LOCAL BINARY PATTERN BASED FEATURE EXTRACTION AND CLASSIFICATION OF FACIAL EXPRESSIONS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree
of

MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

By

KEHAR SINGH

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled "**LOCAL BINARY PATTERN BASED FEATURE EXTRACTION AND CLASSIFICATION OF FACIAL EXPRESSIONS**" towards the partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science and Engineering** submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand (India) is an authentic record of my own work carried out during the period from July 2010 to June 2011, under the guidance of **Dr. D. Toshniwal, Assistant Professor,** Department of Electronics and Computer Engineering, IIT Roorkee.

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

Date: 29 June 2011

Place: Roorkee

(KEHAR SINGH)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 29 June 2011

Place: Roorkee

(Dr. D. Toshniwal)

Assistant Professor

Department of Electronics and Computer Engineering

IIT Roorkee.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my heartfelt gratitude to my guide and mentor **Dr. D. Toshniwal**, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for her invaluable advices, guidance, encouragement and for sharing her broad knowledge. Her wisdom, knowledge and commitment to the highest standards inspired and motivated me. She has been very generous in providing the necessary resources to carry out my research. She is an inspiring teacher, a great advisor, and most importantly a nice person.

I also wish to thank Vivek Yadav for his valuable suggestions and timely help regarding the domain knowledge and datasets. I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample moral supports and valuable suggestions.

On a personal note, I owe everything to the Almighty and my parents. The support which I enjoyed from my father, mother and other family members provided me the mental support I needed.

KEHAR SINGH

Abstract

Study of facial expression is a challenging problem because of facial expressions of different people are similar but not same for any particular emotion. Data mining can be applied for various tasks related to the analysis of facial expression after proper preprocessing of the images. Classification of facial expression is an important data mining technique and has many applications.

Most of the existing methods of facial expression classification are based on Facial Action Coding System (FACS). The FACS is based on large number of Action Units (AUs) which are used independently of in combination form. Thus FACS encoding is very complex. Hence in the present work, Local Binary Patterns (LBP) have been used for classification of facial expressions as they make use of limited number of bins for feature extraction from images. LBP uses spatial features or micro patterns such as spots, flat areas, corner and lines etc.

We propose a new approach for facial expression classification that is based on extraction of attributes using the rotational invariant local binary pattern (RIULBP) and relative intensity in pixel groups of images. The features have been reduced using Principal Component Analysis (PCA). After reducing these features, we have used Adaptive Neuro Fuzzy Inference System (ANFIS) for classification of the facial expressions images. This approach combines the advantage of reduced and simplified process of feature extraction and ANFIS. Secondly features based on spatial aspects as well as intensity both are used together and improves the results.

The experiment of this work has been done on Japanese Female Facial Expression (JAFFE) data set. Results show that the proposed RIULBP and intensity based feature extraction method and ANFIS based classification method is robust and can be used on diverse application.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction and Problem Statement

In the following sections, a brief introduction and motivation for undertaking the present study and the problem statement for the report has been included.

1.1 Introduction

With the rising of technology, communication between people and machine is growing up. It is also said that a picture can express the meanings effectively more than thousands of words. So if machine are so much intelligent that can understand the expression of human, their feelings and emotions then the communication between machines and human would be more realistic. That would be a strong step in the human's artificial world creation.

Expressions are an elementary way to express human emotions and an effective method of non-verbal communication [1]. Different emotions are expressed in different ways and from different part of organs. For example if a person is surprised or fear his heart beat grown up and his facial expression can be noticed which shows us how much excited or angry he is on us, if we are responsible to surprised him/her. Facial expressions play a significant role in emotion recognition. Automatic and real-time facial expression system helped for many real-life applications such as games, virtual avatars, interactive TV, human-computer interaction, behavioral research, chat programs, video conferencing and various vision systems.

It is widely accepted from psychological theory that human emotions can be classified into six typical emotions: surprise, fear, disgust, anger, happiness, and sadness. These categories are a reflection of the six primary expressions identified by Ekman and Friesen [2].

Facial expression classification has been a focused theme in computer vision research for over three decade. Abundant literature has been dedicated to this research area and incredible progress has been made. There are two main goals of facial expression analysis:

1. Identifying the pattern inside the facial image represented by the different facial expressions.
2. Classification of facial expression/identification of emotion using the pattern obtained from image representing facial expression.

Both of these goals require that the pattern of facial images is identified and describe so that facial expressions are separable. Once the pattern is recognized, we can classify the facial images. Thus, identification of pattern/feature from images one of the major tasks in data mining known as feature extraction. The feature extraction from images is useful in identification of emotions and further classification makes the communication more realistic.

Local patterns in images are the natural way to represent the image. Local patterns such as lines, spot, flat areas etc. are a great deal in image analysis [3]. These local patterns represent the spatial feature in images. Images are made by thousands of pixels and each pixel represents the different color value. In a broad way each pixel is a feature attribute which represent to the image. So image analysis is a challenging and broad field of research.

1.2 Motivation

Facial expression classification is a classic example of a problem that is relatively easy for humans to solve but difficult for computers. In facial expression classification, it is assumed that each expression has some pattern and an emotion represents similar facial expression by every people. Under this assumption facial expression classification is under study. But an emotion represents similar facial expression but not same it could be different person to person. So the motivation is to propose an approach which uses human kind of interpretation. So the idea behind using local binary pattern and ANFIS utilize these 3 facts:

1. Local Binary Pattern is that it represents the spatial feature flawlessly such as lines, corner, flat area, spots etc.
2. Local Binary Pattern histogram represents the image feature in such a way that the pixel attributes are reduce drastically and can be represented the image in features into limited bins.

3. ANFIS is a hybrid technique which has capability of interpretation like human and used in many classification problem due to high accuracy.

Thus, the local binary patterns are extracted with the motivation to represent the pattern of different facial expression and use the ANFIS classification model to classify facial expression.

1.3 Statement of the Problem

The problem statement for the present work can be stated as follows:

"To extract local binary pattern based features from facial expression images and use them for purpose of classification."

The above problem statement can be divided into the following sub problems:

- *To propose a generic framework for facial expression classification.*
- *To extract features from images using local binary pattern and intensity.*
- *To reduce the feature space using PCA.*
- *To classify the facial expression images based on extracted features..*
- *To evaluate the effectiveness of proposed method.*

The assumption has been made that each emotion is represented by different facial expressions.

1.4 Organization of the Report

This dissertation report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the report is organized as follows.

Chapter 2 gives the background of facial expression classification study; discuss some of the well known concepts for feature extraction and classification approaches uses in facial expression classification and brief literature review of existing techniques for emotion recognition. In our framework we uses Rotational Invariant Uniform Local

Binary Pattern (RIULBP) and Adaptive Neuro Fuzzy Inference System (ANFIS), hence a brief review about them is presented, including the research gaps.

Chapter 3 describes the methodology designed for classification of facial expression, the major steps and process involved, their functionality and need in the system.

Chapter 4 gives the implementation details of the proposed methodology.

Chapter 5 gives the description of experimental dataset used, discusses the results that were obtained using our method in different scenario for classification using the facial expressions images of JAFFE dataset and compare the results with some existing techniques.

Chapter 6 concludes the dissertation work and gives suggestions for future work.

Chapter 2

Background and Literature Review

Facial expression classification commonly known as emotion recognition can be processed information which cannot convey through speech or keyboard typing etc. Interaction between computers and human beings will be more natural if computers are able to perceive and respond to human non-verbal communication such as hand gestures, facial expression and tone of the voice, which are used to express feelings and give feedback. Emotions are expressed by different organs of person in which face plays substantial role. So most of the research work found in the facial expression classification. Facial expression classification is two step process: facial feature extraction and classification. In some research work researchers add automatic face detection algorithm and says automatic facial expression classification. Popular techniques which are used in facial expression classification process are discussed in subsequent sections under facial feature extraction and classification.

2.1 Facial Feature Extraction

Facial feature extraction is an important part of facial expression classification. An effective feature extraction can improve the performance of the system significantly. Feature extraction methods can be distinguished according to the imagery data such as feature extraction from the static image and feature extraction from the image sequence [1]. In this section we have discussed some popular techniques under both categories which are used in literature.

2.1.1 Facial Action Coding System Based Feature Extraction

The Facial Action Coding System (FACS), is a system developed by Paul Ekman *et. al.* as a way to measure visible facial muscle movement [4]. This system locates individual features instead of whole face. Several studies in facial emotion recognition have used this technique for feature extraction [5-8]. In this technique facial muscle movements have been coded into facial action units "AU". Facial movement can occur as a single unit, or more often occur in combination with other muscle movements. Liao uses this approach for facial expression classification [9] Some action units are shown in the Figure 2.1.

There are 44 basic AUs which can be used independently or combination to describe almost any expression. Tian *et. al.* worked to recognize AUs [10] and reported good accuracy. These AUs are best suitable to be used in studies on facial expression recognition. Any possible facial expressions can be described by these 44 basic AUs. To describe the facial expression FACS tables were derived by studies of the human face. Once facial images are encoded using the appropriate AUs they are used to train a classification system.

A disadvantage to using this approach for feature extraction is that identification of action unit is complex itself and method takes over 100 hours of training to achieve minimal competency for a human expert [11]. In further research work for emotional features identification and extraction Ekman developed the Emotion Facial Action Coding System (EMFACS). This framework builds upon FACS by targeting specifically those areas of the face directly related to emotional expression. Emotional FACS (EMFACS) specify the relation between facial expressions and emotions.

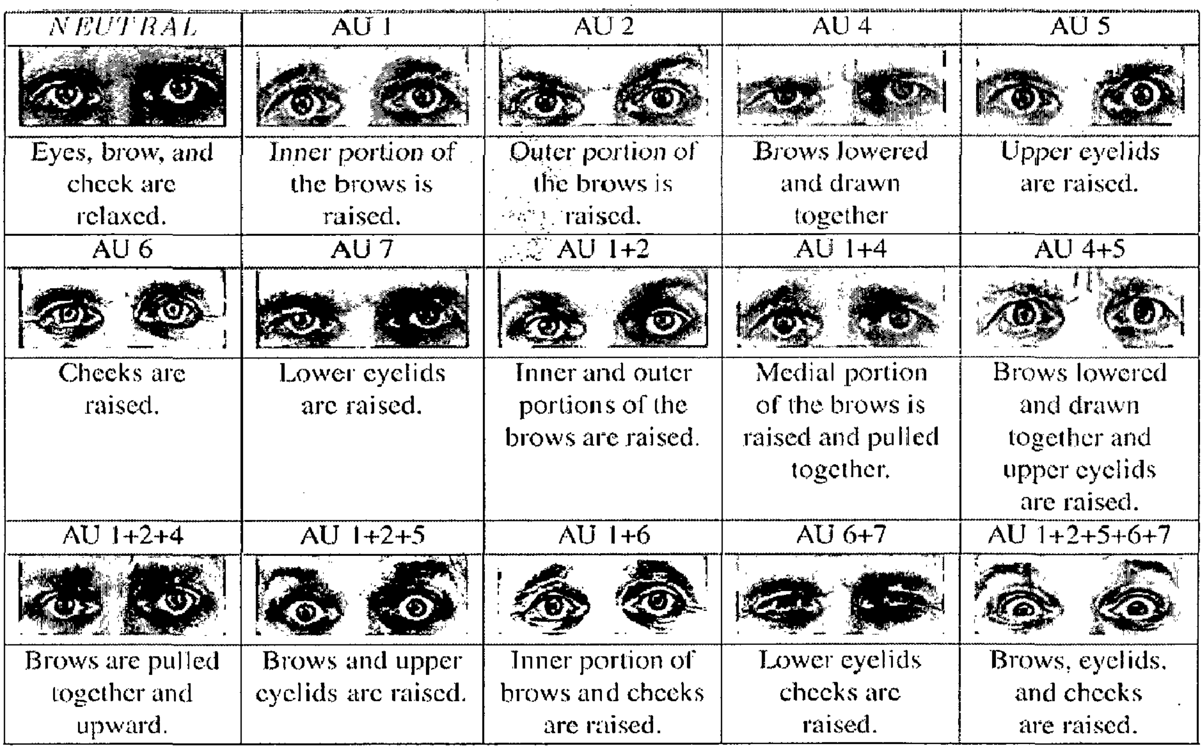

Figure 2.1 Description of action units (AUs) and corresponding facial expressions

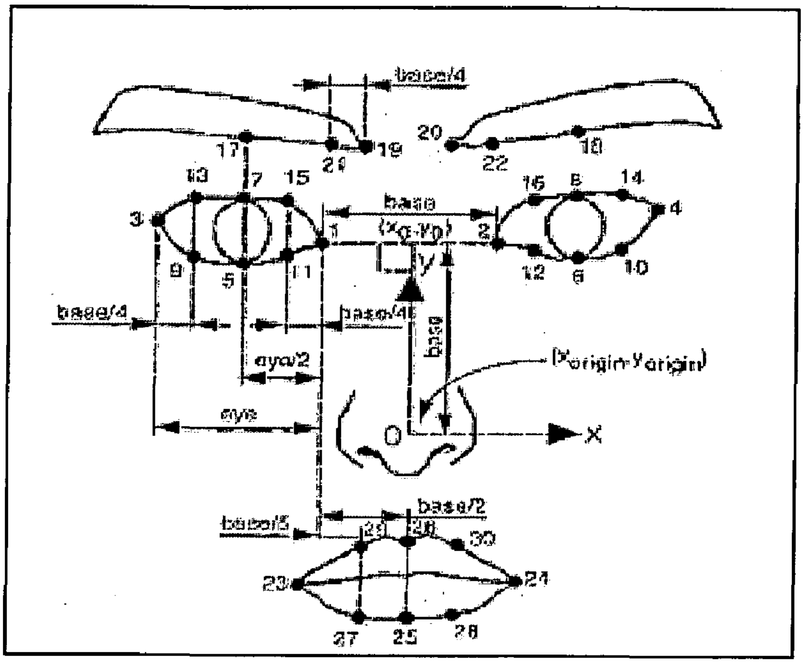

Figure 2.2 Facial characteristic points

2.1.2 Feature Based Method for Feature Extraction

In the earlier work, Kobayashi and Hara proposed a geometric face model of 30 facial characteristic points [12, 13] as shown in Figure 2.2. They normalize an input image by using an affine transformation so that the distance between irises becomes 20 pixels. From the distance between the irises, the length of the vertical lines is empirically determined.

