finding two points on the line between the lips near the previous positions of the corners in the image

2) Searching along the darkest path to the left and right, until the corners are found.

Finding the points on the line between the lips can be done by searching for the darkest pixels in search windows near the previous mouth corner positions. Because there is a strong change from dark to bright at the location of the corners, the corners can be found by looking for the maximum contrast along the search path [122].

4.3.2.2 Location Feature Representation

After extracting the location features, all faces are normalized to 90 x 90 pixels. We transform the extracted features into a set of parameters. We represent the face location features by 5 parameters, which are shown in figure 4.14. These parameters are the distances between the eye-line and the corners of the mouth, the distances between the eye-line and the inner eyebrows, and the width of the mouth (the distance between two corners of the mouth).

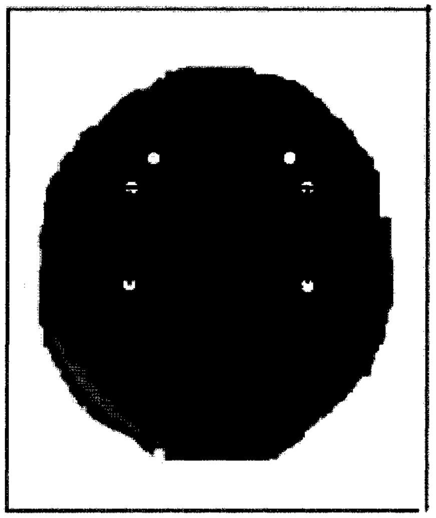


Fig. 4.14. Face location feature representation for expression recognition

In figure 4.12, L1 and L2 are the distances between the eye-line and the inner eyebrows, L3 and L4 are the distances between the eye-line and the corners of the mouth and L5 is the width of the mouth (the distance between two corners of the mouth).

4.3.2.3 Shape Feature Extraction

In order to extract the mouth shape features, first an edge detector is applied to the normalized face to get an edge map. Here the edge map is divided into 2 x 2 zones as shown in figure 4.15.

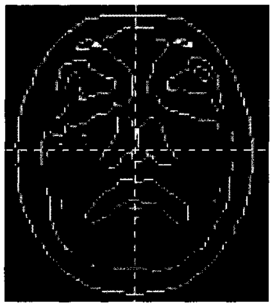
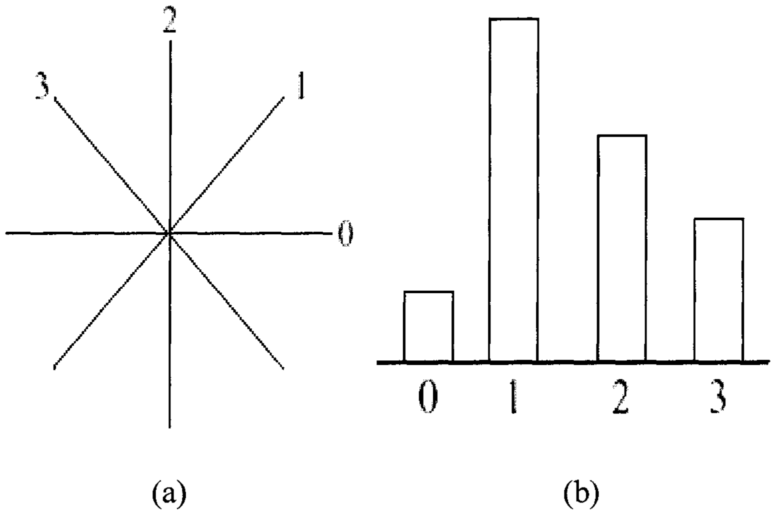


Fig. 4.15. Zones of the edge map of a sample normalized face

The eyes and mouth shape features are computed from zonal shape histograms of the edges in the mouth and eyes region. To place the 2 x 2 zones onto the face image, the upper two zones are placed at the locations of the eyes and the lower two portions are placed at the location of mouth. The coarsely quantized edge directions are represented as local shape features and more global shape features are presented as histograms of local shape (edge directions) along the shape contour. The edge directions are quantized into 4 angular segments (figure 4.16(a)). Representing the whole mouth as one histogram does not capture the local shape properties that are needed to distinguish pain expressions. Therefore we use the zones to compute four histograms of the edge directions. Hence, the eyes and mouth is represented as a feature vector of 16 components (4 histograms of 4 components). An example of the histogram of edge directions corresponding to the lower right zone is shown in figure 4.16(b).



(a) (b)

Fig. 4.16. (a) Four quantization levels and (b) Histogram corresponding to the middle zone of the mouth.

4.3.3 Pain Recognition using Neural Network

The proposed system is applied on a wide variety of painful and neutral video sequences collected from a variety of peoples (students, faculties, officers etc.) of the University of Rajshahi, Bangladesh. The videos were taken individually in different lightening conditions and different backgrounds. It is found that the system successfully detects skin region of the images collected from video analysis. However, it is important to note that not all detected regions contain faces. Some correspond to parts of human body, while other corresponds to objects with colors similar to those of skin. We implemented the entire algorithm in MATLAB 7.0 on a PENTIUM-IV windows XP workstation.

We have used a neural network-based recognizer having the structure shown in figure 4.17. The standard back-propagation in the form of a layered neural network with varying number of hidden layers was used to recognize facial expressions. The inputs to the network were the 5 location features (figure 4.14) and the 16 zone components of shape features of the eyes and mouth regions (figure 4.16). Hence, a total of 21 features were used to represent the amount of pain in a face image. The outputs were a set of two values – painful face or painless face. We tested various numbers of hidden units and found that 10 hidden units gave the best performance.

