FACE RECOGNITION

SUBMITTED BY

BINAYA DILLIP SWAIN

ARJUN RAJENDRAN NAIR

RISHI RAJ RAJANI

UNDER THE GUIDANCE OF

MS. URVI KORE

ASSISTANT PROFESSOR

COMPUTER ENGINEERING DEPARTMENT

COMPUTER ENGINEERING DEPARTMENT

THADOMAL SHAHANI ENGINEERING COLLEGE

UNIVERSITY OF MUMBAI

2013-2014

FACE RECOGNITION

PROJECT REPORT

Submitted in partial fulfillment of the
Requirements for the degree of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

By

Seat No.	Group No. 36
_________	Binaya Swain
_________	Arjun Nair
_________	Rishi Rajani

Under the guidance of
Ms. Urvi Kore
(Assistant Professor, Computer Engineering Department)

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Computer Engineering Department
Thadomal Shahani Engineering College
University of Mumbai

2013-2014

Project entitled: **FACE RECOGNITION**

Submitted by: BINAYA SWAIN

ARJUN NAIR

RISHI RAJANI

in partial fulfillment of the degree of B.E. in Computer Engineering is approved.

Guide	Examiners
______________________	______________________
Ms. Urvi Kore Assistant Professor Computer Engineering Department	
	______________________
______________________	______________________
Head of the Department	Principal

Date: ________________

TABLE OF CONTENTS

Sr. No.	Topic	Pg. No.
	LIST OF FIGURES	**i**
1.	**INTRODUCTION**	**1**
	1.1 Abstract	1
	1.2 Introduction and Motivation	1
	1.3 Problem Statement	2
	1.4 Scope of the Project	3
	1.5 Organization of the Project Report	4
2.	**REVIEW OF LITERATURE**	**5**
	2.1 Facial Expression	5
	2.2 Methods of expression recognition	6
	2.2.2 PCA	6
	2.2.3 LBP	9
	2.3 Image Processing	12
	2.4 Template Matching	12
	2.5 Current Technology and Methodology	12
3.	**ANALYSIS AND DESIGN**	**14**
	3.1 Requirements Analysis	14
	3.2 Project Design	17
	3.3 GUI Design	31
4.	**IMPLEMENTATION AND RESULTS**	**32**
	4.1 Implementation Details	32
	4.2 Results	35

5.	**TESTING**	**39**
5.1	Unit Testing	39
5.2	System Testing	40
6.	**CONCLUSION AND FURTHER WORK**	**42**
6.1	Conclusion	42
6.2	Further Work	43
	REFERENCES	
	ACKNOWLEDGEMENT	
	SUMMARY	

LIST OF FIGURES

Figure No.	Figure Caption	Pg. No.
Figure 2.1	Set of input images.	7
Figure 2.2	An image displaying Eigen faces obtained from the input images.	8
Figure 2.3	An illustration of the basic LBP operator.	9
Figure 2.4	Different texture primitives detected by LBP.	10
Figure 2.5	58 different patterns in (8,R) neighbourhood.	11
Figure 3.1	Block diagram for training phase.	20
Figure 3.2	Block diagram for testing phase.	20
Figure 3.3	Input Images from Different Users.	21
a)	Input Set of Class 1	21
b)	Input Set of Class 2	22
c)	Input Set of Class 2	23
Figure 3.4	Cropping of the input image.	24
Figure 3.5	A face image divided into 6 x 7 sub-regions.	26
Figure 3.6	An illustration of the basic LBP operator.	26
Figure 3.7	LBP histograms.	27
Figure 3.8	Activity Diagram of Facial Expression Recognition System .	30
Figure 3.9	GUI design.	31
Figure 4.1	Input image.	35
Figure 4.2	Cropped image.	35
Figure 4.3	An image displaying extracted features.	36

Figure 4.4	An illustration of the generated LBP.	36
Figure 4.5	Histogram of the LBP features.	37
Figure 4.6	Output on the GUI showing detected expression.	38
Figure 4.7	Output on the command window showing detected expression.	38
Figure 5.1	Select training images for testing.	40
Figure 5.2	Output on the GUI when training image is used for testing.	40
Figure 5.3	Select unknown images for testing.	41
Figure 5.4	Output on the GUI when unknown image is used for testing.	41

1. INTRODUCTION

This chapter consists of the introduction to the project entitled ‘Face Recognition - Detection and Recognition of Facial Expressions’. It also contains the abstract, motivation, problem statement and scope of the project. The last part of the chapter gives the organization of the project report.

1.1 ABSTRACT

Automatic facial expression analysis is an interesting and challenging problem, and impacts important applications in many areas such as human–computer interaction and data-driven animation. Deriving an effective facial representation from original face images is a vital step for successful facial expression recognition. We empirically evaluate facial representation based on statistical local features, Local Binary Patterns, for person-independent facial expression recognition. Different machine learning methods are systematically examined on several databases. Extensive experiments illustrate that LBP features are effective and efficient for facial expression recognition. Thus we use LBP features and template matching to stably and robustly recognize a facial expression.