Pantic proposed a point-based model collected of two 2D facial views, the frontal and the side view [14]. The frontal-view face model is composed of 30 features, from these, 25 features are defined in respect with a set of 19 facial points and the rest are some specific shapes of the mouth and chin. Pantic utilized side-view face model consists of 10 profile points correspond to the peaks and valleys of the curvature of the profile curve function.

The effect of identification of geometrical features is based upon the accuracy of extracting the reference points. It is difficult to achieve good accuracy in case of low quality and complex background images. At the same time, Extraction of geometrical features ignores the information of other parts of the face such as skin texture changes, therefore, it is difficult to distinct the expression of the subtle changes.

2.1.3 Texture Based Feature Extraction

Recently, texture based feature extraction from the images are widely used. Examples of such techniques are Gabor filter. Gabor wavelet-based methods are widely used in facial expression feature extraction. It can detect multi-scale, multi-direction texture changes, as well as by the effects of lightning condition in poses is relatively small. In 2006, Yu *et. al.* proposed a method which can identify the features automatically in a genetic programming-based approach that uses Gabor wavelet representation for primitive features and linear/nonlinear operators to produce new features [15]. Liu and Zhang Gabor transform is combined with the hierarchical histogram to extract facial expressional features, which can hierarchically represent the change of texture in local area, and thus capture the natural facial features involved in facial expression analysis [16]. In addition, Gabor transform is relatively less sensitive to the change of lighting conditions so less affected from certain rotation and deformation of images. All these properties make this scheme fairly robust in various conditions and more powerful than traditional representation scheme using 1-D Gabor coefficients. In addition, Wang *et. al.* used similar rectangular Harr wavelet features are used in facial expression feature extraction [17] and proposed a techniques for facial expression recognition with Adaboost. Xiao-Bo *et. al.* proposed multichannel filter based Gabor wavelet [18] in an application study on facial expression recognition.

2.1.4 Local Binary Pattern Based Feature Extraction

Another texture based feature extraction method is local binary pattern (LBP) initially used by Ojala *et. al.* for texture analysis [19]. In further study extension of basic LBP has been proposed such as uniform local binary pattern and rotational invariant local binary pattern [20]. Because of its simplicity in comparison to Gabor filter and less attributes it is widely used in place of Gabor filter in many research work of facial feature extraction [3, 21, 22]. Zu-li used local binary pattern and PCA for feature extraction [21] from facial images. Gomathi used RIULBP for feature extraction for facial expression and got good results [23]. Wencheng *et. al.* [24] and Koutlas *et. al.* [25] also used LBP in his work with slight modification for feature extraction. It maps the attributes of image into few robust attributes. Thus many recent work uses in facial expression uses LBP. Gomathi Another variation of LBP is introduced by Zhao *et. al.* as volume local binary pattern (VLBP) and local binary pattern from three orthogonal planes (LBP-TOP) [26] are used for dynamic texture recognition in study of facial expression recognition in image sequences. Rotational invariant uniform local binary pattern extraction approach is given in detail in this section.

The LBP operator labels to each pixel by a value which is evaluated by using its neighborhood surrounding pixel. Initially LBP operator is defined on 3 × 3 pixel window. To obtain LBP label of center pixel there is 8 neighbor pixels surrounding it, see in Figure 3.6.

Later to remove this limitation of LBP operator, it is extended on any radius *R* and any neighborhood pixels *P* as shown in Figure 3.6 using the bilinear interpolation of pixel values. So that gray value of any pixel when applying any radius can be calculated by interpolation. Coordinates of interpolated p^{th} pixels ($p=0, 1, \ldots, P-1$) can be calculated by Eq. 3.1.

$$x = R \times cos(\tfrac{2\pi p}{P}), \; y = -R \times sin(\tfrac{2\pi p}{P}) \qquad (2.1)$$

where *R* is any radius, *P* is the number of neighborhood pixels of center pixels.

For example consider $P=8$ and $R=2$, in this value of pixels which are on horizontal line and vertical line can directly taken but pixel which are on diagonal their value should be interpolated. Gray value of left upper corner pixel is can be obtained by the formula:

$$g = w_1 \times g_{(x-2,y-2)} + w_2 \times g_{(x-2,y-1)} + w_3 \times g_{(x-1,y-2)} + w_4 \times g_{(x-1,y-1)} \qquad (2.2)$$

where x, y are coordinate of center pixel, w_1, w_2, w_3, w_4 are weight, g is gray value of interpolated pixel. Here in our example for $P=8$ and $R=2$ weights are as $w_1= 0.1716$, $w_2 = 0.2426$, $w_3 =0.2426$, $w_4 = 0.3431$.

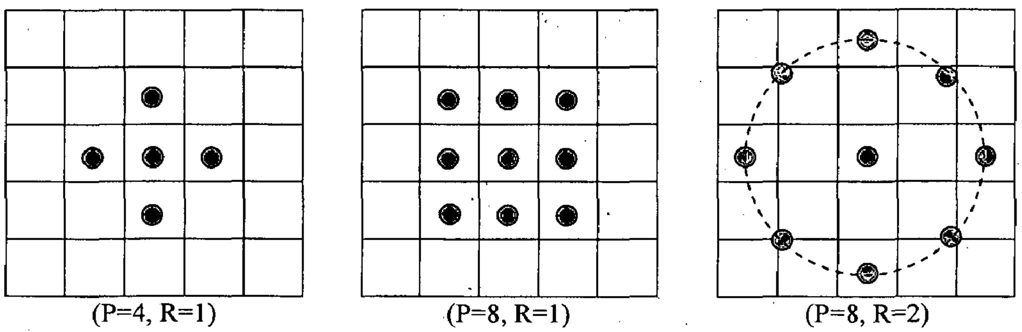

Figure 2.3 Different Neighborhood pixels P on different radius R

The pixel which has to be label is considered as center pixel and after comparing the variations in intensity (gray value) of from neighborhood pixel a value is calculated as shown in Figure 3.7. LBP texture can be defined for a central pixel g_c with respect to its local P neighborhood pixels as –

$$LBP_{P,R} = \sum_{i=0}^{P-1} s\left(g_i - g_c\right) \times 2^i \qquad (2.3)$$

and

$$s(x) = \begin{cases} 1, & if\ x \geq 0 \\ 0, & otherwise \end{cases} \qquad (2.4)$$

where P is the number of neighborhood pixels, R is radius in terms of pixel, g_c is the gray value of central pixel and (g_0, g_1, . . ., g_{p-1}) is the gray value of surrounding pixels, $s(x)$ is a binary function.

So applying the LBP we get 256 bins of LBP histogram. Another extension to the original LBP operator is called uniform pattern. A LBP is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. Number of transitions can be calculated by the formula:

$$T(LBP_{P,R}) = \left|s(g_{P-1} - g_c) - s(g_0 - g_c)\right| + \sum_{i=1}^{P-1} \left|s(g_i - g_c) - s(g_{i-1} - g_c)\right| \quad (2.5)$$

where, T is number of bit transition 0 to 1 or vice versa , P is number of neighborhood pixels, g_c is the gray value of central pixel and g_i is the gray value of i^{th} neighborhood pixel.

110	100	140
125	130	145
120	148	142

0	0	1
0		1
0	1	1

Binary: 00111100

Decimal: 60

Figure 2.4 Calculation of label value of any pixel using LBP

For example 00001100, 01111000, 11111110 are uniform pattern because there is 2 bits change and 11100011, 11100010 are not uniform because have 4 bit transition. Ojala *at al.* noticed 90% of pattern found when $LBP_{8,1}$ operator is applied and 70% uniform pattern found in when $LBP_{16,2}$ is used [20]. Thus uniform patterns reduced 256 bins into 58 bins.

Another extension of LBP operator is Rotational Invariant LBP [3, 20]. When the image is rotated, the total number of 1s and 0s in the neighbor set remains the same but the output binary value gets changed. To remove the effect of rotation, i.e. to assign a unique identifier to each rotation invariant local binary pattern uses the equation.

$$LBP_{P,R}^{riu2} = min\{ROR(LBP_{P,R}, i) | i = 0, 1, \ldots, P-1\} \tag{2.6}$$

where, ROR(x,i) is a function which circular right shift i^{th} times the bits of x. Using the rotation invariant uniform local binary pattern, any pixel intensity can be represented in nine bins i.e. 00000000, 00000001, 00000011, 00000111, 00001111, 00011111, 00111111, 01111111 and 11111111. Thus RIULBP further reduce the 58 bins into 9 bins now there are 9 bins of RIULBP and 1 bin is for others patterns, total 10 bins we have. Instead of calculating rotation and then assign minimum value we can directly assign each pattern into particular bin by following formula:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_i^{P-1} s(g_i - g_c), & if\ T(LBP_{P,R}) \leq 2 \\ P+1, & otherwise \end{cases} \tag{2.7}$$

where, function T represents the number of transition in LBP as given in Eq. 3.5. The notation of RIULBP is as $LBP_{P,R}^{riu2}$ where subscript value presenting number of neighbor pixels and radius. And in superscript *ri* represents to the rotation invariant and *u2* for uniform binary pattern. Generally RIULBP collect the information about the number of spots, lines, flat, corner areas in images.

2.2 Feature Reduction

Feature reduction transforms the data in the high-dimensional space to a space of fewer dimensions. The data transformation may be linear, as in principal component analysis (PCA) [27, 28], but many nonlinear dimensionality reduction techniques are also exist.

The main linear technique for dimensionality reduction, principal component analysis, performs a linear mapping of the data to a lower dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the correlation matrix of the data is constructed and the eigenvectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. The first few eigenvectors can often be used as per requirement and tolerance in accuracy. The original space (with higher dimension of the number of points) has been reduced (with data loss, but hopefully retaining the most important variance) to the small space by a few eigenvectors another linear technique is Linear discriminant analysis (LDA). LDA attempts to express one dependent variable as a linear combination of other features or measurements [29, 30]. LDA is closely related to principal component analysis (PCA) and factor analysis in that both look for linear combinations of variables which best explains the data [31]. LDA explicitly attempts to develop the difference between the classes of data.

There are another nonlinear Dimension reduction techniques are available such as Kernel PCA, locally linear embedding (LLE), Hessian LLE, Laplacian eigenmaps, and LTSA. These techniques construct a low-dimensional data representation using a cost function that retains local properties of the data [32].

2.3 Facial Expression Classification

Classification is the process of categorizing the data into some known class-labels. It can be defined as the task of training a model that will assign class label to the data after mapping attribute set to one of the predefined class labels [33, 34]. It is a supervised process since the class labels are known in advance. The classification model is built on the basis of the training data set provided. After training, classifier can predict the class label to unseen data. In facial expression classification, following classification approaches are used frequently used in research work.