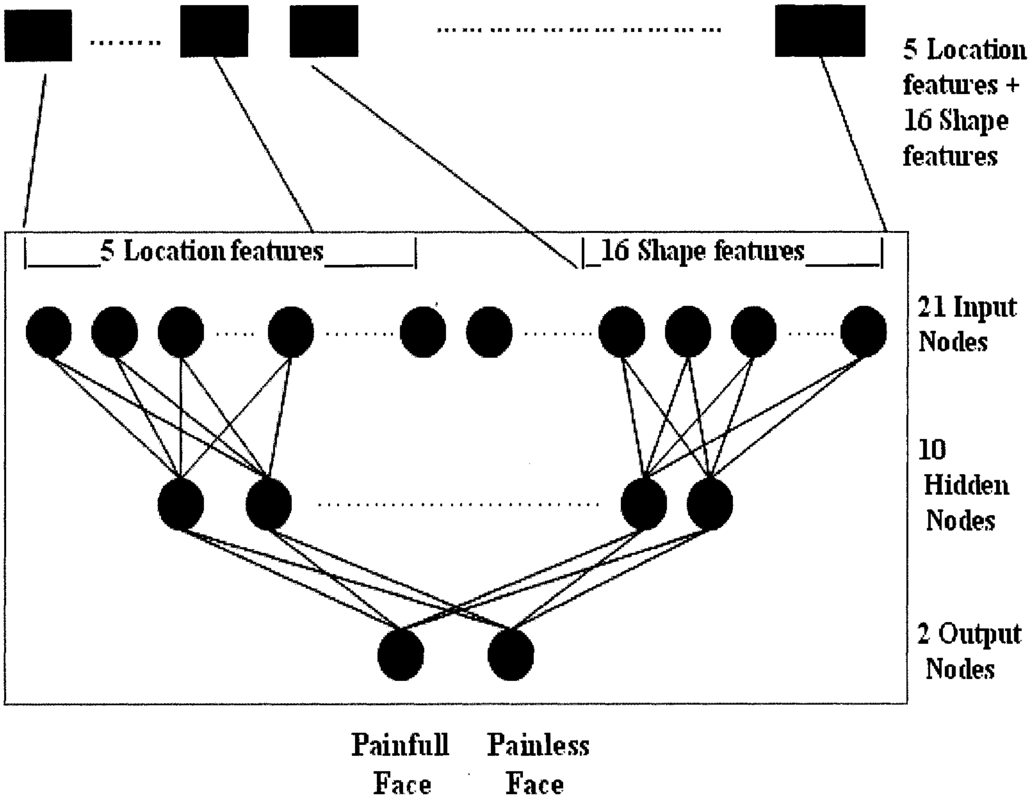


Fig. 4.17. Neural network-based pain recognizer

4.4 Conclusion

In this chapter we have discussed the two machine learning approaches for pain recognition. After some introduction in section 4.1, an eigenimage-based pain recognition system is described in section 4.2. In section 4.3, multilayer neural network-based pain recognition system is illustrated.

In the next chapter, we will report on the results obtained using eigenimage-based and neural network-based pain recognition approaches and then will compare the results.

Chapter 5

Simulations and Results

1.1 Introduction

Two image-based pain recognition systems are developed in this work. One of the systems uses eigenimage method, while the other uses multilayer neural network for learning and recognition. Also for face detection from the available frame, skin color modeling technique is used. We have described the face detection procedure in chapter 3 and in chapter 4, we have described the learning and recognition procedures by eigenimage and neural network methods. The results of these experiments will be presented and explained in this chapter.

In section 5.2, some details of the obtained dataset will be presented. The origin of the dataset, the number of videos, the nature of the contents and the technical details of those videos and the image extraction procedure will also be discussed.

In section 5.3 and 5.4, the results of the eigenimage-based method and neural network- based method will be discussed respectively. Three eigenimage-based methods will be presented in section 5.3. They are eigenface, eigeneye and eigenlip. The individual and combined results of these three methods will be presented. In section 5.4, the results with the multilayer neural network-based classifier will be discussed. The variations of the results with different numbers of hidden layers and different numbers of neurons of hidden layers will be compared.

Section 5.5 will show the comparison between these two machine learning methods for pain recognition. Two types of comparison results will be shown – speed comparison and accuracy comparison.

Finally section 5.6 will summarize the results of the research.

5.2 Image Acquisition

We have used a video database of persons with painful and neutral mood. This database was collected from the Computer Science & Engineering department from University of Rajshahi, Bangladesh. In this database we have 68 video files of 34 persons. These persons are of different colors, ethnicities, ages and genders and all have shoulder pain. Two videos were taken for each person. One was in neutral mood and the other was in painful mood. The pain was generated for raising hand or shaking head etc.. All of these videos were taken intentionally (i.e., the respondents are aware of the event). The resolutions of the videos were 96 x 96. These video files were first read and the numbers of frames of each video were determined. The middle frame of the videos were then stored as an image in the database for further processing. The reason for taking middle frame was that, in almost all the pain videos, the expression for pain begins after some time from the starting of the videos and ends some time before the ending of the videos. So, by taking the middle frame, it was ensured that the expression for pain in a pain video will be captured. The videos were roughly 1 to 1.5 second long.

5.3 Results of Eigenimage Method

The general block diagram of eigenimage-based pain recognition method is depicted in figure 5.1.

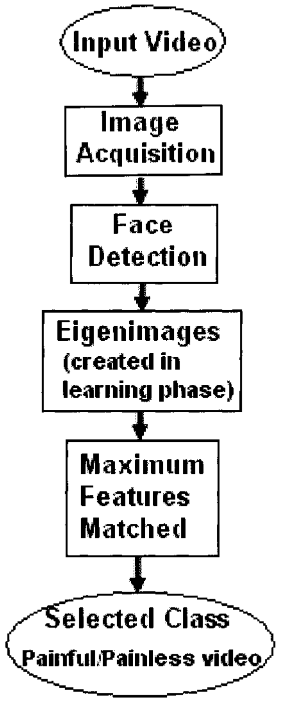

Input Video
Image Acquisition
Face Detection
Eigenimages (created in learning phase)
Maximum Features Matched
Selected Class Painful/Painless video

Fig. 5.1. Block diagram of eigenimage-based pain recognition system

The motivation behind use of the eigenimage is that, previous work ignored the question of which features are important for classification, and which are not. The Eigenimage approach seeks to answer this by using Principal Component Analysis (PCA) of the facial images. This analysis reduces the dimensionality of the training set, leaving only those features that are critical for face recognition.

The system is initialized by first acquiring the training set (ideally a number of examples of each subject with varied lighting and expression). Eigenvectors and

Eigenvalues are computed on the covariance matrix of the training images. The M highest eigenvectors are kept. Finally, the known individuals are projected into the face space, and their weights are stored. This process is repeated as necessary.

The steps for recognition process are as follows:

1. When an unknown image is found, we project it into eigenspace.
2. We first measure the Euclidean distance between the unknown image's position in eigenspace and all the know images' positions in eigenspace.
3. Then we select the face closest in eigenspace to the unknown image as the match.

The system is implemented in Matlab 7.0 on a PENTIUM-IV windows XP workstation. The following table shows the average skin region detection rate, average false skin detection rate and average face detection rate of the detected skin region of the proposed system:

Table 5.1. Average face detection rate

Test no.	Number of test videos	Skin region detection rate	False skin region detection rate	No. of detected Face
1	10	89% ± 1%	14% ± 2%	9 ± 1
2	20	91% ± 2%	12% ± 2%	17 ± 1
3	40	81% ±1%	8% + 3%	35 ± 3
4	55	89% ± 3%	16% ± 3%	52 ± 2
5	68	92% ± 3%	15% ± 1%	63 ± 3

From the above resultant table it is found that the average face detection rate is

90% ± 2% and it does not depend on the number of input videos. From experimental data of Gaussian distribution it is also observed that there is no difference between the chromatic color space for infants and adults.

We have used three eigenimage techniques in our research. They are eigenface, eigeneye and eigenlip. To check the accuracy of our pain recognition system, we have tested each eigenimage method individually and collectively. Table 5.2 shows the accuracy results of different eigenimage methods.