1.2 INTRODUCTION AND MOTIVATION

Facial expression is one of the most powerful, natural and immediate means for human beings to communicate their emotions and intensions. Automatic facial expression analysis is an interesting and challenging problem, and impacts important applications in many areas such as human–computer interaction and data-driven animation. Due to its wide range of applications, automatic facial expression recognition has attracted much attention in recent years. Though much progress has been made, recognizing facial expression with a high accuracy remains difficult due to the subtlety, complexity and variability of facial expressions.

There are numerous potential applications for recognizing facial expressions. Police can use expressions to detect abnormal behaviour. Doctors can detect suppressed emotions in patients to recognize when additional reassurance is needed. Teachers can recognize unease in students and give a more careful explanation. Business negotiators can use glimpses of happiness to determine when they have proposed a suitable price. Thus due to the various important applications, software for recognizing facial expressions would be very valuable.

One limitation of the existing facial expression recognition methods is that they attempt to recognize facial expressions from data collected in a highly controlled environment given high resolution frontal faces. However, in real-world applications such as smart meeting and visual surveillance, the input face images are often at low resolutions. Obviously low-resolution images in real world environments make real-life expression recognition much more difficult. Recently Tian et al. made a first attempt to recognize facial expressions at low resolutions. Tian studied the effects of different image resolutions for each step of automatic facial expression recognition.

In this project we use LBP (Local Binary Patterns) features for person-independent facial expression recognition. Template matching machine learning method is used to classify expressions in the database.

1.3 PROBLEM STATEMENT

In this project, a Facial Expression Recognition system has been implemented which is based on the images captured using a web-cam and the dynamic features of the face. Images from 10 different users for each expression have been captured (total of 30) for training the system. The images are processed and the LBP features that are extracted are used to create templates for each class of expression corresponding to the six basic emotions. Testing has been performed after the training is complete.

The project is more accurate, cost efficient, can be executed in real-time, and does not require very expensive and advanced equipments.

1.4 .SCOPE OF THE PROJECT

There are numerous potential applications for recognizing facial expressions. Police can use expressions to detect abnormal behaviour. Doctors can detect suppressed emotions in patients to recognize when additional reassurance is needed. Teachers can recognize unease in students and give a more careful explanation. Business negotiators can use glimpses of happiness to determine when they have proposed a suitable price.

In addition to being used for security systems, authorities have found a number of other applications for facial recognition systems. Police can use facial recognition software to search for potential criminals and terrorists in attendance at the event.

In the 2000 presidential election, the Mexican government employed facial recognition software to prevent voter fraud. Some individuals had been registering to vote under several different names, in an attempt to place multiple votes. By comparing new facial images to those already in the voter database, authorities were able to reduce duplicate registrations. Similar technologies are being used in the United States to prevent people from obtaining fake identification cards and driver’s licenses.

There are also a number of potential uses for facial recognition that are currently being developed. For example, the technology could be used as a security measure at ATMs. Instead of using a bank card or personal identification number, the ATM would capture an image of the customer's face, and compare it to the account holder's photo in the bank database to confirm the customer's identity.

Because of certain limitations of fingerprint recognition systems, facial recognition systems are used as an alternative way to confirm employee attendance at work for the claimed hours.

1.5 ORGANIZATION OF THE REPORT

This report provides a detailed analysis of each and every stage in the development of the project. The report is organized as follows:

1.5.1 Chapter 2 Review of Literature

This chapter consists of the explanation on the current technologies and methodologies used in this domain and the technologies and methodologies used by the facial expression recognition system.

1.5.2 Chapter 3 Analysis and Design

This chapter includes the detailed Analysis of the project which is further broken down into functional and non-functional requirements. It also consists of the proposed system which is implemented in this project. This chapter also focuses on the major design considerations and the design details which have been used in the project. Also the major implementation issues which had been encountered and their remedies have been specified.

1.5.3 Chapter 4 Implementation and Results

This chapter includes the details related to implementation and explains the results produced after the implementation of the project. This chapter also focuses on the major implementation issues which had been encountered during the implementation and their remedies.

1.5.4 Chapter 5 Testing

This chapter focuses on testing of the Facial Expression Recognition System. This chapter includes the details of the testing phase that followed the implementation. It includes various levels of testing such as unit testing and system testing.

1.5.5 Chapter 6 Conclusion and Further Work

This chapter majorly focuses on the conclusion and further work which can be carried out in this domain with a view to enhance this project.

2. REVIEW OF LITERATURE

This chapter consists of the explanation on the current technologies and methodologies used in this domain and the technologies and methodologies used by the facial expression recognition system.

2.1 FACIAL EXPRESSIONS

A facial expression is one or more motions or positions of the muscles beneath the skin of the face. These movements convey the emotional state of an individual to observers. Facial expressions are a form of nonverbal communication. They are a primary means of conveying social information between humans, but they also occur in most other mammals and some other animal species.