2.3.1 Nearest Neighbor Classifier

A NN-classifier [33, 35] represents all the training data objects in n-dimensional space where n is the number of attributes in the data. The class label of an experimental data is computed on the basis of its closeness to the data points in the training dataset. The k-nearest neighbors are chosen for this purpose. Hence this scheme is also called the k-NN classification scheme. This approach thus requires a closeness measure and a classification function that gives the predicted class based on the closeness. Euclidian distance mostly uses to calculate closeness between data points.

The k-NN classifiers offer several advantages. Model building is not needed and can produce arbitrary shaped decision boundaries. But limitation is that classifier tries to predict the class label of an unseen record only on the basis of information supplied at the time of training (local information).

2.3.2 Naïve Bayes Classifier

The Naïve Bayes classifier makes the estimation of the conditional probability of a class. It assumes that given a class label, the attributes are conditionally independent. To classify a data, the naïve Bayes classifier computes the posterior probability for each class Y as follows [33, 36, 37].

$$P\,(Y/X) = \frac{P(Y)\prod_{i=1}^{d} P(\frac{X_i}{Y})}{P(X)} \tag{2.8}$$

The $P(X)$ is prior probability for Y, and we choose the class that maximizes $P(Y)\prod_{i=1}^{d} P(\frac{X_i}{Y})$.

Naïve Bayes classifiers have several advantages. They are robust to isolated noise points because such points are averaged out when estimating conditional probabilities from data. But the limitation is that they assume conditional independence in the attributes.

2.3.3 Support Vector Machine (SVM)

A popular classification approach is based on SVM [33, 38]. A SVM is based on maximal margin hyper-plane in order to make sure that their worst-case generalization errors are minimized. It uses a nonlinear mapping to transform the original training data into a higher dimension. Then it searches for the linear optimal separating decision boundary within this new dimension. A linear SVM classifier is based on a linear decision boundary and non-linear SVM is used non-linear decision boundaries. It assumes that data from two classes can always be separated using a suitable nonlinear function to enough high dimension. *Support vectors* (training tuples) and *margins* (defined by support vectors) are used to find this hyper plane. The training time for SVMs can be quite large but the accuracy is very high. They are much less prone to over fitting than other methods.

2.3.4 Artificial Neural Network (ANN) Based Classifier

Artificial neural network (ANN), also known as neural network (NN) is a mathematical model motivated from biological neurons [39, 40]. A neural network consists of an interconnected group of artificial neurons. It processes information using a mathematical function, connection weight and threshold value associated with each neuron. There are various ANN models proposed - perceptron and multilayer feed forward ANN. ANNs have at least one hidden layer. They offer many advantages such as they can be used to approximate any function [41]. But ANNs also suffer from some limitations like they are quite sensitive to noise. In absence of appropriate network topology, they tend to become over-fitted. How to decide topology of ANN is itself a problem. Another limitation of ANNs is training can be time consuming if the number of hidden nodes is large.

2.3.5 Adaptive Neuro Fuzzy Inference System (ANFIS)

ANFIS is an ingenious technique to build computationally intelligent systems in complex real world systems. These intelligent systems are supposed to hold adeptness similar to humans within a specific domain, adjust themselves, and realize to do better in changing environments. Neuro-fuzzy computing is the perfect example where neural networks recognize patterns and adapt themselves to deal with the changing situations, fuzzy inference systems incorporate human knowledge and perform inferencing and decision making [42]. Adaptive networks can be used in a wide variety of applications of modeling, decision making, signal processing, and control.

In this section, we describe the architecture and hybrid learning rule used for ANFIS based on Sugeno fuzzy model. There are other Fuzzy models, such as Tsukamoto and Mamdani, but due to their complexity and with no increase in computational accuracy, Sugeno Fuzzy model is widely used in ANFIS. For the purpose of explaining the structure of ANFIS, we assume that the fuzzy inference system under consideration has two inputs x and y and one output z. For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is given as under:

$$\left.\begin{array}{l} \text{Rule } 1{:}\, If\ x\ is\ A_1\ and\ y\ is\ B_1,\ then\ f_1 = p_1x + q_1y + r_1 \\ \text{Rule } 2{:}\, If\ x\ is\ A_2\ and\ y\ is\ B_2,\ then\ f_2 = p_2x + q_2y + r_2 \end{array}\right\} \tag{2.9}$$

ANFIS is a 5 layer feed forward network in which layer 1 and layer 4 are adaptive nodes. The reasoning mechanism for ANFIS model for the rules is given in Eq. 2.2 has been illustrated in Figure 2.3(a) and the corresponding equivalent ANFIS architecture is shown in Figure 2.3(b), where nodes of the same layer have similar functions, output of any node is denoted by O_k^i, it represents output of i^{th} node in k^{th} layer as described follows:

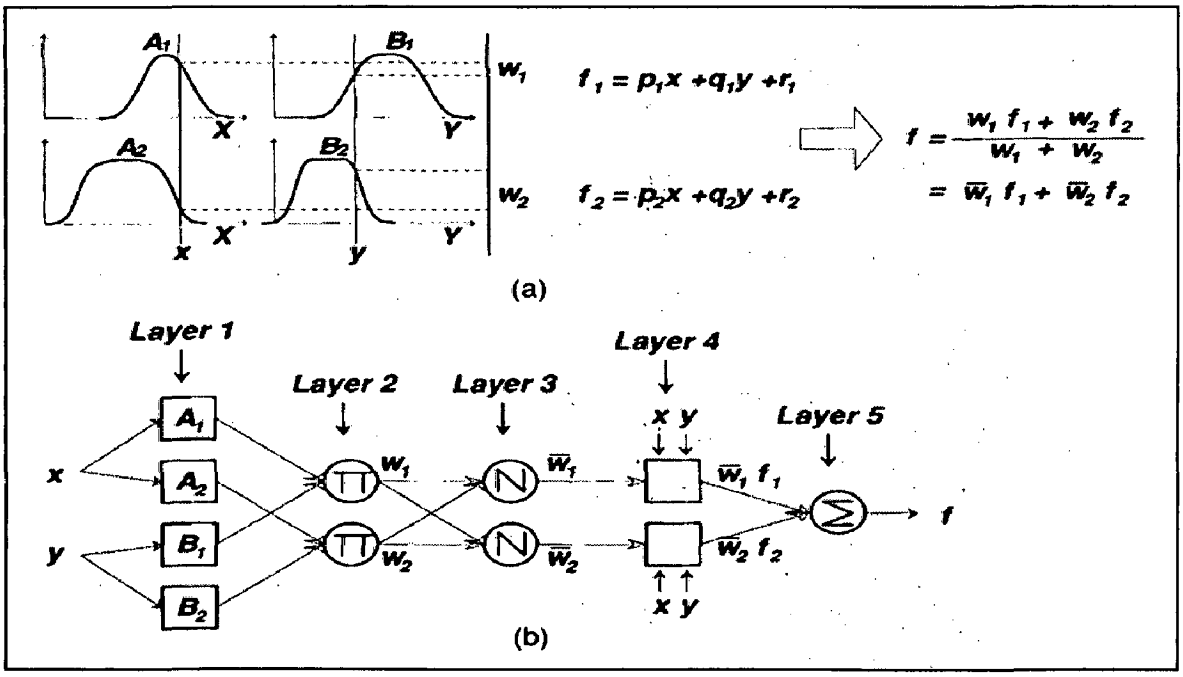

Figure 2.5 (a) A two input first order Sugeno fuzzy model with two rules and (b) Equivalent ANFIS architecture

Layer 1: Each node in this layer is an adaptive node with a node function. Output of the i^{th} node in layer 1 denoted as O_1^i where i represent to the node number in 1^{st} layer is:

$$O_1^i = \mu_{A_i}(x),\ for\ i = 1,2\ and\ O_1^i = \mu_{B_{i-2}}(y),\ for\ i = 3,4 \qquad (2.10)$$

Where, x and y are the input to node i and A_i and B_{i-2} are linguistic label (such as "*small*" or "*large*") associated with this node and $\mu_{A_i}(x), \mu_{B_{i-2}}(y)$ can adopt any appropriate parameterized membership function generally used Bell function with minimum equal to 0 and maximum equal to 1 shown in Eq. 2.4.

$$\mu_A(x) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b}} \qquad (2.11)$$

where, a_i, b_i and c_i are the parameters of the membership function. Parameters are referred to as premise parameters.

Layer 2: This layer also known as rule layer. Every node in this layer is a fixed node labeled Π, whose output is the product of all the incoming signals presented as:

$$O_2^i = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), i = 1,2 \qquad (2.12)$$

Each node output represents the firing strength of a rule. In general, any other T-norm operators that perform fuzzy AND can be used as the node function in this layer.

Layer 3: This layer also known as normalization layer. Every node in this layer is a fixed node labeled N. The i^{th} node calculates the ratio of the i^{th} rules' firing strength to the sum of all rules' firing strengths. Output of this layer is also called normalize firing strength:

$$O_3^i = \overline{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \qquad (2.13)$$

Layer 4: This layer is also known as defuzzification layer. Every node in this layer is an adaptive node with a node function. Output of this layer is simply product of firing strength and first order polynomial. As shown in Eq. 2.7.

$$O_4^i = \overline{w}_i \times f_i = \overline{w}_i(p_i x + q_i y + r_i) \qquad (2.14)$$

where, $\overline{w}_i$ is a normalized firing strength from layer 3 and $\{p_i,\ q_i,\ r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node labeled Σ, which computes the overall output as the summation of all incoming signals:

$$overall\ output\ =\ O_5^i\ =\ \sum_{i=1} \overline{w}_i \times f_i = \frac{\sum_{i=1} w_i f_i}{\sum_{i=1} w_i} \tag{2.15}$$

In the above mentioned architecture, Layers 2 and 3 can also be combined to perform the functions of calculating the firing strengths and then normalizing it. Though the architecture has been described for two inputs with two membership functions each and two rules only, but it can be extended for more number of inputs and rules.

It can be observe from the ANFIS architecture shown in Figure 2.3(b), that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. The output f can be also written as:

$$f = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2$$

$$= \overline{w}_1(p_1x + q_1y + r_1) + \overline{w}_2(p_2x + q_2y + r_2)$$

$$= (\overline{w}_1x)p_1 + (\overline{w}_1y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2x)p_2 + (\overline{w}_2y)q_2 + (\overline{w}_2)r_2 \tag{2.16}$$

which is linear in the consequent parameters p_1, q_1, r_1, p_2, q_2, and r_2. From this observation, we have

S = set of total parameters,

S_1 = set of premise (nonlinear) parameters,

S_2 = set of consequent (linear) parameters

Table 2.1 Hybrid learning procedure for ANFIS in forward pass and backward pass

	Forward Pass	**Backward Pass**
Premise Parameters	Fixed	Gradient descent
Consequent Parameters	Least square estimator	Fixed
Signals	Node outputs	Error signals

The learning formulas for the premise and consequent parameters are separate in the hybrid learning rule, as shown in Table 2.1. In the forward pass, learning of consequent parameters is carried out using Least Squares Estimator method. In the backward pass, learning of premise parameters is carried out by Gradient descent or Backpropagation method. Further speedup of learning is possible by using variants of the gradient method or other optimization techniques.

2.4 Measures of Performance Evaluation

In this study, we have used following performance measures [43] which are shown as follows:

(i) *Overall Accuracy* (ACC): Accuracy is the percentage ratio of correctly classified observation and total observation

$$Accuracy = \frac{Correctly\ classified\ data}{Total\ data} \tag{2.17}$$

(ii) *Sensitivity* (S_n) (TP-Rate): Sensitivity is the probability of correct prediction of positive examples.

$$Sensitivity = \frac{TP}{TP+FN} \tag{2.18}$$

(iii) *FP-Rate*: FP-Rate is the probability of false prediction of negative examples.

$$FP\text{-}Rate = \frac{FP}{FP+TN} \tag{2.19}$$

(iv) *Precision* (PPV) (TN-Rate): Precision also known as Positive Predictive Rate is the probability of correct prediction of total prediction of positive examples.

$$Precision = \frac{TP}{TP+FP} \tag{2.20}$$

(v) *F-Measure*: It is the ratio of 2 times correctly classified and summation of actual presented class and predicted class

$$F\text{-}Measure = \frac{2TP}{(P+P')} \tag{2.21}$$

where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, *FN* = False Negatives, *P* = Actual Classes, P' is predicted classes.

2.5 Research Gaps

- The facial expression classification approach should not be complex and must be scalable for a large number of people and emotions. Its accuracy cannot be restricted to a certain individual or small group of people and emotions.
- Approach should not be affected from the illumination condition of pose. Such as one image is in some bad light condition and some is in good. So classifier should not be affected from this problem. Many approaches have this inherent problem and can be able to classify some limited facial expression like anger, sad, happy, surprise.
- In a real environment, it is impractical for a system to be trained from images of a single individual, so the collection of images used for training of the feature detectors must include many different face shapes, sizes, and ethnicities. The system should be more people independent.