Table 5.2. Comparison of various eigenimage methods for pain recognition

Method(s) used	Average pain recognition rate
Eigenfaces	89% ± 2%
Eigeneyes	82% ± 3%
Eigenlips	84% ± 5%
Eigenfaces & Eigeneyes	89% ± 1%
Eigenfaces & Eigenlips	90% ± 3%
Eigeneyes & Eigenlips	86% ± 4%
Eigenfaces, Eigeneyes & Eigenlips	92% ± 2%

As the table shows, among the eigenface, eigeneye and eigenlip methods, eigenface method gives us the best performance which is 89% ± 2%. We also notice the results obtained by combining two or three eigenimage methods. Combination of eigenface and eigeneye methods gives us 89% ± 1%, combination of eigenface and eigenlip methods gives us 90% ± 3% and combination of eigeneye and eigenlip methods gives us 86% ± 4%. But the best result (92% ± 2%) we obtained is by combining eigenface, eigeneye and eigenlip methods.

The results are also shown by a bar diagram in figure 5.2.

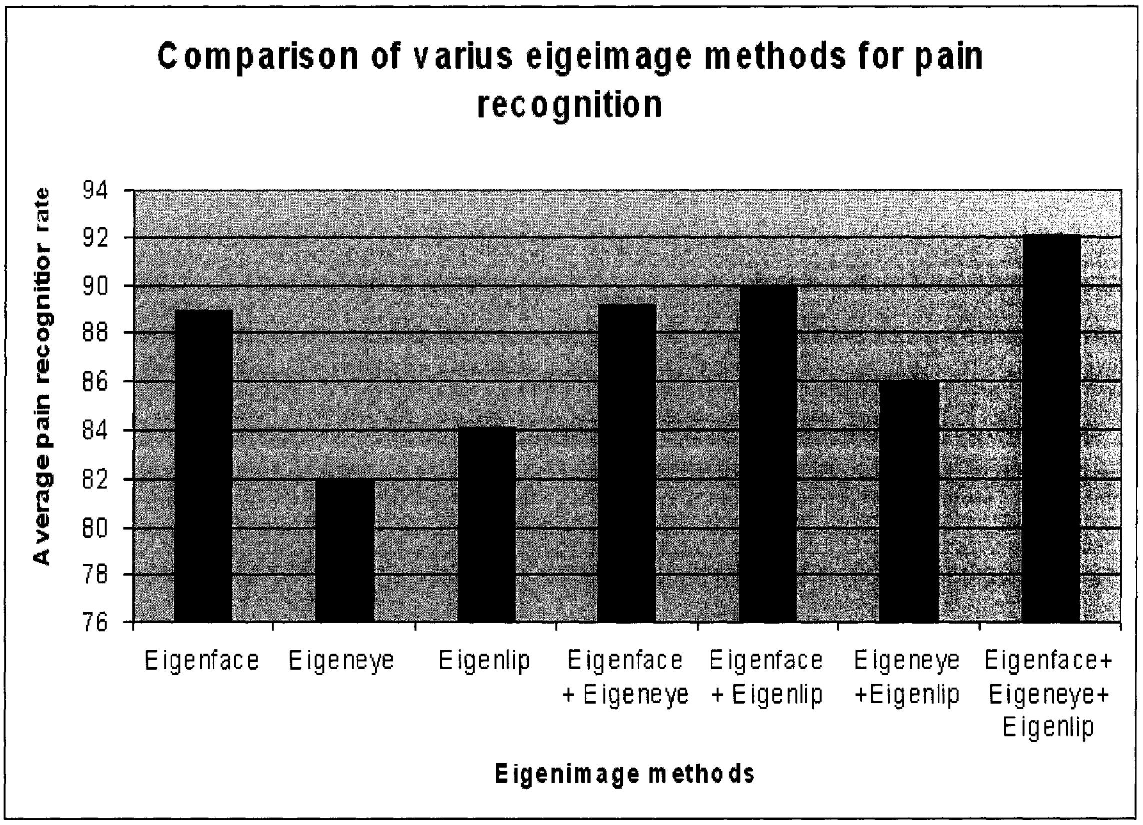


Fig. 5.2. Bar diagram of accuracy results for various eigenimage methods

This section has given us the results of the eigenimage-based pain recognition process. The following section will give the results obtained with multilayer back propagation neural network-based pain recognition technique.

5.4 Results of Neural Network Method

The multilayer neural network with error backpropagation technique is used for pain recognition. In this approach, the errors are backpropagated by the environment through the network from the output units to the input units, and weight and bias updates. The purpose of backpropagation is to adjust the internal state (weights and biases) of the multilayer perceptron so that the multilayer perceptron produces the desired output for the specified input. The general block diagram of neural network-based pain recognition is shown in figure 5.3.

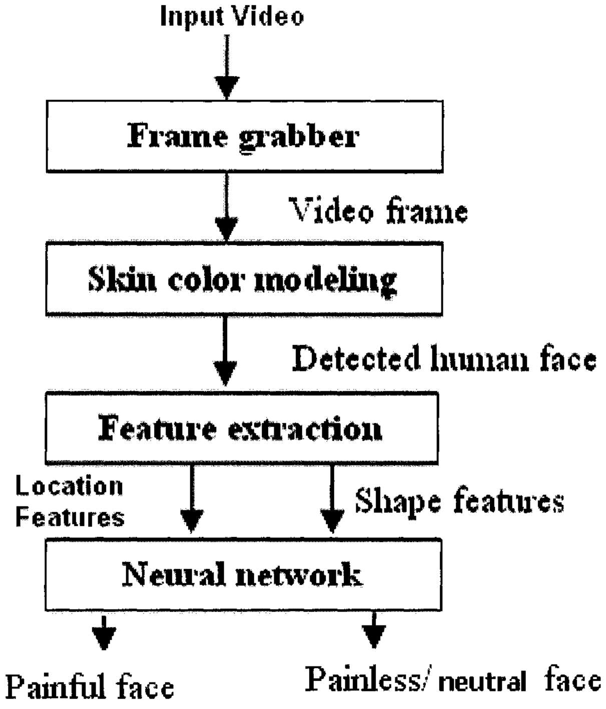


Fig. 5.3. Block diagram of neural network-based pain recognition system

We have used two types of facial features as input to our neural network system. They are location features and shape features. Five location features and sixteen shape features are inputted to the 21-input neural network system. The number of output units in the output layer is two, as there are only two possible outcomes – painful face and no-painful face. Table 5.3 shows the effects of the number of hidden units for 1 hidden layered network with 21 inputs:

Table 5.3. Effect of the number of neurons (in 1 hidden layer) on system accuracy

1 Hidden layer					
Number of neurons	Training Time (min)	Recognition time (min)	Non-painful face recognition rate	Painful face recognition rate	Average accuracy
5	1.01 ± 0.1	0.45 ± 0.04	71% ± 3%	46% ± 4%	59% ± 3%
10	1.56 ± 0.14	0.65 ± 0.09	95% ± 1%	88% ± 3%	92% ± 2%
20	1.89 ± 0.03	0.69 ± 0.02	80% ± 2%	73% ± 1%	76% ± 2%

From the above table, we can see that for 1 hidden layered network, with 10 hidden neurons, the network gives us the best accuracy for both painful and non painful face recognition. With 5 hidden units, the network gives the worst accuracy.