Humans can adopt a facial expression voluntarily or involuntarily, and the neural mechanisms responsible for controlling the expression differ in each case. Voluntary facial expressions are often socially conditioned and follow a cortical route in the brain. Conversely, involuntary facial expressions are believed to be innate and follow a sub cortical route in the brain.

Facial recognition is often an emotional experience for the brain and the amygdala is highly involved in the recognition process.

The eyes are often viewed as important features of facial expressions. Aspects such as blinking rate can be used to indicate whether or not a person is nervous or whether or not he or she is lying. Also, eye contact is considered an important aspect of interpersonal communication. However, there are cultural differences regarding the social propriety of maintaining eye contact or not.

Beyond the accessory nature of facial expressions in spoken communication between people, they play a significant role in communication with sign language . Many phrases in sign language include facial expressions in the display.

There is controversy surrounding the question of whether or not facial expressions are worldwide and universal displays among humans. Supporters of the Universality Hypothesis claim that many facial expressions are innate and have roots in evolutionary ancestors. Opponents of this view question the accuracy of the studies used to test this claim and instead believe that facial expressions are conditioned and that people view and understand facial expressions in large part from the social situations around them.

2.2 METHODS OF EXPRESSION RECOGNITION

2.2.1 PRINCIPAL COMPONENT ANALYSIS

It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data.

The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, i.e., by reducing the number of dimensions, without much loss of information.

The task of facial recognition is discriminating input signals (image data) into several classes (persons). The input signals are highly noisy (e.g. the noise is caused by differing lighting conditions, pose etc.), yet the input images are not completely random and in spite of their differences there are patterns which occur in any input signal. Such patterns, which can be observed in all signals, could be – in the domain of facial recognition – the presence of some objects (eyes, nose, mouth) in any face as well as relative distances between these objects. These characteristic features are called eigenfaces in the facial recognition domain (or principal components generally). They can be extracted out of original image data by means of a mathematical tool called Principal Component Analysis (PCA). The objective is to extract the relevant information in a face image, encode it as efficiently as possible, and compare one face encoding with a database of models encoded in the same way. In mathematical terms, the objective is to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images. These eigenvectors can be thought of as a set of features which together characterize the variation between face images. Each image 16 location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of ghostly face called an *eigenface*.

By means of PCA one can transform each original image of the training set into a corresponding eigenface. An important feature of PCA is that one can reconstruct any original image from the training set by combining the eigenfaces.

Eigenfaces are nothing less than characteristic features of the faces. Each face image in the training set can be represented exactly in terms of a linear combination of the eigenfaces. The number of possible eigenfaces is equal to the number of face images in the training set. The eigenfaces are essentially the basis vectors of the eigenface decomposition. Therefore one could say that the original face image can be reconstructed from eigenfaces if one adds up all the eigenfaces (features) in the right proportion. Each eigenface represents only certain features of the face, which may or may not be present in the original image.