Chapter 3

Proposed Method for Facial Expression Classification

We consider the related issues associated with the emotion recognition i.e. facial expression classification in facial images. We propose a technique for facial expression classification, which extract efficient and reduced features from the images based on texture analysis and differential intensity of interesting regions together. The Rotational Invariant Uniform Local Binary Pattern (RIULBP) [20] texture feature are extracted from the images after dividing image into two blocks - eye block and mouth block. These blocks are identified after the normalization of image. The proposed method can be seen into three major steps. These steps in the proposed method, for facial expression classification are as shown in Figure 3.1. Figure 3.2 shows whole structure of the facial expression classification model. Steps in methodology are discussed in detail, describing their functionality and their need in the framework in the subsequent section.

3.1 Image Segmentation and Normalization

The first step in facial expression classification is identification of interesting regions in images which contributes major part in terms to good objective features. We want to compute efficient features to classify the emotions. To determine the emotion present in the image, all images need some preprocessing work. So image should be of same size and the object present in the image such as eyes and mouth should be aligned in the facial image. In the feature extraction we consider only the interested region eye region and mouth region. As the human observe the emotion of person through the eyes and mouth region same thing applied here. All images are cropped

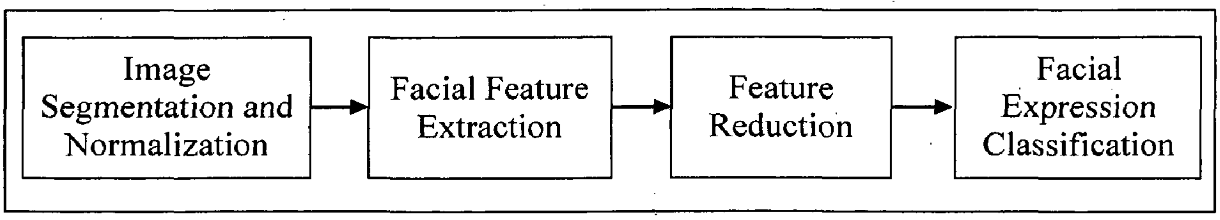

Figure 3.1 Major steps in Facial Expression Classification

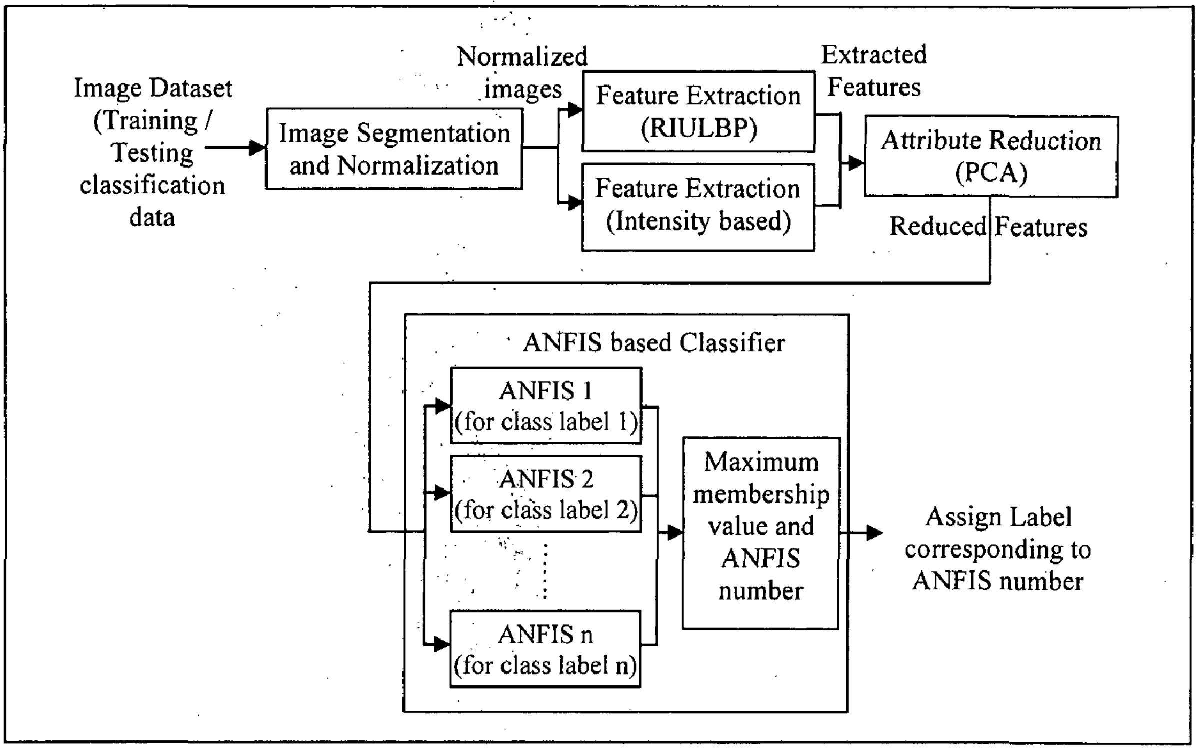

Figure 3.3 Proposed framework of Facial Expression Classification

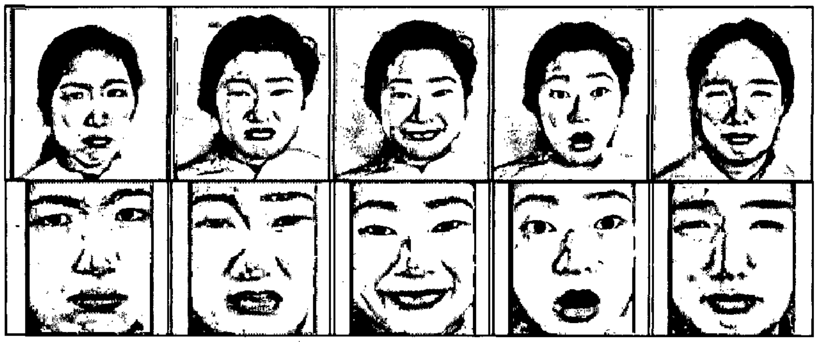

Figure 3.2 Cropped images from JAFFE dataset

into size 130 × 110 pixel sizes such as each image has only facial region not even hair portion also. Background present in the image is also not interesting part, so it has been automatically removed from image because we only cropped the facial part. To normalize the images, images are cropped such as eyes and mouth has same position in the images. Some images which are cropped shown in Figure 3.3.

3.2 Feature Extraction

Effective and efficient feature leads to higher accuracy in classification. Whole process of feature extraction as follows.

3.2.1 Division of Face into Sub Blocks

To extract the feature from the normalized facial images (as discussed in section 3.1) we have divided it into two blocks. First block is eye block which cover the eye region. Eye region covers eyes including the eyebrow. Second block is mouth block which cover the whole mouth. Some surrounded area of eyes and mouth also taken in block, so that the part of face affected by muscles movement in various emotions to be cover. Movements of face muscles in different emotions change the look and shape of face. The eye block and mouth block are shown in the Figure 3.4 and 3.5 respectively. Size of eye block is $M \times N$ and mouth block is $P \times Q$. Eye block is further divided into four sub blocks each block of size $M / 4 \times N$. And mouth block also divided further into four of size $P/2 \times Q/2$ as shown in the figures. Sub blocks can be increase as per requirement.

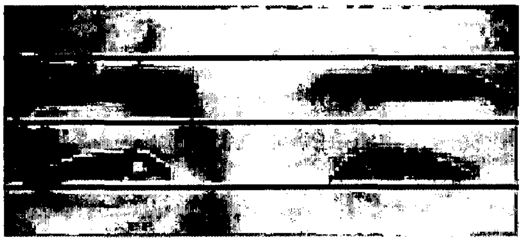

Figure 3.4 Eye block and its 4 sub blocks

Figure 3.5 Mouth block and its 4 sub blocks

3.2.2 Rotational Invariant Local Binary Pattern (RIULBP)

We have used Rotational Invariant Local Binary Pattern (RUILBP) [20] (discussed in Section 2.1.4) operator after dividing the facial images into two blocks. These blocks are eyes block and mouth. RIULBP gives 9 bins histogram for each block. So we have total of 18 attributes at the level. As discussed in Section 2.1.4, size of radius and neighborhood pixels can be increase and decrease corresponding to the implementation and accuracy. Selection of radius may depend on the resolution of images available.

3.2.3 Intensity Based Feature Extraction

As discussed in section 3.2.1, average intensity calculated for each particular sub blocks. To represent the feature to this average intensity of regions we calculate the difference of average intensity value of in eye sub blocks respectively. So we get three attributes from eye regions. Reason behind is that in any expression, portion of face moves between different regions and the average intensity value change. But this is not the case in case of mouth so only average intensity of mouth regions taken as such and we represent them as intensity based features attributes.

3.2.4 Feature Attributes Selection

As discussed in previous section we have calculated the feature vector using discussed techniques. RIULBP is calculated from whole eye block and mouth block. So we have 9, 9 attributes from the eye regions and mouth regions respectively and N attributes from differential intensity of sub blocks. If we combine these attributes then we get very large dimensional data that cannot be applied on proposed classifier used for classification. So two reduce the number of dimension PCA [44] (Section 2.2) is used it is applied on intensity based attributes as well as on RIULBP separately and combined later with again performing PCA. So after all we get total 4 strong attributes which represent to the facial feature we have extracted.

3.3 ANFIS Based Classifier

In this dissertation work, we propose to use adaptive neuro fuzzy inference system (ANFIS), as it is a hybrid technique with strengths of both Neural Networks and

Fuzzy Logic. In this technique Neural Network provides learning and adaptation from a given set of data and Fuzzy Logic provides the human like knowledge representation through fuzzy if then rules. We have N emotion which represents as N classes in dataset. Due to multiple classes we have used multiple ANFIS, one ANFIS for each class to classify the facial expression. Each ANFIS used membership value 1.0 for corresponding class label and 0.0 for others.

In ANFIS structure, numbers of rules are depends on the number of input membership functions corresponding to each input. As in our cases we have 4 attributes and let us choose 3 member functions for each input so total number of rules are $3^4 = 81$ rules. That is the main reason so that we have reduced the number of feature attributes so much. If we take more attributes system goes slow down. Even in case of more than 6 attributes ANFIS toolbox may go out of memory. There are many membership functions available for the inputs example triangular membership function (*trimf*), Bell membership function (*gbellmf*) etc. The function which given minimum root mean square error overall can been chosen for the training and testing purpose.

There are three kinds of fuzzy models which are generally used in Fuzzy inference systems, i.e. Sugeno, Tsukamoto and Mamdani fuzzy models. Mamdani fuzzy model is not used in ANFIS because of complexity in terms of structure and computation is higher than Sugeno and Tsukamoto fuzzy models. Mamdani fuzzy model also does not necessarily ensure better learning capability or approximation power. So either of Sugeno and Tsukamoto Fuzzy models based ANFIS can be used. Sugeno fuzzy base model been suggested to use because of its greater transparency and efficiency.

3.4 Facial Expression Determination

As discussed in the previous section MANFIS generate N membership values corresponding to each ANFIS designed for each emotion to be used for classification. To decide the class label for the particular image we find out maximum member ship value from these ANFIS. That emotion has highest membership value; image is

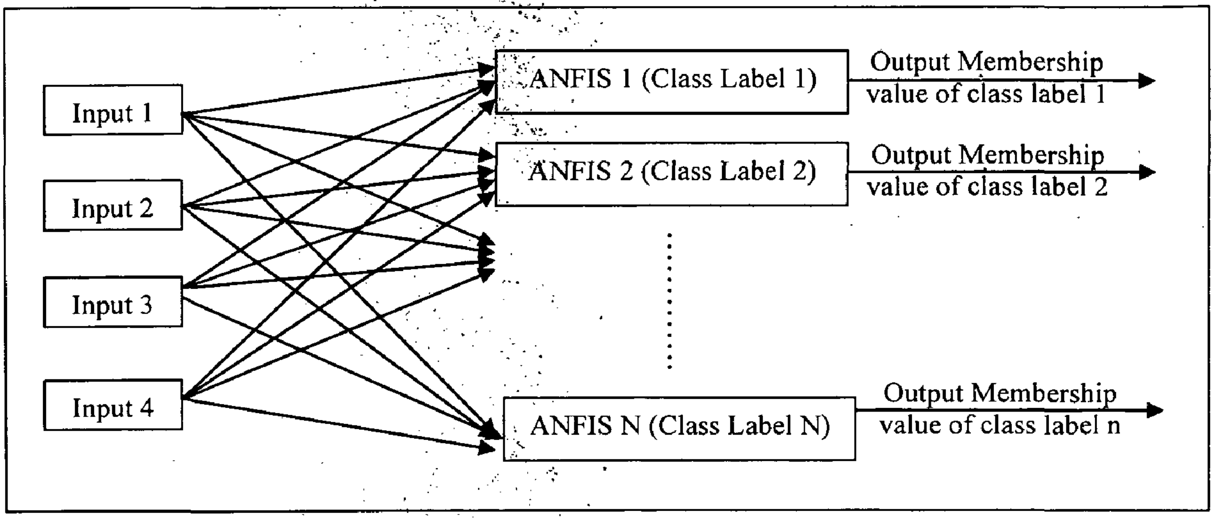

Figure 3.6 Structure of ANFIS based clssifier

classified into that category and assigns the class label. For example we apply image feature vector on MANFIS model then we get N membership values in which we found that output membership value of ANFIS k is highest in comparison of other $N-1$ ANFIS. Thus the image is classified corresponding to k^{th} ANFIS class label. Structure of classifier is shown in Figure

3.5 Flow Diagram of the Facial Expression Classification

The flow diagram of complete approach is broken in two phases as shown in Figure 3.10 and Figure 3.11. The first phase is training phase; where model learns and generate *FIS* format file. This *FIS* file stored into the disk and the second phase is classification phase, where MANFIS model read the information about the rules, functions, parameter values and structure stored in training phase then predicts the emotion. These steps are applied after the preprocessing step.