To obtain the best accuracy we have checked our system with 2, 3 and 5 hidden layers. Table 5.4, table 5.5 and table 5.6 show the effect of number of hidden layers and the number of neurons in those hidden layers for 21 inputs on training and recognition time and system accuracy.

Table 5.4. Effect of the number of neurons (in 2 hidden layers) on system accuracy

2 Hidden layer					
Number of neurons	Training Time (min)	Recognition time (min)	Non-painful face recognition rate	Painful face recognition rate	Average accuracy
5	1.07 ± 0.05	0.53 ± 0.02	76% ± 5%	66% ± 3%	71% ± 4%
10	1.90 ± 0.02	0.71 ± 0.08	97% ± 1%	91% ± 2%	94% ± 2%
20	3.56 ± 0.25	0.97 ± 0.04	82% ± 1%	76% ± 3%	78% ± 3%

Table 5.4 shows that for a 2 hidden layered network, the system gives the best performance with 10 neurons in the hidden layers. The training and recognition time with this network setup is 1.90 ± 002 minutes and 0.71 ± 0.08 minutes respectively. The average accuracy is 94% ± 2% for 10 hidden neurons in each hidden layer. It can also be inferred that the training and recognition time are increased with the number of hidden layers and the number of neurons in hidden layers.

Table 5.5. Effect of the number of neurons (in 3 hidden layers) on system accuracy

3 Hidden layer					
Number of neurons	Training Time (min)	Recognition time (min)	Non-painful face recognition rate	Painful face recognition rate	Average accuracy
5	1.98 ± 0.3	0.98 ± 0.09	92% ± 2%	91.68%	92% ± 2%
10	3.16 ± 0.34	1.77 ± 0.21	92% ± 1%	90% ± 1%	91% ± 1%
20	5.89 ± 1.23	1.99 ± 0.67	87% ± 3%	80% ± 4%	83% ± 3%

Table 5.5 shows the simulation results for a three hidden layered network. The system gives the best performance with 5 neurons in each hidden layer. The training and recognition time with this network setup is 1.98 ± 0.3 minutes and 0.98 ± 0.09 minutes respectively. The average accuracy is 92% ± 2% for 5 hidden neurons in each hidden layers.

Table 5.6. Effect of the number of neurons (in 5 hidden layers) on system accuracy

5 Hidden layer					
Number of neurons	Training Time (min)	Recognition time (min)	Non-painful face recognition rate	Painful face recognition rate	Average accuracy
5	2.89 ± 0.3	1.65 ± 0.14	82% ± 3%	75% ± 5%	79% ± 3%
10	5.78 ± 1.01	2.01 ± 0.6	92% ± 1%	89% ± 4%	91% ± 2%
20	9.98 ± 2.31	2.99 ± 0.8	89% ± 1%	90% ± 1%	89% ± 1%

Table 5.6 shows the simulation results for a five hidden layered network. Also here, the system gives the best performance with 5 neurons in each hidden layer. The training and recognition time with this network setup is 2.89 ± 0.3 minutes and 1.65 ± 0.14 minutes respectively. The best average accuracy for 5 hidden layers is 91% ± 2% and it is with the 10 neurons in each hidden layers combination.

From the above tables, it is clear that the system works the best for the neural network with 2 hidden layers and 10 neurons in each hidden layer. Also it can be seen that, the times for training and recognition are proportional to the number of hidden units of the system.

The next section will show the timing and accuracy comparison results between two machine learning techniques.

5.5 Comparison between Eigenimage Method and Neural Network Method

In this section, we compare between the eigenimage method and neural network method. At first in section 5.5.1, results of the speed comparison and then in section 5.5.2, the accuracy comparison result will be shown. For speed and accuracy comparison, the best eigenimage method and the neural network with proper number of hidden layers and hidden units are considered.

5.5.1 Speed Comparison

Processing speed is an important factor from the computational point of view. In this speed comparison process, for the eigenimage method, we consider the combination of eigenface, eigeneye and eigenlip methods, because this combination has given us the best results. For neural network-based method, we consider a network with 2 hidden layers and 10 neurons in each of those layers. Table 5.7 shows the speed comparison results.

Table 5.7. Speed comparison

Machine learning methods	Training time (min)	Recognition time (min)
Eigenimage	3.06 ± 0.56	1.87 ± 0.22
Neural network	1.90 ± 0.02	0.71 ± 0.08

From the above table, it is clear that the neural network-based pain recognition system is faster than the eigenimage-based pain recognition system. The training and recognition time for the neural network-based system are 1.90 ± 0.02 minutes and 0.71 ± 0.08 minutes respectively. Whereas, the training and recognition time for the eigenimage-based system are 3.06 ± 0.56 minutes and 1.87 ± 0.22 minutes respectively.

5.5.2 Accuracy Comparison

Accuracy of a pain recognition system means what percentage of painful and neutral videos are correctly recognized by the system. Timing is not considered in this measurement. For both approaches, 34 neutral videos and the same number of painful videos were considered in accuracy comparison. Table 5.8 shows the accuracy comparison result.

Table 5.8. Accuracy comparison

Machine learning methods	No. of input faces		No. of correctly recognized painful faces	No. of correctly recognized non-painful faces	Accuracy (%)
	Painful	Non-painful			
Eigenimage	34	34	29 ± 2	33 ± 1	91% ± 2%
Neural network	34	34	32 ± 3	33 ± 1	93% ± 2%

From the above table, it can be seen that neural network-based pain recognition system gives us better results than the eigenimage-based pain recognition system. But the difference of the accuracy results of the two methods is very small. The accuracy of the eigenimage-based pain recognition system is 91% ± 2% and the accuracy of the neural network-based pain recognition system is 93% ± 2%.

5.6 Summary

In this chapter, we have first shown the accuracy results of face detection technique using skin color modeling technique. Then we have shown the results of eigenimage-based pain recognition system and neural network-based pain recognition system and then have shown the comparison results between these two methods in terms of their processing speed and accuracy.

The average face detection rate is 90% ± 2% and it does not depend on the number of input videos. From experimental data of Gaussian distribution it is also observed that there is no difference between the chromatic color space for infants and adults.

For eigenimage-based pain recognition, we have used seven different eigenimage techniques. They are eigeneye, eigenlip, eigeneye, combination of eigeneye and eigenlip, combination of eigenface and eigenlip, combination of eigeneye and eigenface and combination of eigenface, eigeneye and eigenlip. Among these methods, eigenface method gives us the best performance which is 89% ± 2% if we use it alone. We also notice the results obtained by combining two or more of these three eigenimage methods. Combination of eigenface and eigeneye methods gives us 89% ± 1%, combination of eigenface and eigenlip methods gives us 90% ± 3% and combination of eigeneye and eigenlip methods gives us 86% ± 4%. But the best result (92% ± 2%) we obtained is by combining eigenface, eigeneye and eigenlip methods.

For neural network-based pain recognition, we have used multilayer backpropagation algorithm. In this approach, the errors are backpropagated by the environment through the network from the output units to the input units, and weight and bias updates. For 1 hidden layered network, with 10 neurons, the network gives us the best accuracy for both painful and non painful face recognition. The system is also checked with more than 1 hidden layer (and with different number of hidden neurons). From the obtained results, it is clear that the system works the best for the neural network with 2 hidden layers and 10 neurons in each hidden layer. Also it can be seen that, the time for training and recognition are proportional to the number of hidden units of the system.