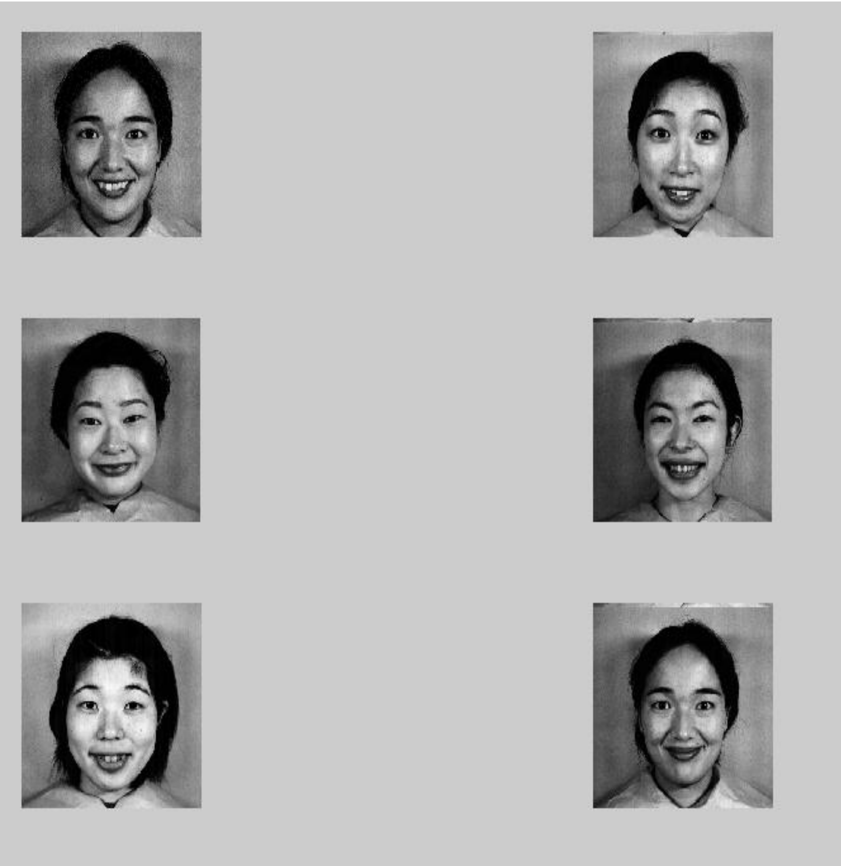

Figure 2.1 Set of input images

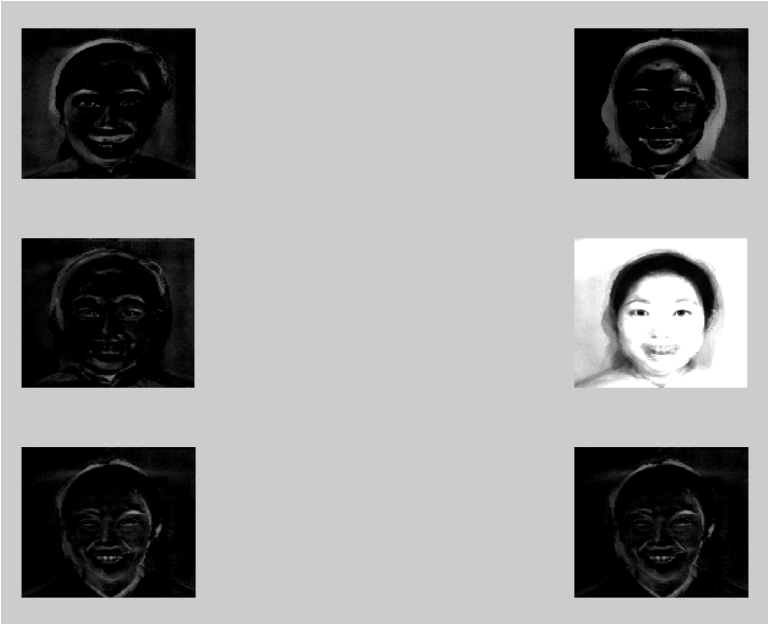

Figure 2.2 An image displaying the Eigen faces obtained from the input images

If the feature is present in the original image to a higher degree, the share of the corresponding eigenface in the “sum” of the eigenfaces should be greater. If, contrary, the particular feature is not (or almost not) present in the original image, then the corresponding eigenface should contribute a smaller (or not at all) part to the sum of eigenfaces. So, in order to reconstruct the original image from the eigenfaces, one has to build a kind of weighted sum of all eigenfaces. That is, the reconstructed original image is equal to a sum of all eigenfaces, with each eigenface having a certain weight. This weight specifies, to what degree the specific feature (eigenface) is present in the original image.

If one uses all the eigenfaces extracted from original images, one can reconstruct the original images from the eigenfaces exactly. But one can also use only a part of the eigenfaces. Then the reconstructed image is an approximation of the original image. However, one can ensure that losses due to omitting some of the eigenfaces can be minimized. This happens by choosing only the most important features (eigenfaces). using only the “best” eigenfaces—those that have the largest eigenvalues, and which therefore account for the most variance within the set of face images. The primary reason for using fewer eigenfaces is computational efficiency. The most meaningful M eigenfaces span an M-dimensional subspace—“face space”—of all possible images.

2.2.2 LOCAL BINARY PATTERN

The local binary pattern operator is an image operator which transforms an image into an array or image of integer labels describing small-scale appearance of the image. These labels or their statistics, most commonly the histogram, are then used for further image analysis. The most widely used versions of the operator are designed for monochrome still images but it has been extended also for color (multichannel) images as well as videos and volumetric data.

The basic local binary pattern operator, introduced by Ojala et al. [52], was based on the assumption that texture has locally two complementary aspects, a pattern and its strength. In that work, the LBP was proposed as a two-level version of the texture unit to describe the local textural patterns.

The original version of the local binary pattern operator works in a 3×3 pixel block of an image. The pixels in this block are thresholded by its center pixel value, multiplied by powers of two and then summed to obtain a label for the center pixel. As the neighborhood consists of 8 pixels, a total of 28 = 256 different labels can be obtained depending on the relative gray values of the center and the pixels in the neighborhood.

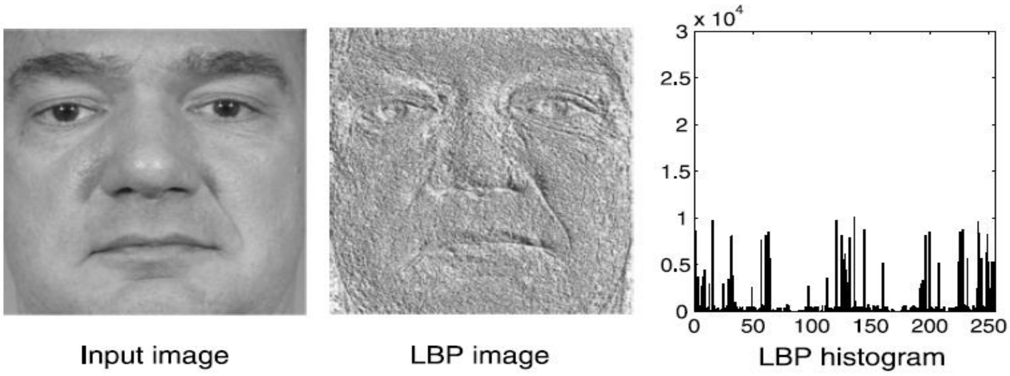

Figure 2.3 An illustration of the basic LBP operator.