Training Steps:

- Read the gray value of pixels in images.
- Obtain Feature vector using RIULBP and Intensity based method for every image in training data.
- Reduce the attributes using PCA and choose only those attributes which has maximum rank store the eigenvalues matrix into file for later use.
- Apply the feature onto to the MANFIS classification model of n classes.

- Train the MANFIS and generate the *FIS* file which contains the rules, parameter values and structure information for each ANFIS.

Classification Steps:

1. Read the gray value of pixels in images.
2. Obtain Feature vector using RIULBP and Intensity based method for the image to classify.
3. Reduce the attributes by multiplying the feature matrix with eigen values matrix stored at the time of training.
4. Apply the feature onto to the MANFIS classification model of n classes
5. Find the maximum membership output obtained from n ANFIS.
6. Image is assigned the emotion class label which has highest output membership value.

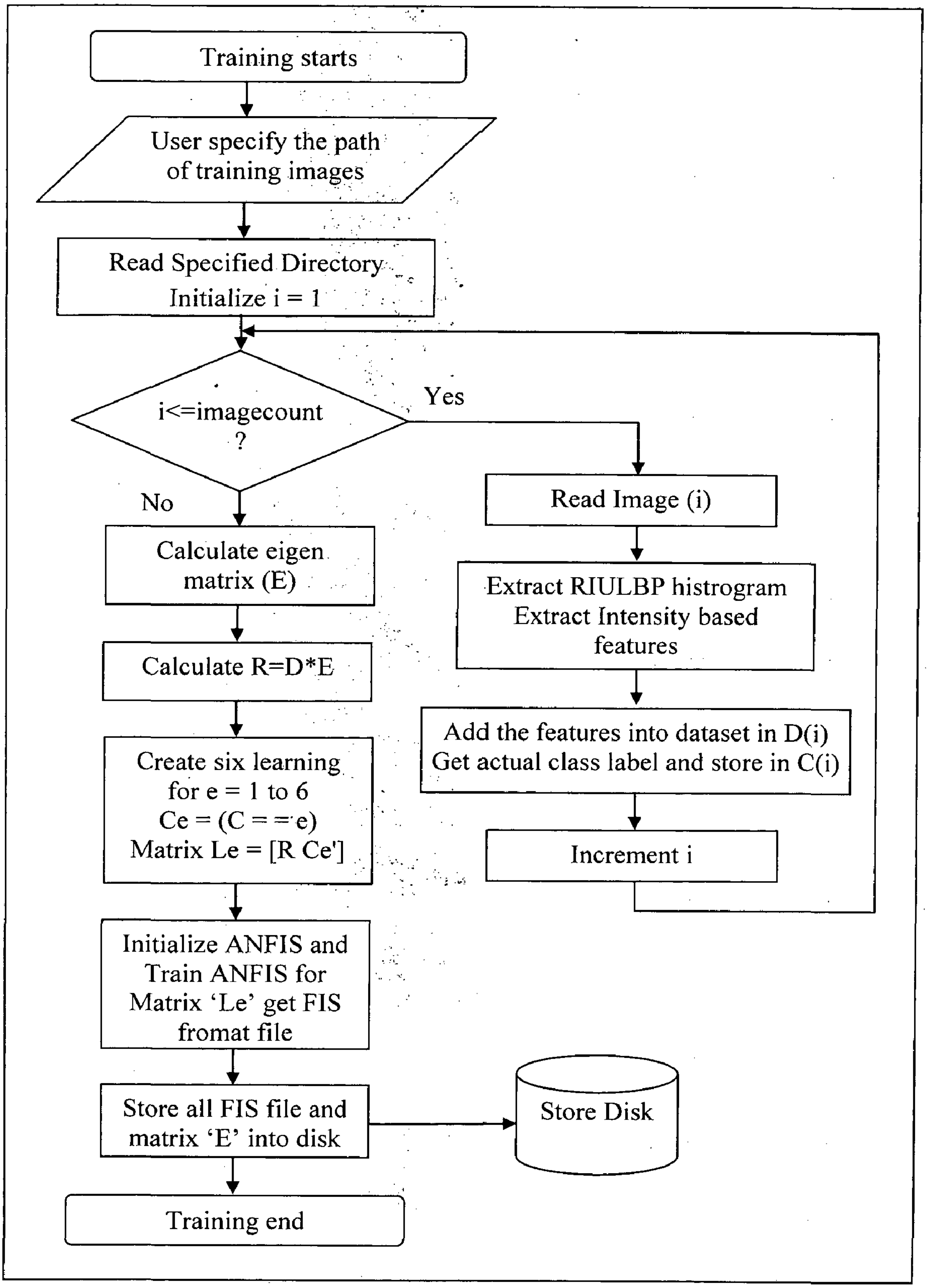

Figure 3.7 Flow diagram of the traing phase in LBP based feature extraction and classification of facial expression

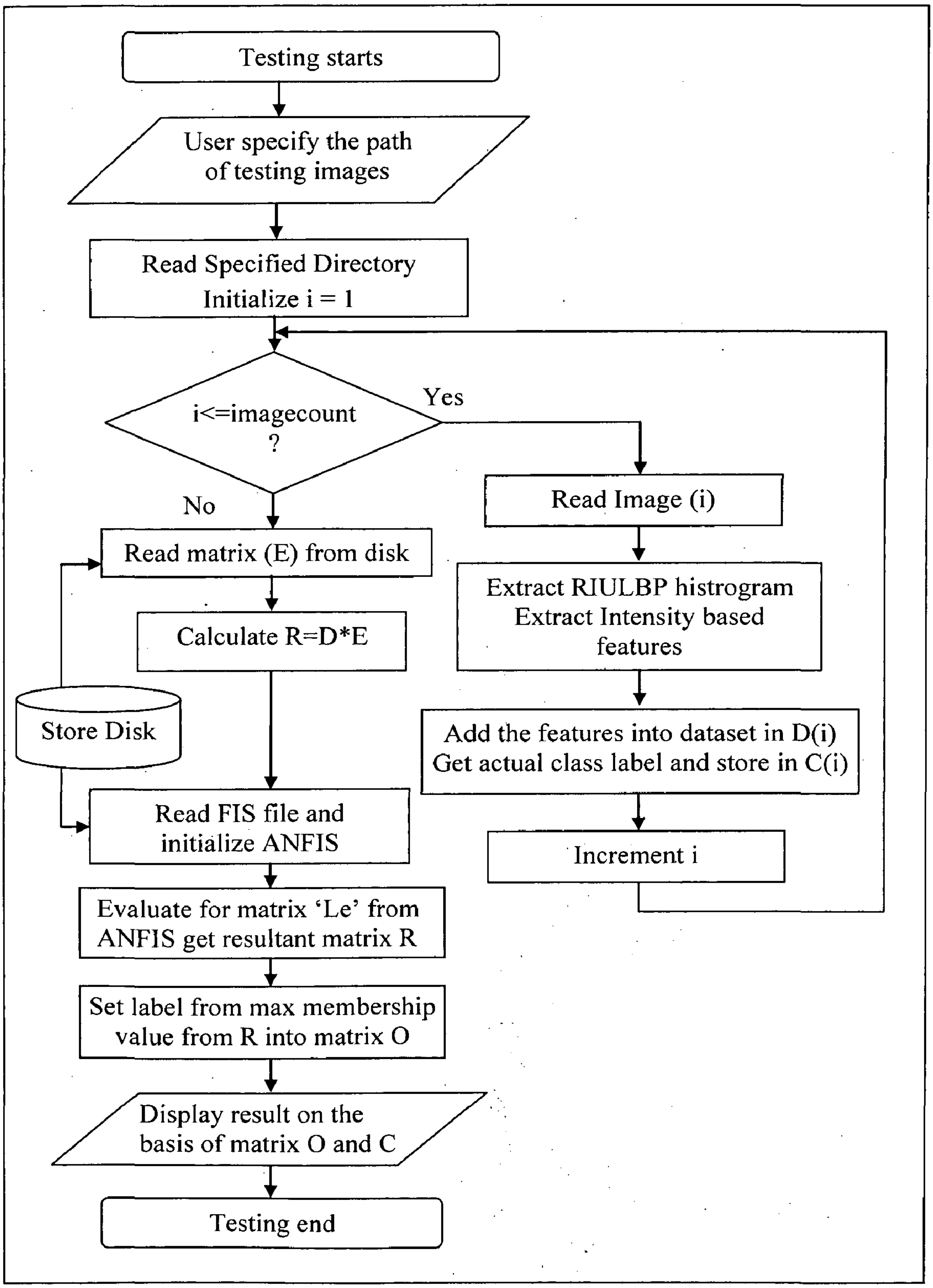

Figure 3.8 Flow diagram of the testing phase in LBP based feature extraction and classification of facial expression

Chapter 4

Implementation of the Proposed Framework

The project has been implemented in Java programming language using the NetBeans 7.0 Integrated Development Environment (IDE) and in MATLAB 2011a. The NetBeans IDE is written in Java and runs on every operating system platform where a JVM (Java Virtual Machine) is installed, including Windows, Mac OS, Linux, and Solaris. A JDK (Java Development Kit) is required for Java development functionality. For this project JAVA version 6 is used. We installed Java development kit JDK 1.6.0 freely available [45]. The NetBeans IDE allows fast development of application by auto generating some code and providing hints about the methods and data members. The IDE is freely available from internet [46].

This application has been implemented on Windows XP Operating System. Java only used to design graphical interactive user interface and for final determination of result. MATLAB has been used to develop the neuro fuzzy analyzer and RIULBP which is the main component of the application. MATLAB version 7.12.0.635 (R2011a) has been installed. Fuzzy Logic Toolbox has been used for finalizing the structure of ANFIS and then developing executable files of the MATLAB program, for use with the java program [47]. Using *anfisedit* command in 'Command Window' of MATLAB GUI, ANFIS editor can be reached where experimentation with various options available can be carried out and best one with least error can be selected.

The whole application consists of eight important modules. Functions and Classes that are used to implement these modules are described in the succeeding subsections. Basic Modules in the application are:

- Interactive Graphical User Interface
- Rotational Invariant Uniform Feature Extraction
- Intensity based Feature Extraction
- Feature Reduction
- Training of MANFIS.
- Evaluate Result from MANFIS

4.1 Modules in Facial Expression Classification Application

The modules in rotational invariant LBP feature extraction and classification using multiple adaptive neuro fuzzy inference systems (MANFIS) based application are:

4.1.1 Emotion_RecognitionApp Class

It is the class which has *public static void main()* method in java program which launches the application. At the startup, it is the first thread created by user. It initializes the application and passes the control to a new instance of *Emotion_ReognitionView* class.

4.1.2 Emotion_ReognitionView Class

It is the class in java program which provides Graphical User Interface to the user. As the program start from the main file program creates new instance of this class. This class has GUI components like jFrame, jButton etc. which initialize and a GUI appears in from of user. Now user is free to give command to perform test on images like training of MANFIS and testing the images, which category it will initialize. To train MANFIS model after specifying the path of training images as soon as user click on *train MANFIS Model* button, a executable code generated by MATLAB that is *Train_MANFIS* for training the MANFIS model execute. That code generates the *FIS* format files for each emotion in *FIS* folder and training complete. There is again an option from which user specify either single file or full folder and click on *Test* button then *EVAL_MANFIS* executable file executes and write a *result.txt* file after reading the structure of every ANFIS written in *FIS* format file. This *result.txt* file contains the membership values of all emotions calculated by MANFIS. To notify which folder has training files or which file has to be test or folder has to be test is the path is written into a file *DataPipe.dat.* Executable file simply read the file and start the execution after taking data from specified input path. Interactive Graphical User Interface for facial expression classification is shown in the Figure 4.1.