For speed and accuracy comparison between eigenimage and neural network methods, we have considered the best option for both eigenimage and neural network-based method. From the results, we can say that neural network-based method is better in terms of both speed and accuracy than eigenimage-based method.

Chapter 6

Conclusions

In this chapter, we present a summary of contributions, a discussion of limitations, and suggestions for future works.

6.1 Contributions

In this study, we have introduced two machine learning approaches for pain recognition from facial images collected from video sequences.

First, we introduced the architecture and an algorithm for eigenimage-based pain recognition. We have used three eigenimage techniques for this purpose – eigenface, eigeneye and eigenlip. We used these eigenimage techniques individually and collectively to check the performance of the system.

Secondly, we described an implementation of the architecture and the algorithm for multilayer neural network-based pain recognition system. Error back propagation technique is used for learning and recognition. We have tested the system with different number of hidden layers and different numbers of neuron in hidden layers.

Face detection was the first step of our implementation and we have used skin color modeling technique for this purpose. The average skin region detection rate is 88% ± 2%, whereas the average false skin detection rate is 12% ± 3%. The average face detection rate is 90% ± 2%. From experimental data of Gaussian distribution it is also observed that there is no difference between the chromatic color space for infants and adults. Videos of the people of different ethnicities are used as input for this system. But the system's behaviors were unique to all the videos.

The input and testing video files are collected from the video database of Computer Science & Engineering department of University of Rajshahi, Bangladesh. In this database, there are two video files for every person and a total of 34 persons with different colors, ethnicities, ages and genders are considered. In one file, the subject is in neutral mood and in the other, the person is in painful mood due to moving shoulder or moving heads etc.. The resolutions of the videos are 96 X 96. The videos are roughly 1 to 1.5 second long.

Among the three eigenimage techniques, eigenface method gives us the best performance which is 89% ± 2%. We also noticed the results obtained by combining two or more of these three eigenimage methods. Combination of eigenface and eigeneye methods gave us 89% ± 1%, combination of eigenface and eigenlip methods gave us 90% ± 3% and combination of eigeneye and eigenlip methods gave us 86% ± 4%. But the best result (92% ± 2%) we obtained was by combining eigenface, eigeneye and eigenlip methods.

For multi layer neural network-based approach, the system works the best for the neural network with two hidden layers and 10 units in each hidden layer. Also in this case, the time for training and recognition are proportional to the number of hidden units of the system.

Comparing between the eigenimage and neural network-based approach, the second one is faster and also gives us the better accuracy than the first one. The training and recognition time for the neural network-based system are 1.90 ± 0.02 minutes and 0.71 ± 0.08 minutes respectively. Whereas, the training and recognition time for the eigenimage-based system are 3.06 ± 0.56 minutes and 1.87 ± 0.22 minutes respectively. The accuracy of the eigenimage-based pain recognition system is 91% ± 2% and the accuracy of the neural network-based pain recognition system is 93% ± 2%.

6.2 Limitations

In this section, we describe the limitations of the architecture and results presented in this study.

The limitations are followings:

. With skin color modeling technique for face detection, the system cannot give us the best result in the case of videos of fully or partially bald headed people. In those cases, portions of head are detected as face. Also the system has some problem in face detection if the color of the dress of the person in the video is almost similar to skin color.

. The system is not tested with the videos of low resolution and poor quality. So no steps were taken to extract the image from a low quality video. Hence it will not work well in that case.

. The implementation of the neural network approach is limited to layered neural networks with full connections between layers and the algorithm is also limited to error backpropagation.

. The choices of parameters such as numbers of units in hidden layers, learning constants, and momentum are arbitrary, mostly by trial and error.

. The system is implemented in MATLAB, so it is comparatively slower.

. The results were not compared with other facial expression recognition system.

6.3 Future Works

In this section, we provide suggestions for future work in the research area of pain or expression recognition. Even though we have discussed two approaches of pain recognition, there are still items to be explored to make them really useful tools in practical life. These items include the followings:

. Development and implementation of algorithms or methods that work in real-time environment, i.e. in hospital.

. Comparisons with other algorithms and methods that can improve the system in terms of accuracy. It can also enhance the application areas by recognizing all expressions from facial images.

. Choosing a better face detection technique that can work better in all situations and in the case of lower quality videos.

. Testing of other neural network algorithms and different architectures which give us better results.

Bibliography

[1] Beat Fasel and Juergen Luettin (2003), "Automatic Facial Expression Analysis: A Survey", Pattern Recognition, Vol. 36, No. 1, pp. 259-275.

[2] J. J. Weng and D. L. Swets (1999), "Face Recognition", in A. K. Jain, R. Bolle, and S. Pankanti (Editors), BIOMETRICS: PERSONAL IDENTIFICATION IN NETWORKED SOCIETY, Kluwer Academic Press.

[3] S.A. Rizvi, P.J. Phillips, and H. Moon (1998), "A verification protocol and statistical performance analysis for face recognition algorithms", in the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Santa Barbara, USA, pp. 833-838.

[4] V. Bruce (1999), "Identification of Human Faces", Image Processing and Its Applications, Conference Publication No. 465, IEEE, pp. 615-619.

[5] R. Chellappa, C. L. Wilson and S. Sirohey (1995), "Human and Machine Recognition of Faces: A Survey", Proceedings of the IEEE, Vol. 83, No. 5, pp. 705-740.

[6] P. Temdee, D. Khawparisuth, and K. Chamnongthai (1999), "Face Recognition by Using Fractal Encoding and Backpropagation Neural Network", in the proceedings of the 5th International Symposium on Signal Processing and its Applications, ISSPA '99, Australia, pp. 159-161.

[7] J. Huang (1998), "Detection Strategies For Face Recognition Using Learning and Evolution", PhD. Thesis, George Mason University.

[8] R. Brunelli, and T. Poggio (1993), "Face Recognition: Features versus Templates", IEEE Transactions, PAMI, 15(10), pp. 1042-1052.

[9] M. A. Turk and A. P. Pentland (1991), "Face recognition using Eigenfaces", in the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA, pp. 586-591.

[10] L. Sirovich, and M. Kirby (1987), "Low-dimensional Procedure for the Characterization of Human Faces", Journal of the Optical Society of America, Vol. 4, No. 3, pp. 519-524.

[11] M. Kirby, and L. Sirovich (1990), "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp. 103-108.

[12] B. Kepenekci (2001), "Face Recognition Using Gabor Wavelet Transform", MSc. Thesis, Middle East Technical University, Turkey.

[13] B. Moghaddam, and A. Pentland (1995), "An Automatic System for Model-Based Coding of Faces", in the proceedings of the IEEE Data Compression Conference, pp. 362-370.

[14] S. J. Lee, S. B. Yung, J. W. Kwon, and S. H. Hong (1999), "Face Detection and Recognition Using PCA", IEEE TENCON, pp. 84-87.