In many texture analysis applications it is desirable to have features that are invariant or robust to rotations of the input image. As the LBPP,R patterns are obtained by circularly sampling around the center pixel, rotation of the input image has two effects: each local neighbourhood is rotated into other pixel location, and within each neighbourhood, the sampling points on the circle surrounding the center point are rotated into a different orientation.

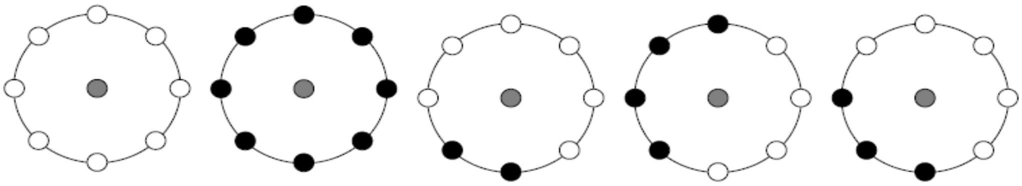

Figure 2.4 Different texture primitives detected by LBP.

Another extension to the original operator uses so called uniform patterns. For this, a uniformity measure of a pattern is used: U (“pattern”) is the number of bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. A local binary pattern is called uniform if its uniformity measure is at most 2.

For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and11001111 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. In uniform LBP mapping there is a separate output label for each uniform pattern and all the non-uniform patterns are assigned to a single label. Thus, the number of different output labels for mapping for patterns of P bits is $P(P - 1) + 3$. For instance, the uniform mapping produces 59 output labels for neighbourhoods of 8 sampling points, and 243 labels for neighbourhood of16 sampling points. The reasons for omitting the non-uniform patterns are twofold. First, most of the local binary patterns in natural images are uniform. Ojala et al. noticed that in their experiments with texture images, uniform patterns account for a bit less than 90% of all patterns when using the (8, 1) neighbourhood and for around 70% in the (16, 2) neighbourhood. In experiments with facial images [4] it was found that 90.6% of the patterns in the (8, 1) neighbourhood and 85.2% of the patterns in the (8, 2) neighbourhood are uniform.

The second reason for considering uniform patterns is the statistical robustness. Using uniform patterns instead of all the possible patterns has produced better recognition results in many applications. On one hand, there are indications that uniform patterns themselves are more stable, i.e. less prone to noise and on the other hand, considering only uniform patterns makes the number of possible LBP labels significantly lower and reliable estimation of their distribution requires fewer samples. The uniform patterns allows to see the LBP method as a unifying approach to the traditionally divergent statistical and structural models of texture analysis.

Each pixel is labelled with the code of the texture primitive that best matches the local neighbourhood. Thus each LBP code can be regarded as a micro-text on. Local primitives detected by the LBP include spots, flat areas, edges, edge ends, curves and so on. The combination of the structural and statistical approaches stems from the fact that the distribution of micro-textons can be seen as statistical placement rules. The LBP distribution therefore has both of the properties of a structural analysis method: texture primitives and placement rules. On the other hand, the distribution is just a statistic of a non-linearly filtered image, clearly making the method a statistical one. For these reasons, the LBP distribution can be successfully used in recognizing a wide variety of different textures, to which statistical and structural methods have normally been applied separately.