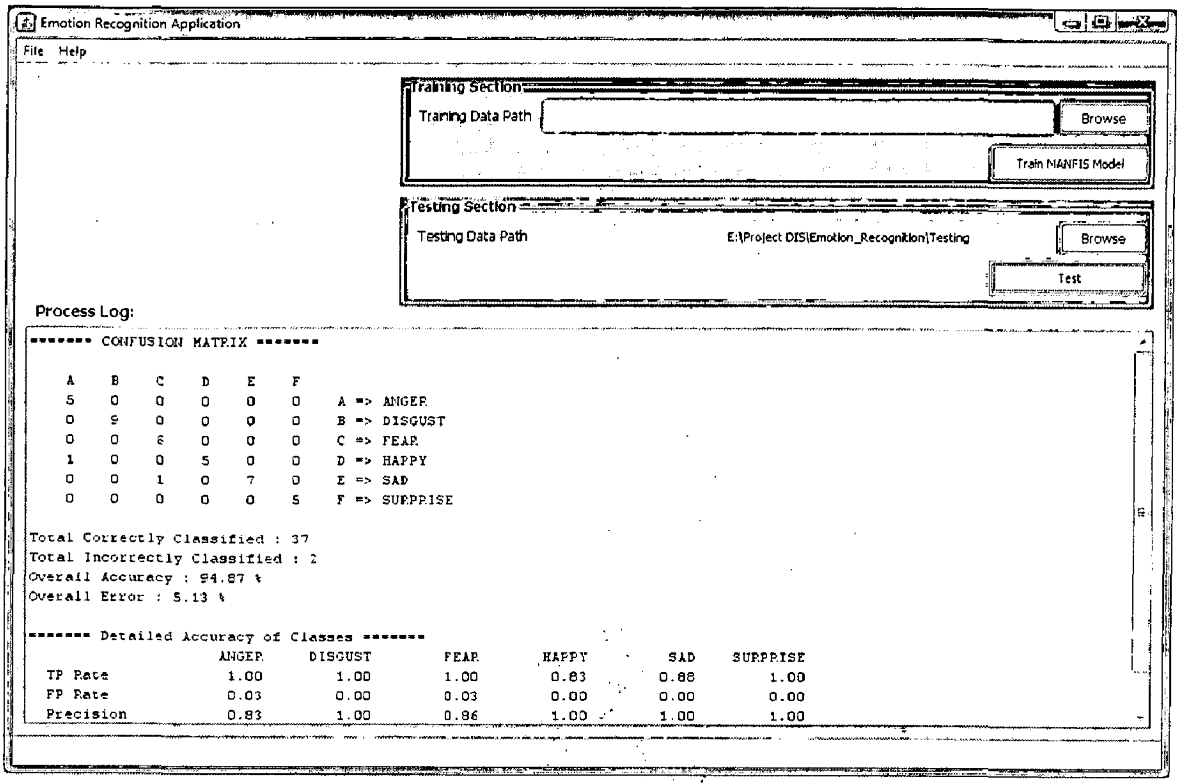

Figure 4.1 Interactive graphical user interface for facial expression classification

4.1.3 RIULBP82 Function

As discussed in section 3.2.2 rotational invariant uniform local binary pattern (RIULBP) is extracted to get effective feature vector. It is also discussed that RIULBP can be extracted taking any number of neighborhood pixels and radius. So this function as the name is used to extract the RIULBP histogram at number of neighborhood pixel $P=8$ and radius $R=2$. As discussed in starting of the section whole core code is implemented in MATLAB. *RIULBP82* function is written in *RIULBP82.m* file. This function takes $M \times N$ array as the input and the window size in which we have to calculate RIULBP and return the array of 9 values (bins). These 9 values are the frequency of each type of rotation invariant binary pattern or we can say this function return the histogram of RIULBP. To calculate the RIULBP from $M \times N$ matrix first we copy the window into another 2-d array and then calculate the LBP for the array. This $M \times N$ array is the pixel array of gray values which is passed from the function which wants to evaluate the RIULBP for any image. That particular function which calls to *RIULBP* function should read the image by its own and pass the array and the interesting window where RIULBP has to be calculated. Reason behind to taking window is that why should we calculate the RIULBP where we have not interested and make fast computation. Algorithm for calculating the RIULBP is shown in Table 4.1. This algorithm is pseudo code against MATLAB program.

Table 4.1 Pseudo code of *RIULBP82* function which evaluate RIULBP on $P = 8$, $R = 2$

Algorithm – RIULBP82: Calculate the frequency of each type of RIULBP pattern

Input:

- IMG: 2-d Array having pixel values of image.
- lr, lc: Starting coordinates from which we have to calculate RIULBP.
- rows, cols: It is size of the window in which we are interesting to calculate RIULBP.

Output:

- H: Array of size 9 representing the bins of RIULBP.

Method:

```
1.  initialize P = 8, R = 2, image = IMG(lr : lr + rows – 1, lc : lc + cols - 1)
2.  for i = 1 to 8  [calculate surrounding points]
3.          spoints(i, 1)= -R sin((i - 1)2π/P)
4.          spoints(i, 1)= R cos((i - 1)2π/P)
5.  end
6.  initialize blocksizex = 5, blocksizey = 5 [LBP window for a pixel]
7.  initialize originpointx = 3, originpointy = 3, dx = imagesizex - blocksizex,
    dy = imagesizey - blocksizey
8.  for i = to P [for each neighbor pixel]
9.          x = originpointx + spoint(i, 1)
10.         y = originpointy + spoint(i, 2)
11.         fx = floor(x), fy = floor(y)
12.         rx = round(x), ry = round(y)
13.         cx = ceil(x), cy=ceil(y)
            [check whether interpolation is needed or not]
14.         if abs (x – rx) <1e-6 && abs (y – ry) <1e-6 ) [not needed]
15.             N = image(ry : ry + dy, rx : rx + dx)
16.         else
17.             ty = y - fy
18.             tx = x - fx
```

```
            [calculate weight]
                wl = (1 - tx) × (1 - ty)
                w2 = tx × (1 - ty)
                w3 = (1 - tx) × ty
                w4 = tx × ty
                N= wl × image(fy : fy + dy, fx : fx + dx) + w2 × image(fy : fy +
                dy, cx : cx + dx) + w3 × image(cy : cy + dy, fx : fx + dx) + w4 ×
                image(cy : cy + dy, cx : cx + dx)
            End
D = N>=image
[update the result matrix
v = 2^(i - 1)
result = result + v × D
end
for i = 1 to rows in result matrix
            for j = 1 to columns in result matrix
                        Index = patternmap(result(i, j) + 1)
                        H(index) = H(index) + 1
            end
end
return H(1:9)
```

In this pseudo code *patternmap* is a function implementation of Eq. 3.7 which returns the index number of bin in which pattern value has to be added. Because we have only 256 possible values in *result* array so to make the process fast we have use a *patternmap* as a single dimensional array of size 256. We stored the bin number corresponding to array script value instead of calculating each time after generating the bin number corresponding to the numbers 0 to 255. To make this array following java code has been written, which print binary equivalent and bin number of particular decimal number as shown in Table 4.2:

Table 4.2 Java Program to decide which LBP should go on particular bin

Program – BinHelp: prints number from 0 to 255, their binary equivalent and Bin number in which it has to store.

Code:

```
public static void main(String []args)
{
    int bittransition, binnumber;
    System.out.println("Number\tBinary Eqv.\tParticular Bin Number");
    for(int i=0;i<256;i++)
    {
        binnumber=bittransition=0;
        System.out.print(i+ "\t");
        for(int j=0;j<8;j++)
        {
            bittransition+= Math.abs(((i>>>j)&1) - ((i>>>((j+1)%8))&1));
            System.out.print((i>>>(7-j)) &1);
            binnumber+=(i>>>j) &1;
        }
        binnumber=bittransition>2?10:binnumber+1;
        System.out.println("\t"+ binnumber);
    }
}
```

4.1.4 RIULBP81 Function

This is a function implemented in MATLAB. It calculates RIULBP on It is similar to RIULBP82 but used to extract the RIULBP histogram at number of neighborhood pixel $P=8$ and radius $R=1$. In this function we have not interpolate the points just taken surrounding pixels, so no interpolation needed. Algorithm of RIULBP81 is same as discussed in Table 4.1; just change is that no code for interpolation and the coordinates of surrounding pixels are taken as in static 2-d array. Those surrounding points coordinates *[(-1, -1), (-1, 0), (-1, 1), (0, 1), (1, 1), (1, 0), (1,-1), (0, -1)]* are stored in a two-dimensional array named *spoints*. Algorithm of *RIULBP81* function as shown in Table 4.3.

Table 4.3 Pseudo code for RIULBP81 function which calculate RIULBP on P=8, R=1

Algorithm – RIULBP81: Calculate the frequency of each type of RIULBP pattern

Input:

- IMG: 2-d Array having pixel values of image.
- lr, lc: Starting coordinates from which we have to calculate RIULBP.
- rows, cols: It is size of the window in which we are interesting to calculate RIULBP.

Output:

- H: Array of size 9 representing the bins of RIULBP.

Method:

```
1.  initialize P = 8, R = 1, image = IMG(lr : lr + rows – 1, lc : lc + cols - 1)
2.  spoints=[-1, -1; -1, 0; -1, 1; 0, 1; 1, 1; 1, 0; 1,-1; 0, -1]
3.  initialize blocksizex = 3, blocksizey = 3 [LBP window for a pixel]
4.  initialize originpointx = 2, originpointy = 2, dx = imagesizex - blocksizex,
    dy = imagesizey - blocksizey
5.  for i = to P [for each neighbor pixel]
6.          rx = spoint(i, 1) + originpointx, ry = spoint(i, 2) + originpointy
7.          N = image(ry : ry + dy, rx : rx + dx)
8.          D = N>=image
    [update the result matrix
9.  result = result + 2^(i - 1) × D
10. end
11. for i = 1 to rows in result matrix
12.         for j = 1 to columns in result matrix
13.                 Index = patternmap(result(i, j) + 1)
14.                 H(index) = H(index) + 1
15.         end
16. end
17. return H(1:9)
```

4.1.5 Avg_FV Function

This function is implemented in MATLAB. It calculates the feature vector from sub blocks as discussed in section 3.2.3. In this function we calculate the sum of pixels window specified in particular block using function *sum()* presented in MATLAB and after divide the value obtain from this function by the number of pixels in the window so we get average intensity of each sub block. Now we created average intensity feature vector taking the difference of adjacent block of eyes and concatenated the average vector of mouth region. Later on PCA has been applied and all 7 vectors are reduced into 4 final intensity based feature vectors. Algorithm of this function is given as follows in Table 4.4:

Table 4.4 Pseudo code for calculating intensity based feature vector

Algorithm – Avg_FV: Calculate the intensity based feature

Input:

- IMG: 2-d Array having pixel values of image.

Output:

- R: Array of size 7 representing the intensity features

Method:

1. A(1) = sum(IMG(Eye sub block 1))/14*104
2. A(2) = sum(IMG(Eye sub block 2))/14*104
3. A(3) = sum(IMG(Eye sub block 3))/14*104
4. A(4) = sum(IMG(Eye sub block 4))/14*104
5. V(1:3) = A(2:4) - A(1:3)
6. V(4) = sum(IMG(Mouth sub block 1))/30*40
7. V(5) = sum(IMG(Mouth sub block 2))/30*40
8. V(6) = sum(IMG(Mouth sub block 3))/30*40
9. V(7) = sum(IMG(Mouth sub block 4))/30*40
10. return V

4.1.6 Final_FV Function

This function is also implemented in MATLAB. This function receives attributes evaluated using *Avg_FV* function and *RUILBP82* or *RUILBP81* as per requirement. After that this function select best 4 attributes from received attributes after applying the PCA if it is training data and otherwise it reads the eigen vector and just multiply with input features array. The final feature attributes are evaluated and return the array of 4 elements that contain final 4 attribute has to be used for classification. Algorithm of this function is given as follows in Table 4.5:

In this algorithm we have choose best 4 attributes after applying PCA on both vectors. To choose best for we have decided to choose different 4 attributes for on different radius. When $P=2$, we select 2 attributes as 1st attribute of *R1* and *R2*, 3rd as second attributes of *(R1 + R2) / 2* and 3rd attribute of *R2*.