[15] S. Z. Lee, and J. Lu (1998), "Generalizing Capacity of Face Database for Face Recognition", IEEE, pp. 402-406.

[16] J. L. Crowley, and K. Schwerdt (1999), "Robust Tracking and Compression for Video Communication", in the proceedings of the IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real Time Systems, Corfu, Greece, pp. 2-9.

[17] B. Moghaddam, and A. Pentland (1998), "Beyond Euclidean Eigenspaces: Bayesian Matching for Visual Recognition", FACE RECOGNITION: FROM THEORIES TO APPLICATIONS, pp. 921-930.

[18] B. Moghaddam, T. Jebara, and A. Pentland (2000), "Bayesian Face Recognition", Pattern Recognition, Vol. 33, No. 11, pp. 1771-1782.

[19] B. Moghaddam, and A. Pentland (1995), "Probabilistic Visual Learning for Object Detection", in the proceedings of the 5th International Conference on Computer Vision, Cambridge, MA, USA.

[20] B. Moghaddam, and A. Pentland (1997), "Probabilistic Visual Learning for Object Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7.

[21] B. Moghaddam (2002), "Principal Manifolds and Probabilistic Subspaces

for Visual Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 6.

[22] C. Liu, and H. Wechsler (1998), "A Unified Bayesian Framework for Face Recognition", in the proceedings of the 1998 IEEE International Conference on Image Processing (ICIP '98), Chicago, Illinois, USA, pp. 151-155.

[23] C. Liu, and H. Wechsler (1998), "Probabilistic Reasoning Models for Face Recognition", in the Proceedings of the 1998 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98), Santa Barbara, CA., USA, pp. 827-832.

[24] K. C. Chung, S. C. Kee, and S. R. Kim (1999), "Face Recognition using Principal Component Analysis of Gabor Filter Responses", IEEE, p. 53-57.

[25] K. Etemad, and R. Chellappa (1996), "Face Recognition Using Discriminant Eigenvectors", in the proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, Georgia, USA, pp. 2148-2151.

[26] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman (1997), "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7.

[27] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng (1998), "Discriminant Analysis of Principal Components for Face Recognition", in the proceedings of the International Conference on Automatic Face and Gesture Recognition, Nara, Japan, pp. 336-341.

[28] W. Zhao, R. Chellappa, and N. Nandhakumar (1998), "Empirical Performance Analysis of Linear Discriminant Classifiers", in the Proceedings of the 1998 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98), Santa Barbara, CA., USA, pp. 164-169.

[29] W. Zhao (1999), "Subspace Methods in Object/Face Recognition", in the proceedings of the International Joint Conference on Neural Networks, Washington DC, USA, pp. 3260-3264.

[30] C. Podilchuk, and X. Zhang (1996), "Face Recognition Using DCT-Based

Feature Vectors", in the proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, Georgia, USA, pp. 2144-2146.

[31] R. O. Duda, P. E. Hart, and D. G. Stork (2001), "PATTERN CLASSIFICATION", John Wiley & Sons, 2nd Edition.

[32] S. Eickeler, S. Müller, and G. Rigoll (1999), "High Quality Face Recognition in JPEG Compressed Images", in the proceedings of the IEEE International Conference on Image Processing (ICIP '99), Cobe, Japan, pp. 672-676.

[33] A. V. Nefian, and M. H. Hayes (1998), "Hidden Markov Models for Face Recognition", in the proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, Georgia, USA, pp. 2721-2724.

[34] H. Spies, and I. Ricketts (2000), "Face Recognition in Fourier Space", in the proceedings of "Vision Interface 2000", Montreal, Canada, pp. 38-44.

[35] P. J. Phillips (1999), "Support Vector Machines Applied to Face Recognition", Advances in Neural Information Processing Systems 11, MIT Press, USA, pp. 803-809.

[36] C. S. Bobis, R. C. Gonzalez, J. A. Cancelas, I. Alvarez, and J. M. Enguita (1999), "Face Recognition Using Binary Thresholding for Features Extraction", in the proceedings of the IEEE CIAP '99, pp. 1077-1080.

[37] S. Cagnoni, A. Poggi (1999), "A Modified Modular Eigenspace Approach to Face Recognition", in the proceedings of the IEEE CIAP '99, pp. 490-495.

[38] A. X. Guan, and H. H. Szu (1999), "A Local Face Statistics Recognition Methodology beyond ICA and/or PCA", in the proceedings of the International Joint Conference on Neural Networks, Washington DC, USA, pp. 1016-1027.

[39] A. Martinez (1999), "Face Image Retrieval Using HMMs", in the proceedings of the IEEE Workshop on Content-Based. Access of Image and Video Libraries, Fort Collins, CO, USA, pp. 35-39.

[40] P. Temdee, D. Khawparisuth, and K. Chamnongthai (1999), "Face Recognition by Using Fractal Encoding and Backpropagation Neural Network", in the proceedings of the 5th International Symposium on Signal Processing and its Applications (ISSPA '99), Brisbane, Australia, pp. 159-161.

[41] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. Von der Malsburg, R. P. Wurtz, and W. Konen (1993), "Distortion Invariant Object Recognition in the Dynamic Link Architecture", IEEE Transactions on Computers, Vol. 42, pp. 300-310.

[42] L. Wiskott, J. M. Fellous, N. Krüger, and C. von der Malsburg (1997), "Face Recognition by Elastic Bunch Graph Matching", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 129-132.

[43] M. J. Er, S. Wu, and J. Lu (1999), "Face Recognition Using Radial Basis Function (RBF) Neural Networks", in the proceedings of the 38th Conference on Decision & Control, Phoenix, Arizona USA, pp. 2162-2167.

[44] C. E. Thomaz, R. Q. Feitosa, and A. Veiga (1998), "Design of Radial Basis Function Network as Classifier in Face Recognition Using Eigenfaces", IEEE, pp. 118-123.

[45] A. Martinez (1999), "Face Image Retrieval Using HMMs", in the proceedings of the IEEE Workshop on Content-Based. Access of Image and Video Libraries, Fort Collins, CO, USA, pp. 35-39.

[46] Z. Liposcak, and S. Loncaric (1999), "Face Recognition from Profiles Using Morphological Operations", IEEE Computer Society, ISBN: 0-7965-0378-0, pp. 47-52.

[47] P. Ekman (1994), "Strong Evidence for Universals in Facial Expressions: A Reply to Russell's Mistaken Critique", Psychology Bulletin, Vol. 115, No. 2, pp. 268–287.

[48] C.E. Izard (1994), "Innate and Universal Facial Expressions: Evidence from Developmental and Cross-cultural Research", Psychology Bulletin, Vol. 115, No. 2, pp. 288–299.

[49] P. Ekman and W.V. Friesen (1978), "FACIAL ACTION CODING

SYSTEM: INVESTIGATOR'S GUIDE", Consulting Psychologists Press, Palo Alto, CA, USA.

[50] M.J. Blackm and Y. Yacoob (1995), "Tracking and Recognizing Rigid and Non-rigid Facial Motions using Local Parametric Models of Image Motion", in the proceedings of the International Conference on Computer Vision (ICCV '95), Cambridge, MA, USA, pp. 374–381.