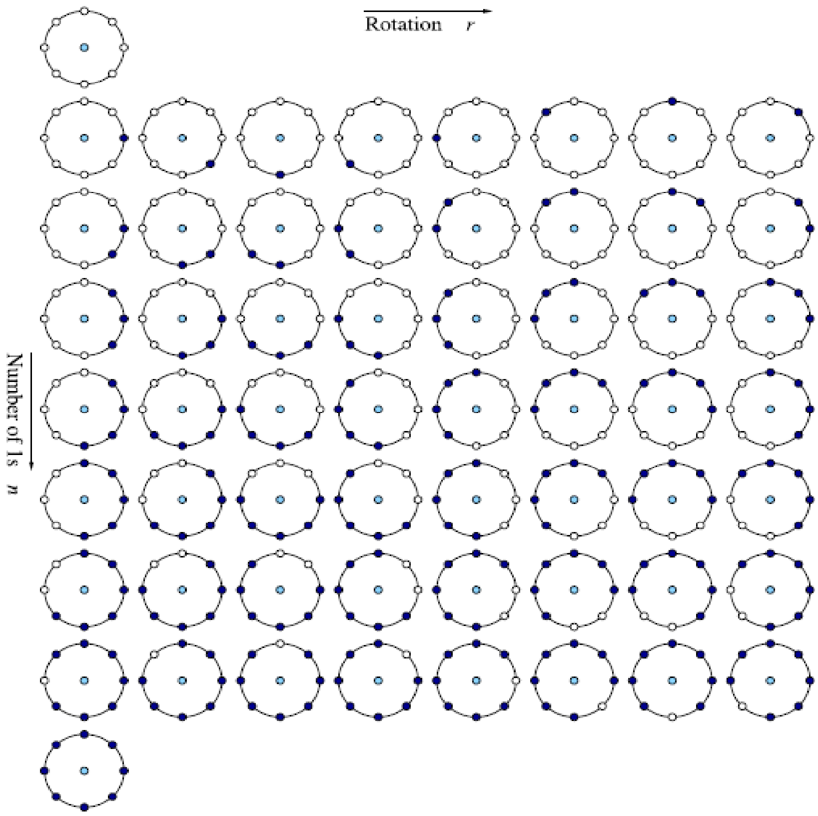

Figure 2.5 58 different patterns in (8,R) neighbourhood.

2.3 IMAGE PROCESSING

Image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing are also possible. In digital image processing, various information hiding schemes can be used to hide important data. Text, image, audio, and video can be represented as digital data. The explosion of Internet applications leads people into the digital world, and communication via digital data becomes recurrent. However, new issues also arise and have been explored, such as data security in digital communications, copyright protection of digitized properties, invisible communication via digital media, etc.

2.4 TEMPLATE MATCHING

Template matching is a technique in digital_image_processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control, a way to navigate a mobile robot, or as a way to detect edges in images.

Template matching was used in to perform face recognition using the LBP-based facial representation: a template is formed for each class of face images, and then a nearest-neighbour classifier is used to match the input image with the closest template. Here we first adopted template matching to classify facial expressions for its simplicity. In training, the histograms of expression images in a given class were averaged to generate a template for this class.

2.5 CURRENT TECHNOLOGY AND METHODOLOGY

Much of the work in computer recognition of faces has focused on detecting individual features such as the eyes, nose, mouth, and head outline, and defining a face model by the position, size, and relationships among these features. Such approaches have proven difficult to extend to multiple views, and have often been quite fragile, requiring a good initial guess to guide them.

- Research in human strategies of face recognition, moreover, has shown that individual features and their immediate relationships comprise an insufficient representation to account for the performance of adult human face identification (Carey & Diamond, 1977) put it in form. Nonetheless, this approach to face recognition remains the most popular one in the computer vision literature.

- Bledsoe(1966,b)was the first to attempt semi-automated face recognition with a hybrid human-computer system that classified faces on the basis of fiducially marks entered on photographs by hand. Parameters for the classification were normalized distances and ratios among points such as eye corners mouth corners, nose tip and chin point.

- Later work at Bell Labs (Goldstein, Harmon, & Lesk, 1971; Harmon, 1971) developed a vector of up to 21 features, and recognized faces using standard pattern classification techniques. The chosen features were largely subjective evaluations (e.g. shade of hair, length of ears, lip thickness) made by human subjects, each of which would be quite difficult to automate.

- An early paper by Fischler and Eischlager (1973)attempted to measure similar features automatically. They described a linear embedding algorithm that used local feature template matching and a global measure of fit to find and measure facial features. This template matching approach has been continued and improved by the recent work of Yuille, Cohen, and Hallinan (1989).Their strategy is based on "deformable templates," which are parameterized models of the face and its features in which the parameter values are determined by interactions with the image.

- Recent work by Burt (1988a, b) uses a smart sensing approach based on multi-resolution template matching. This coarse-to-fine strategy uses a special-purpose computer built to calculate multi-resolution pyramid images quickly, and has been demonstrated identifying people in near-real-time. This system works well under limited circumstances, but should suffer from the typical problems of correlation-based matching, including sensitivity to image size and noise. The face models are built by hand from face images.

3. ANALYSIS AND DESIGN

This chapter includes the detailed Analysis of the project which is further broken down into functional and non-functional requirements. It also consists of the proposed system which is implemented in this project. This chapter also focuses on the major design considerations and the design details which have been used in the project. Also the major implementation issues which had been encountered and their remedies have been specified.

3.1 REQUIREMENT ANALYSIS

In this section, the functional and non-functional requirements of the project have been analysed. Also, the hardware and software requirements for the development of this system have been mentioned.

3.1.1 Functional Requirements

A functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs.

1. Data acquisition

The input to the system, the images, can be provided by the users by means of a web camera. These images will be stored as a matrix containing details about each pixel.

2. Feature extraction

Features should be extracted from the images. These features are further, used to generate LBP labels and feature histograms which are then used by machine learning methods to create classes of expressions. Features extracted should be such that images of two individuals can be distinguished.

3. Training and Testing

Training and testing of the system is done by a set of pictures. The database is a collection of pictures from 10 subjects exhibiting expressions corresponding to the 6-class of emotions. Images of 9 subjects are used for training and the rest are used for testing.