Table 4.5 Algorithm for Final_FV function that reduce the extracted features

Algorithm – Final_FV: Calculate the intensity based feature

Input:

• av_fv: 2-d Array having intensity based features.

• lbp_fv: 2-d Array having RIULBP based features.

• option: string specifying training or testing and radius used by LBP separated by ";"

Output:

• R: Array of size 4 representing the best features

Method:

1. if Training
2. EV1 = PCA(lbp_fv)
3. EV2 = PCA(av_fv)
4. csvwrite(filename1, EV1), csvwrite(filename2, EV2)
5. else
6. EV1 = csvread(filename1). EV2 = csvread(filename2, EV2)
7. end
8. R1 = av_Fv × EV1, R2 = lbp_fv × EV2, R3= Selectbest4(R1,R2)
9. return R3

4.1.7 Train_MANFIS Function

This is a main function of our method contain neuro fuzzy classifier. First this function read the file *DataPipe.dat* in which we write the path where training images exists from the GUI. After that this function read the directory and read all bit map images stored in that folder. In next step, this function calls the above mention function *Final_FV* function and obtain feature array. After that it makes the matrix of size $M \times 4$ for store the feature vector. This matrix passed to the ANFIS function with concatenating the class vector according to the image and particular ANFIS. Then it trains the 6 ANFIS and save resultant output of each ANFIS into the disk in *FIS* format after training completes. The output contains the information about the rule, function, parameter etc. The algorithm of this function is shown in Table 4.6:

Table 4.6 Algorithm to train MANFIS from training images

Algorithm – Train_MANFIS: Generate FIS format file after training images

Method:

1. read DataPipe.dat file and get training data path in DirName
2. read Directory DirName for bmp files
3. for each file in Directory Dirname
4. IMG=read(file);
   [add new row of intensity feature into avg_feature]
5. avg_feature += Avg_FV(IMG)
   [add new row of RIULBP feature into lbp_feature]
6. lbp_feature += LBP_FV82(IMG)
   [extract class name from IMAGE add new row in ClassName]
7. ClassName += GetActualClass(IMG)
8. end
   [get reduced feature vector from intensity based and RIULBP feature]
9. FeatureVector = FinalFv(avg_feature, lbp_feature, 'training;2')
   [initialize ANFIS parameters]
10. numFs = 3
11. inmftype = 'gaussmf'
12. outmftype = 'linear'

13. epoch_n = 10
14. generate classinfo AN for Angry from ClassName [angry = 1, other = 0]
    [initialize ANFIS structure]
15. in_fis = genfis1([TV AN'], numMFs,InmfType, outmfType)
    [get ANFIS output after traing
16. out_fis = anfis([TV AN'],in_fis,epoch_n)
    [save ANFIS output into file]
17. writefis(out_fis,'./FIS/Angry.fis')
18. repeat step 11 to 14 for each emotion and save FIS format file in FIS folder
19. end

4.1.8 EVAL_MANFIS Function

EVAL_MANFIS is another important function of our implemented application. This function read the files stored in training face into disk which contains the information about the output of each ANFIS. This function loads these files and initializes 6 ANFIS (one ANFIS for each emotion) and evaluates the result after applying the feature obtained from the facial images. In the next step, obtained output after evaluation has written into the *result.csv* file. The algorithm of this function is shown as follows in Table 4.7:

Table 4.7 Algorithm to classify the images

Algorithm – EVAL_MANFIS: write output of each ANFIS into file result.csv
Method:

1. read DataPipe.dat file and get testing data path in DirName
2. read Directory DirName for bmp files
3. for each file in Directory Dirname
4. IMG=read(file);
   [add new row of intensity feature into avg_feature]
5. avg_feature += Avg_FV(IMG)
   [add new row of RIULBP feature into lbp_feature]
6. lbp_feature += LBP_FV82(IMG)

```
      end
      [get reduced feature vector from intensity based and RIULBP feature]
  7.  FeatureVector = FinalFv(avg_feature, lbp_feature, 'training;2')
      [read FIS file stored at the time of traing]
  8.  AN_Anfis=readfis('./FIS/Angry.fis')
      [evaluate membership value of Angry from input features vector]
  9.  AN=evalfis(Input,AN_Anfis)
  10. Repeat step 8 and 9 for each emotion
      [define class label which membership function has maximum value]
  11. Class_label = emotion(max membership value anfis)
      [write resultant output membership value of each anfis into result.csv]
  12. Csvwrite('result.csv', [Class_label])
  13. End
```

Chapter 5

Results and Discussion

The implementation of the local binary pattern based feature extraction and classification of facial expression application is implemented in java and MATLAB. Dataset, performance measures and different parameters associated with the framework has been discussed in the subsequent section.

5.1 The Experiment Dataset

Local Binary Pattern based Feature Extraction and classification of facial expression attempt to recognize and label to a principal set of emotion i.e. anger, fear, sadness, disgust, surprise, and happiness. These categories are primary expressions identified by Ekman and Friesen [2]. Some sample images from the dataset are shown as in Figure 5.1.

In order to train and test the classification system, we have used images from the Japanese Female Facial Expression (JAFFE) database [48], located at (http://www.kasrl.org/jafee.html). This dataset is freely available for academic use and research work. The dataset includes 213 grayscale images of size 256 × 256 pixels. The dataset includes facial images of ten different female models,

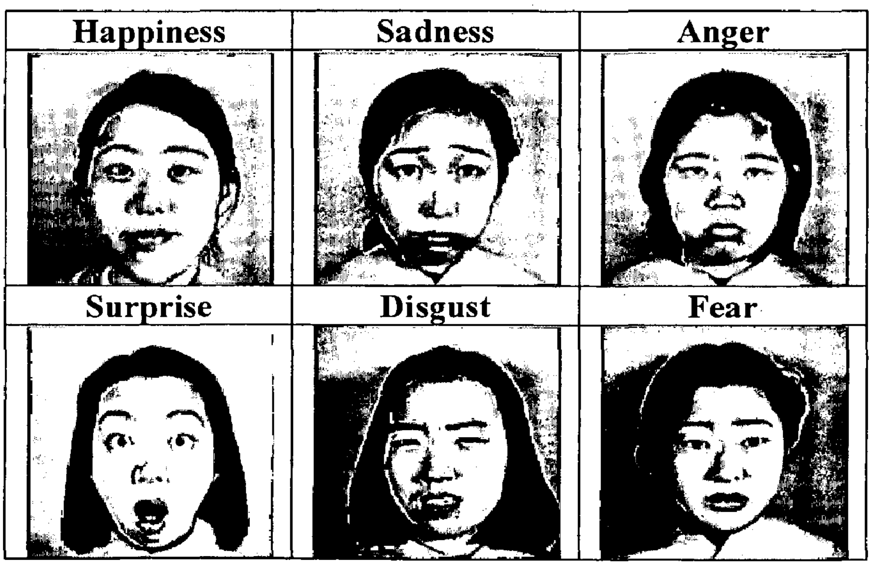

Figure 5.1 Sample images of six basic facial expressions from JAFFE data set

each assuming seven distinct posses (six basic expressions and one neutral pose). Data set has 2-4 images for each facial expression of each female. JAFFE dataset is the dataset which is used in many research works and used to validate and compare their results. In this dissertation work we have use only basic emotion expression images not neutral pose.

5.2 Results for JAFFE Dataset

The dataset has 213 facial images as discussed in previous section; we have used 188 images of basic expression (anger, disgust, fear, happy, sad, surprise) and left the images of neutral pose. We have taken 6 emotions so classifier with 6 ANFIS has been used. Each classifier used four inputs, three Bell membership functions per input. There are total of 193 nodes with 36 nonlinear parameters at input associated membership function layer and 405 linear parameters at output layer. There are 81 fuzzy rules corresponding to each facial expression (emotion). We have selected output function linear because it produces minimum error when experiment is done to select the membership function. The data set is small consisting 188 images which is less than the number of parameters 441 in the ANFIS structure so we get warning from MATLAB tool box when we setup experiment.

To quantify the performance of proposed method, data was partitioned into two parts the training data and the testing data. Because we have small data set so we made 5 sets of (39 + 38 + 37 + 36 + 38 = 188) images. These sets are designed such as any set cannot contain all the images of same pose in a single set of a person. After that we have apply testing 5 times such as 4 sets are used for training and remaining one set is used for testing. This is applied in scenario 1 and scenario 2. It is similar like 5 fold testing technique. Thus we have tested our method on whole 188 images after applying test on each set. After we have added all confusion matrices into one and calculate accuracy and various performance measures. In scenario 3 we have done same thing but differently we have separate the images by person. All images of are used to train the ANFIS except one person and test is applied on remaining images. In this study, we have calculated some performance measures – overall accuracy, sensitivity (true positive rate), false positive rate, precision (positive predictive value) and F1-score.

Implemented application tested with Japanese Female Facial Expression (JAFFE) dataset. As discussed in Section 3.2.1 eye block and mouth block are shown defined by size of eye block is 56 × 104 and mouth block is 60 × 80. Eye block is further divided into four sub blocks each block of size 14 × 104. And mouth block also divided further into four block of size 30 × 60. After describing the block we have extracted the features as discussed in Section 3.2.2 and 3.2.3. Results are tested 6 facial expression based ANFIS classifier on 3 different scenarios by changing the parameter radius of RIULBP. These scenarios are:

- **Scenario 1:** Classification of facial expression on taking 8 neighborhood pixels *(P=8)* and radius(R) = 1 in RIULBP feature extraction.
- **Scenario 2:** Classification of facial expression on taking 8 neighborhood pixels *(P=8)* and radius(R) = 2 in RIULBP feature extraction.
- **Scenario 3:** Classification of facial expression on taking 8 neighborhood pixels *(P=8)* and radius(R) = 2 in RIULBP feature extraction on untrained faces

Table 5.1 Notation symbol used for denoting facial expression

S. No.	Facial Expression	Notation Symbol
1.	Anger	A
2.	Disgust	B
3.	Fear	C
4.	Happy	D
5.	Sad	E
6.	Surprise	F

SCENARIO 1

In first scenario we have calculated rotational invariant local binary pattern on neighborhood pixel (P) = 8 and radius (R) = 1. Feature reduced and applied to the multiple adaptive neuro fuzzy classification model and result has been evaluated. To evaluate result all images are used such as after partitioning the all images in 5 sets one set is used for testing and rest 4 for training as discussed. Table 5.1 shows the confusion matrix after applying method on all 5 sets. Table 5.2, shows the detailed accuracy of classes.

Table 5.2 Confusion matrix between classes when on RIULBP (P = 8, R = 1)

	A	**B**	**C**	**D**	**E**	**F**
A	26	0	2	2	0	0
B	1	28	2	0	0	0
C	0	2	28	0	0	2
D	1	0	0	27	3	1
E	0	0	0	2	28	1
F	2	0	0	1	0	29

Total correctly classified : 166
Total Incorrectly Classified : 22
Overall Accuracy : 88.30 %
Overall Error : 11.70 %

Table 5.3 Detailed accuracy in classes and accuracy measures on RIULBP (P = 8, R = 2)

CLASS	TP Rate	FP Rate	Precision	F-measure
ANGER	0.87	0.03	0.87	0.87
DISGUST	0.9	0.01	0.93	0.92
FEAR	0.88	0.03	0.88	0.88
HAPPY	0.84	0.03	0.84	0.84
SAD	0.9	0.02	0.9	0.9
SURPRISE	0.91	0.03	0.88	0.89

SCENARIO 2

In this scenario results are evaluated same as in scenario 1 but we have made little change in feature extraction. Instead of taking radius 1 we have used radius =2 i.e. RIULBP feature extraction is applied on neighborhood pixel *(P)* = *8* and radius *(R)* = 2 pixels. Table 5.3 shows the confusion matrix and Table 5.4 shows the detailed accuracy of classes.

Table 5.4 Confusion matrix between classes on RIULBP (P = 8, R = 2)

	A	B	C	D	E	F
A	29	0	0	0	0	1
B	0	30	0	0	1	0
C	0	0	31	1	0	0
D	1	2	0	29	0	0
E	1	0	1	0	29	0
F	0	0	0	1	0	30

Total Correctly Classified : 180
Total Incorrectly Classified : 9
Overall Accuracy : 95.21 %
Overall Error : 4. 79 %

Table 5.5 Detailed accuracy in classes and accuracy measures on RIULBP (P = 8, R = 2)

CLASS	TP Rate	FP Rate	Precision	F-measure
ANGER	0.97	0.01	0.94	0.95
DISGUST	0.97	0.01	0.94	0.95
FEAR	0.97	0.01	0.97	0.97
HAPPY	0.91	0.01	0.94	0.92
SAD	0.94	0.01	0.97	0.95
SURPRISE	0.97	0.01	0.97	0.97

SCENARIO 3

In this scenario we have used RIULBP feature extraction on neighborhood pixels *(P) = 8* and radius *(R) = 2* because at this experimental setup we got higher accuracy in comparison of scenario 1 when radius (R) was one pixels. In this scenario we have made change such that not even single image of particular person is used for the training but used all for testing i.e. classification has applied on is on untrained face. Table 5.5 shows the confusion matrix in results and Table 5.6 shows detailed accuracy of classes in class.