[51] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman and T.J. Sejnowski (1999), "Classifying Facial Actions", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 21, No. 10, pp. 974–989.

[52] I.A. Essa and A.P. Pentland (1997), "Coding, Analysis, Interpretation and Recognition of Facial Expressions", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 757–763.

[53] A. Lanitis, C.J. Taylor and T.F. Cootes (1995), "A Unified Approach to Coding and Interpreting Face Images", in the proceedings of the 5th International Conference on Computer Vision (ICCV '95), Cambridge, MA, USA, pp. 368–373.

[54] J. Lien (1998), "Automatic Recognition of Facial Expressions using Hidden Markov Models and Estimation of Expression Intensity", Ph.D. Thesis, Carnegie Mellon University, USA.

[55] A. Martinez (1999), "Face Image Retrieval using HMMs", in the proceedings of the IEEE Workshop on Content-Based. Access of Image and Video Libraries, Fort Collins, CO, USA, pp. 35–39.

[56] K. Mase (1991), "Recognition of Facial Expression from Optical Flow", IEICE Transaction, Vol. E74, No. 10, pp. 3474–3483.

[57] A. Nefian and M. Hayes (1999), "Face Recognition using an Embedded HMM", in the proceedings of the IEEE Conference on Audio and Video-based Biometric Person Authentication", Washington DC, USA, pp. 19–24.

[58] N. Oliver, A. Pentland and F. Berard (2000), "LAFTER: A Real-time Face and Lips Tracker with Facial Expression Recognition", Pattern Recognition, Vol. 33, pp. 1369–1382.

[59] T. Otsuka and J. Ohya (1997), "Recognizing Multiple Person's Facial

Expressions using HMM Based on Automatic Extraction of Significant Frames from Image Sequences", in the proceedings of the. International Conference on Image Processing (ICIP '97), Washington D C, USA, pp. 546–549.

[60] M. Rosenblum, Y. Yacoob and L.S. Davis(1996), "Human Expression Recognition from Motion using A Radial Basis Function Network Architecture", IEEE Transaction on Neural Network, Vol. 7 No. 5, pp. 1121–1138.

[61] N. Ueki, S. Morishima, H. Yamada and H. Harashima (1994), "Expression Analysis/Synthesis System Based on Emotion Space Constructed by Multilayered Neural Network", Systems Computation, Japan, Vol. 25, No. 13, pp. 95–103.

[62] M. Pantic and L.J.M. Rothkrantz (2000), "Automatic Analysis of Facial Expressions: the State of the Art", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22 No. 12, pp. 1424–1445.

[63] Y. Yacoob and L.S. Davis (1996), "Recognizing Human Facial Expressions from Long Image Sequences using Optical Flow", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 18, No. 6, pp. 636–642.

[64] T. Otsuka and J. Ohya (1997), "A Study of Transformation of Facial Expressions Based on Expression Recognition from Temporal Image Sequences", Technical Report, Institute of Electronic, Information, and Communications Engineers (IEICE).

[65] M. Stifforring, H. J. Andersen and E. Granurn (1999), "Skin Color Detection Under Changing Lighting Conditions", in the proceedings of the 7th Symposium on Intelligent Robotic Systems, Coimbra, Portugal, pp. 20–23.

[66] M.-H. Yang and N. Ahuja (1998), "Detecting Human Faces in Color Images", in the proceedings of the IEEE Conference on Image Processing (ICIP '98), Chicago, Illinois, USA, pp. 127–130.

[67] G. Yang and T. S Huang (1994), "Human Face Detection in Complex Background", Pattern Recognition, Vol. 27 pp. 53–63.

[68] M. J. T. Reinders, P. J. L. van Beek, B. Sankur and J. C. A. van der Lubbe, (1995), "Facial Feature Localization and Adaptation of a Generic Face Model for Model-based Coding", Signal Processing: Image Communication, pp. 57–74.

[69] H. Schneiderman and T. Kanade (2000), "A statistical Model for 3d Object Detection Applied to Faces and Cars", in the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, South Carolina, USA, pp. 746-751.

[70] H. A. Rowley, S. Baluja and T. Kanade (1998), "Neural Network-based Face Detection", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20, No.1, pp. 23–38.

[71] E. Helomas and B. K. Low (2001),"Face Detection: A Survey", Computer Vision and Image Understanding, Vol. 83, pp. 236–274.

[72] K. Sung and T. Poggio (1998), "Example-based Learning for View-based Human Face Detection", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1, pp.39–51.

[73] T. Sakai, M. Nagao and T. Kanade (1972), "Computer Analysis and Classification of Photographs of Human Faces", in the proceedings of the First USA-Japan Computer Conference, pp. 2–7.

[74] I. Craw, H. Ellis and J. R. Lishman (1987), "Automatic Etraction of Face Features", Pattern Recognition Letter, pp. 183–187.

[75] V. Govindaraju (1996), "Locating Human Faces in Photographs", International Journal on Computer Vision, Vol. 19.

[76] A. Jacquin and A. Eleftheriadis (1995), "Automatic Location Tracking of Faces and Facial Features in Video Sequences", in the proceedings of the IEEE International Workshop on Automatic Face and Gesture Recognition.

[77] J. Wang and T. Tan (2000), "A New Face Detection Method Based on Shape Information", Pattern Recognition Letter, Vol. 21, pp. 463–471.

[78] A. L. Yuille, P. W. Hallinan and D. S. Cohen (1992), "Feature Extraction from Faces using Deformable Templates", International Journal on Computer Vision, Vol. 8, pp.99–111.

[79] J. Choi, S. Kim and P. Rhee (1999), "Facial Components Segmentation for Extracting Facial Feature", in the proceedings of the 2nd International Conference on Audio and Video-based Biometric Person Authentication (AVBPA '99), Washington DC, USA.

[80] R. Herpers, K.-H. Lichtenauer and G. Sommer (1996), "Edge and Key-point Detection in Facial Regions", in the proceedings of the IEEE 2nd International Conference on Automatic Face and Gesture Recognition, Killington, Vermont, USA, pp. 212–217.

[81] T. S. Jebara and A. Pentland (1997), "Parameterized Structure form Motion for 3D Adaptive Feedback Tracking of Faces", in the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97), San Juan, Puerto Rico, pp. 144–150.

[82] S. Satoh, Y. Nakamura and T. Kanade (1999), "Name-It: Naming and Detecting Faces in News Videos", IEEE Multimedia, Vol. 6, pp. 22–35.

[83] J. L. Crowley and F. Berard (1997), "Multi-model Tracking of Faces for Video Communications", in the proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.

[84] N. Oliver, A. Pentland and F. Berard (2000), "A Real-time Face and Lips Tracker with Facial Expression Recognition", IEEE Transaction on Pattern Recognition, Vol. 33, pp.1369–1382.

[85] S. Kim, N. Kim, S. C. Ahn and H. Kim (1998), "Object Oriented Face Detection using Range and Color Information", in the proceedings of the 3rd International Conference on Automatic Face and Gesture Recognition, Nara, Japan, pp. 76–81.