3.1.2 Non-Functional Requirements

Non-functional requirements are called quality of the system. Other terms for non-functional requirements are “constraints”, ”quality goals”, ”quality of service requirements” and “non-behavioral requirements”.

1. Usability

The usability of the software depends on the following factors: The system must be robust yet flexible enough to cater for different images of the same person in different states of mind as the images can be a little different. Users must know how to interact with the hardware.

2. Reliability

There are always variations in the images of the same user. The reliability of the system depends on the fact that the system should be flexible enough to handle small variations in the images of the user and the system should be robust enough to identify the correct alphabet.

3. Performance

Following factors need to be considered in order to maintain system performance. The time taken to perform the image processing should also be taken into account. System speed must be fast enough to cater for larger databases and to perform recognition tasks efficiently. Memory should be large enough to cater for large databases. Greater memory size will result is better speed while performing various operations.

4. Extensibility

The system should be such that the implementation takes into consideration future growth.

5. Testability

Since the system is using template matching, it should implement faster once its templates are obtained. The histograms must be checked from time to time to make sure that the output is accurate.

6. Accuracy

The recognition of the expression must be accurate.

3.1.3 Hardware Requirements

- Webcam of resolution 1.3MP (recommended).
- 4GB RAM.
- Processor with a frequency of 1.5GHz (minimum).

3.1.4 Software Requirements

- MATLAB 7.0 (or higher).
- Operating System: Microsoft Windows XP/ Windows 7/Windows Vista.

3.2 PROJECT DESIGN

There are two parts in expression recognition. Part 1 is the training phase. In this phase a set of training images for each expression is passed to the system. The system then generates LBP features and creates Histograms or each training image. The histograms are then averaged to create a Class Template for each expression.

1. *Acquiring a set of training images*:

   The first step is generic for all biometric technologies training phase; it consists of acquiring images from a folder. These images are categorized based on the expssions they represent.

2. *Extracting features*:

   The generic second step is to extract the relevant data from the captured sample. For expression recognition there is the added difficulty that first the face has to be located within the acquired image. This can either be done manually by marking the location of the eyes or through the use of software. Once this has been accomplished, the features of the face can be extracted.

3. *Creating LBP features*:

   The image is subjected to thresholding with a 3x3 matrix with the central value. Any value in the matrix that is greater than central value is set to 1 and the rest are set to 0. This matrix when read in the clockwise direction gives a binary value which is converted to decimal. The values inside the matrix are substituted by this decimal number. Thus an LBP feature image is obtained.

4. *Crating Class Template*:

   The system creates histograms for each LBP feature image. The histograms of all the images belonging to a particular class of expression are averaged to obtain the class template. This template successfully describes the features of that particular class.

The Part 2 o the system is the testing phase. In this phase an unknown image is passed to the system. The system then generates LBP features and creates histogram for the image. The histogram is then compared to the various Class Templates and thus the expression is recognized.

1. *Acquiring a sample*:

   The first step is generic for all biometric technologies; it consists of a sensor taking an observation. In the case of 2D face recognition, the sensor is a camera and the observation is a photograph or series of photographs. This acquisition can be accomplished by acquiring an existing photograph or by taking a photograph of a live subject. As video is a rapid sequence of individual still images, it can also be used as a source of facial images, though at present the standard of image quality makes this less suitable.

2. *Extracting features*:

   The generic second step is to extract the relevant data from the captured sample. For micro-expression recognition there is the added difficulty that first the face has to be located within the acquired image. This can either be done manually by marking the location of the eyes or through the use of software. Once this has been accomplished, the features of the face can be extracted.

3. *Comparing templates*:

   For identification purposes, this step will be a comparison between the template captured from the subject at that moment and all the templates stored on a database.

4. *Declaring a match*:

   The facial expression recognition system will return a match.

3.2.1 Design Considerations

While designing the project, the following issues and challenges need to be considered:

- *Accurate Grayscale Intensity Matrix*

  Since the resolution of the image to be forwarded to the LBP operator must be low, accuracy in the intensity matrix must be maintained.

- *Relevant Expressions*

  Only relevant images should be used in the training phase. Relevant pictures consist of expressions corresponding to the 6 basic emotions.

- *Effective Templates*

  Certain expressions are ambiguous, i.e. expressions belonging to two different classes might seem similar. Thus template must be efficiently formed to avoid errors.

- *Time Restriction*

  The template matching takes long time to give the output. A fast processor and a large RAM are required.