Table 5.6 Confusion matrix between classes on RIULBP (P = 8, R = 2) and face was untrained

	A	B	C	D	E	F
A	20	1	0	0	6	3
B	4	21	0	0	3	3
C	3	0	22	4	1	2
D	1	0	2	21	3	5
E	3	1	1	1	24	1
F	0	0	2	0	5	25

Total Correctly Classified : 133
Total Incorrectly Classified : 55
Overall Accuracy : 70.74 %
Overall Error : 29.26 %

Table 5.7 Detailed accuracy in classes and some accuracy measures on untrained faces using RIULBP (P = 8, R = 2).

CLASS	TP Rate	FP Rate	Precision	F-Measure
ANGER	0.67	0.07	0.65	0.66
DISGUST	0.68	0.01	0.91	0.78
FEAR	0.69	0.03	0.81	0.75
HAPPY	0.66	0.03	0.81	0.72
SAD	0.77	0.11	0.57	0.66
SURPRISE	0.78	0.09	0.64	0.7

5.3 Comparison of Results

For the comparative analysis of the results has been done from the results of other approaches available in literature. The comparison with the approach and accuracy is given in the Table 5.8.

Table 5.8 Comparative analysis of results

Approach	Accuracy
Liao [9]	94.56%
Zu-li [22]	94.10%
Gomathi [23]	94.26%
Wencheng [24]	90.14%
Koutlas [25]	87.00%
Our Approach	95.21%

5.4 Discussion

The result obtained conclude that

- The results obtained using RIULBP ($P = 8, R= 2$) are better in comparison to RIULBP ($P = 8, R= 1$) since it depends on resolution of images.
- The result on untrained faces also given considerable accuracy.
- A comparison with some other techniques is done and RIULBP and intensity based methodology outperforms it considerably.
- The proposed methodology is robust and easy to implement so can be used for emotion recognition using facial expressions.

Chapter 6

Conclusion and Future Work

In the present work, we have extracted features from the facial expression images using proposed local binary patterns and intensity based method. The extracted features are reduced by using PCA. The reduced features are applied on to the ANFIS based classification system. The performance of the proposed method was evaluated using JAFFE dataset. The conclusions drawn from the present work can be summarized as follows:

6.1 Conclusion

The following conclusion can be drawn from present study:

- A generic framework has been proposed for the purpose of classifying facial expressions. The proposed method is based on adaptive neuro fuzzy inference system. It is easy to use and deploy.
- Feature extraction is performed only on the local areas of interest which may have higher significance in given context.
- Extracted features need to be reduced before applying on ANFIS based classifier.
- Classifier used Sugeno type fuzzy model based ANFIS so has less complexity in terms of computation and structure which yield fast classification.

6.2 Suggestions for the Future Work

Some directions for further research work are as follows:

- Existing methods has given limited performance on untrained face so need some work to increase the accuracy when faces are untrained.
- Method can be made as fully automatic, for facial expression classification using some face detection algorithm.

- A function can be derived for selection the radius $\hat{R}$ in RIULBP automatically which choose best value of R automatically from learning.
- ANFIS based classifier can use adaptive learning after building the classifier model time to time.

REFERENCES

[1] L. Shuai-Shi, T. Yan-Tao, and L. Dong, "New Research Advances of Facial Expression Recognition," in Proc. *8th International Conference on Machine Learning and Cybernetics*, Baoding, China, 2009, pp. 1150-1155.

[2] P. Ekman, W. V. Freisen, and S. Ancoli, "Facial Signs of Emotional Experience," *Journal of Personality and Social Psychology,* vol. 39, no. 6, p. 1125, 1980.

[3] S. Caifeng, G. Shaogang, and P. W. McOwan, "Robust Facial Expression Recognition using Local Binary Patterns," in Proc. *16th IEEE International Conference on Image Processing*, Genoa, Italy, 2005, pp. 370-373.

[4] P. Ekman and W. V. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* San Francisco, CA: Consulting Psychologists Press, 1978.

[5] H. Seyedarabi, W.-S. Lee, A. Aghagolzadeh, and S. Khanmohammadi, "Facial Expressions Recognition in a Single Static as well as Dynamic Facial Images Using Tracking and Probabilistic Neural Networks," in *Advances in Image and Video Technology*. vol. 4319, L.-W. Chang and W.-N. Lie, Eds.: Springer Berlin / Heidelberg, 2006, pp. 292-304.

[6] A. Metallinou, L. Sungbok, and S. Narayanan, "Audio-Visual Emotion Recognition Using Gaussian Mixture Models for Face and Voice," in Proc. *10th IEEE International Symposium on Multimedia*, Berkeley, CA, 2008, pp. 250-257.

[7] T. Yongqiu, L. Shaofa, and W. Minqin, "Intelligent Facial Expression Recognition System R&C-FER," in Proc. *7th World Congress on Intelligent Control and Automation*, Chongqing, China, 2008, pp. 2501-2506.

[8] I. Kotsia and I. Pitas, "Real Time Facial Expression Recognition from Image Sequences Using Support Vector Machines," in Proc. *IEEE International Conference on Image Processing*, Beijing, China, 2005, pp. II-966-969.

[9] T. Yan, L. Wenhui, and J. Qiang, "Facial Action Unit Recognition by Exploiting Their Dynamic and Semantic Relationships," *IEEE transactions on pattern analysis and machine intelligence,* vol. 29, no. 10, pp. 1683-1699, 2007.

[10] Y. I. Tian, T. Kanade, and J. F. Cohn, "Recognizing Action Units for Facial Expression Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 97-115, 2001.

[11] J. J. Wong and S. Y. Cho, "Facial Emotion Recognition by Adaptive Processing of Tree Structures," in Proc. *ACM Symposium on Applied Computing*, New York, USA, 2006, pp. 23-30.

[12] H. Kobayashi and F. Hara, "Recognition of Six Basic Facial Expression and Their Strength by Neural Network," in Proc. *IEEE International Workshop on Robot and Human Communication*, Tokyo , Japan 1992, pp. 381-386.

[13] H. Kobayashi and F. Hara, "Recognition of Mixed Facial Expressions by Neural Network," in Proc. *IEEE International Workshop on Robot and Human Communication*, Tokyo , Japan, 1992, pp. 387-391.

[14] M. Pantic and L. J. M. Rothkrantz, "Facial Action Recognition for Facial Expression Analysis from Static Face Images," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 3, pp. 1449-1461, 2004.

[15] J. Yu and B. Bhanu, "Evolutionary Feature Synthesis for Facial Expression Recognition," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1289-1298, 2006.

[16] L. Xiaomin and Z. Yujin, "Facial Expression Recognition Based on Gabor Histogram Feature and MVBoost," *Journal of Computer Research and Development*, vol. 7, pp. 1089-1096, 2007.

[17] W. Yubo, "Real Time Facial Expression Recognition with Adaboost," in Proc. *17th International Conference on Pattern Recognition*, Cambridge UK, 2004, pp. 926-929.

[18] B. Xiao-Bo, O. Jun, W. Qing-Xiu, and C. Yi-Zhuo, "An Application Study on Facial Expression Recognition Based on Gabor Wavelets," in Proc. *Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on*, 2009, pp. 1-4.

[19] T. Ojala, M. Pietikäinen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Featured Distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51-59, 1996.

[20] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[21] Y. Zi-Lu, Z. You-Wei, and L. Jing-Wen, "Manifold Learning Approach to Facial Expression Recognition on Local Binary Pattern Features," in Proc. *International Conference on Machine Learning and Cybernetics*, Baoding, 2009, pp. 405-410.

[22] H. Lianghua, Z. Cairong, Z. Li, and H. Die, "An Enhanced LBP Feature Based on Facial Expression Recognition," in Proc. *27th Annual International Conference of the Engineering in Medicine and Biology Society*, Shanghai, China, 2005, pp. 3300-3303.

[23] V. Gomathi, K. Ramar, and A. S. Jeevakumar, "Human Facial Expression Recognition using MANFIS Model," *International Journal of Computer Science and Engineering,* vol. 3, no. 2, pp. 93-97, 2009.

[24] W. Wencheng, C. Faliang, Z. Jianguo, and C. Zhenxue, "Automatic Facial Expression Recognition Using Local Pinary Pattern," in Proc. *8th World Congress on Intelligent Control and Automation (WCICA),* Jinan, China, 2010, pp. 6375-6378.

[25] A. Koutlas and D. I. Fotiadis, "An Automatic Region Based Methodology for Facial Expression Recognition," in Proc. *IEEE International Conference on Systems, Man and Cybernetics*, Singapore, 2008, pp. 662-666.

[26] Z. Guoying, "Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions," *IEEE transactions on pattern analysis and machine intelligence,* vol. 29, no. 6, pp. 915-928, 2007.

[27] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine Series 6,* vol. 2, no. 11, pp. 559-572, 1901.

[28] I. T. Jolliffe, *Principal Component Analysis*: Springer, Heidelberg, 2002.

[29] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Human Genetics,* vol. 7, no. 2, pp. 179-188, 1936.

[30] G. J. McLachlan, "Discriminant Analysis and Statistical Pattern Recognition," John Wiley & Sons, Inc., 2005, pp. 1-15.

[31] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-233, 2001.

[32] "Dimension Reduction," Wikipedia, the free encyclopedia, Available: http://en.wikipedia.org/wiki/Dimension_reduction, [20 May, 2011]

[33] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*: Morgan Kaufmann, 2006.

[34] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach* vol. 1: Morgan Kaufmann, 1985.

[35] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*: Morgan Kaufmann, 2005.

[36] G. H. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," in Proc. *International Conference on Uncertainty in Artificial Intelligence*, San Mateo, 1995, pp. 338–345.

[37] P. Langley and S. Sage, "Induction of Selective Bayesian Classifiers," in Proc. *10th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, 1994, pp. 399-406.

[38] V. N. Vapnik, *Statistical Learning Theory*: Wiley-Interscience, 1998.

[39] J. T. Wei, Z. Zhang, S. D. Barnhill, K. R. Madyastha, H. Zhang, and J. E. Oesterling, "Understanding Artificial Neural Networks and Exploring Their Potential Applications for the Practicing Urologist," *Urology,* vol. 52, no. 2, pp. 161-172, 1998.

[40] S. Chung-Kwan, Y. Ui Tak, K. Huy Kang, and P. Sang Chan, "A Hybrid Approach of Neural Network and Memory-based Learning to Data Mining," *IEEE Transactions on Neural Networks,* vol. 11, no. 3, pp. 637-646, 2000.

[41] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks,* vol. 2, no. 5, pp. 359-366, 1989.

[42] J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-Fuzzy and Soft Computing," Computer Science Department, Tsing Hua University, Taiwan, Available: http://www.books.google.co.id, [10 Jun, 2011]

[43] "Sensitivity and Specificity," Wikipedia, The Free Encyclopedia, Available: http://en.wikipedia.org/wiki/Sensitivity_and_specificity, [15 Jun, 2011]

[44] W. Wencheng, C. Faliang, Z. Jianguo, and C. Zhenxue, "Automatic Facial Expression Recognition Using Local Binary Pattern," in Proc. *8th World Congress on Intelligent Control and Automation (WCICA),* Jinan, China, 2010, pp. 6375-6378.

[45] "Java Development Kit (JDK 1.6)," Oracle Corporation, Available, [16 Jun, 2011]

[46] "NetBeans IDE 7.0," Oracle Corporation, Available: http://netbeans.org/, [16 Jun, 2011]

[47] "MATLAB - The Language Of Technical Computing," The MathWorks, Inc., Available: http://mathworks.com/products/matlab, [12 May, 2011]

[48] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding Facial Expressions with Gabor Wavelets," in Proc. *3rd IEEE International Conference on Automatic Face and Gesture Recognition,* Nara, Japan, 1998, pp. 200-205.

LIST OF PUBLICATIONS

[1] **Kehar Singh** and Durga Toshniwal, "Emotion Recognition Using Local Binary Pattern," in *National Seminar on Open Source Software Systems(OSSS2011)* Srinagar, Kashmir: Trends in Information Management (TRIM) 2011.