[86] R. Kjedsen and J. Kender (1996), "Finding Skin in Color Images", in the proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition, Killington, Vermont, USA, pp. 312–317.

[87] K. Sobottka and I. Pitas (1996), "Extraction of Facial Regions and Features using Color and Shape Information, in the proceedings of the International Conference on Pattern Recognition, Vienna, Austria.

[88] H. Wang and S.-F Chang (1994), "A Highly Efficient System for

Automatic Face Region Detection in Mpeg video", IEEE Transaction on Circuits and Systems for Video Technology, pp. 615–628.

[89] Q. Chen, H. Wu and M. Yachida (1995), "Face Detection by Fuzzy Matching", in the proceedings of the 5th IEEE International Conference on Computer Vision, Cambridge, MA, USA.

[90] J. Cai and A. Goshtasby (1999), "Detecting Human Faces in Color Images", Image and Vision Computation, Vol. 18, pp. 63–75.

[91] S. Kawato and J. Ohya (2000), "Real-time Detection of Nodding and Head-shaking by Directly Detecting and Tracking Between the Eyes", in the proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France..

[92] A. Albiol, C. A. Bouman and E. J. Delp (1999), "Face Detection for Pseudo-semantic Labeling in Video Databases", in the proceedings of the International Conference on Image Processing, Kobe, Japan.

[93] J. Yang and A. Waibel (1996), "A Real-time Face Tracker", in the proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), Sarasota, Florida, USA, pp. 142-147.

[94] M. Turk and A. Pentland (1991), "Eigenfaces for Recognition", Cognitive Neuroscience, Vol. 3, No. 1, pp. 71–86.

[95] F. Luthon and M. Lievin (1997), "Lip Motion Automatic Detection", in the proceedings of the Scandinavian Conference on Image Analysis.

[96] S. McKenna, S. Gong and H. Liddell (1995), "Real-time Tracking for an Integrated Face Recognition System", in the proceedings of the Workshop on Parallel Modeling of Neural Operators, Faro, Portugal,.

[97] J. Miao, B. Yin, K.Wang, L. Shen and X. Chen (1999), "A Hierarchical Multi-scale and Multi-angle System for Human Face Detection in a Complex Background using Gravity-center Template", Pattern Recognition, Vol. 32, pp. 1237–1248.

[98] Y. H. Kwon and N. da Vitoria Lobo (1994), "Face Detection using Templates", in the proceedings of the International Conference on Pattern Recognition, pp. 764–767.

[99] A. Lanitis, C. J. Taylor and T.F Cootes (1995), "An Automatic Face Identification System using flexible Appearance Models", Image and Vision Computing, Vol. 13, pp. 393–401.

[100] L. Sirovich and M. Kirby (1987), "Low-dimensional Procedure for the Characterization of Human Faces", Journal of the Optical Society of America, Vol. 4, pp. 519–524.

[101] A Pentland, B. Moghaddam and T. Strarner, (1994), "View-based and Modular Eigenspaces for Face Recognition", in the proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 84–91.

[102] B. Moghaddam and A. Pentland (1997), "Probabilistic Visual Learning for Object Representation", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No. 1.

[103] E. Viennet and F. Fogelman Souli´e (1998), "Connectionist Methods for Human Face Processing", FACE RECOGNITION: FROM THEORY TO APPLICATION, Springer-Verlag, Berlin/New York.

[104] H. A. Rowley (1999), "Neural Network-based Face Detection", PhD thesis, Carnegie Mellon University, USA.

[105] S.-H. Lin, S.-Y. Kung and L.-J. Lin (1997), "Face Recognition/Detection by Probabilistic Decision-based Neural Network", IEEE Transaction on Neural Networks, Vol. 8, pp. 114–132.

[106] D. Roth, M.-H. Yang and N. Ahuja (2000), "A Snow-based Face Detector", Advances in Neural Information Processing Systems, Vol. 12.

[107] A. J. Colmenarez and T. S. Huang (1997), "Face Detection with Information-based Maximum Discrimination", in the proceedings of the IEEE International Conferene on Computer Vision and Pattern Recognition.

[108] C. Papageorgiou, M. Oren and T. Poggio (1998), "A General Framework for Object Detection", in the proceedings of the 6th International Conference on Computer Vision, pp. 555–562.

[109] E. Osuna, R. Freund and F. Girosi (1997), "Training Support Vector Machines: An Application to Face Detection", in the proceedings of the

IEEE International Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico.

[110] V. Kumar and T. Poggio (2000), "Learning-based Approach to Real Time Tracking and Analysis of Faces", in the proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France.

[111] H. Schneiderman and T. Kanade (1998), "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition", in the proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.

[112] R. O. Duda, P. Hart and D. G. Stork (2001), "PATTERN CLASSIFICATION", Wiley- Interscience, 2nd edition.

[113] Crowley, J. L. and Coutaz, J. (1997), "Vision for Man-Machine Interaction," Robotics and Autonomous Systems, Vol. 19, pp. 347-358.

[114] J. Cai and A. Goshtasby (1999), "Detecting Human Faces in Color Images", Image and Vision Computation, Vol. 18, pp. 63–75.

[115] G. Wyszecki and W.S. Styles (1982), "COLOR SCIENCE: CONCEPTS AND METHODS, QUANTITATIVE DATA AND FORMULAE", 2nd edition, John Wiley & Sons, New York, USA.

[116] Y. Gong and M. Sakauchi (1995), "Detection of Regions Matching Specified Chromatic Features", Computer Vision and Image Understanding, Vol. 61, No. 2, pp. 263–269.

[117] J. Yang and A. Waibel (1996), "A Real-time Face Tracker", in the proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), Sarasota, Florida, USA, pp. 142-147.

[118] J. J. Hopfield (1982), "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", in the proceedings of the Natural Acadmic Science, Vol. 79, pp. 2554–2558.

[119] G. E. Hinton and T. J. Sejnowski (1986), "Learning and Relearning in Boltzmann Machines", Parallel Distributed Processing, Vol. I, chap. 7, the MIT Press, USA.

[120] D. E. Rumelhart, G. E. Hinton and R. J. Williams (1986), “Learning Internal Representations by Error Propagation”, Parallel Distributed Processing, Vol. 1, chap. 8. the MIT Press, USA.

[121] J. Yang, R. Stiefelhagen, U. Meier and A. Waibel (1998), “Real-time Face and Facial Feature Tracking and Applications”, in the proceedings of the Auditory-Visual Speech Processing (AVSP ‘98), New South Wales, Australia.

[122] Yingli Tian and Lisa Brown (2003), “Real World Real-time Automatic Recognition of Facial Expressions”, in the proceedings of the IEEE Workshop on Performance Evaluation of Tracking and Surveillance, Graz, Austria.

[123] Jeffrey Cohn and T. Kanade (2006), “Use of automated facial image analysis for measurement of emotion expression”, The Handbook of Emotion Elicitation and Assessment, Oxford University Press Series in Affective Science, J. A. Coan & J. B. Allen, ed..