3.2.2 Design Details

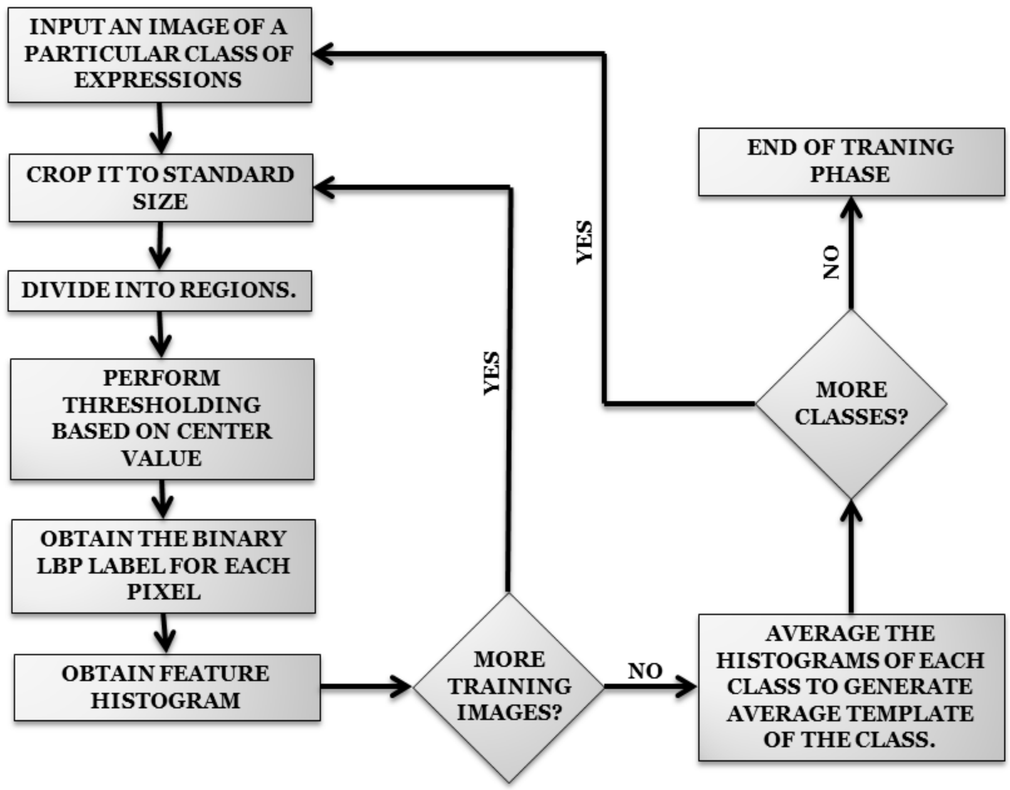

Figure 3.1 Block diagram for training phase.

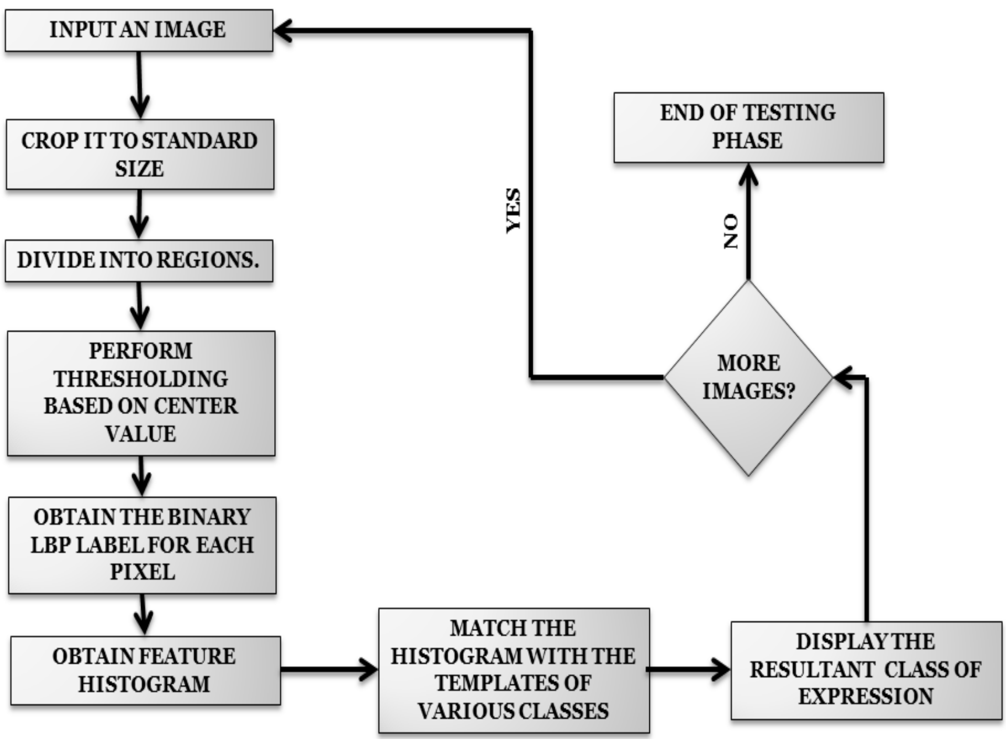

Figure 3.2 Block diagram for testing phase.

3.2.2.1 Training Stage

Images of people with facial expressions of 480x640 pixels are captured from the web cam as shown in Figure 3.3 (a), (b) and (c). These can be different shapes. 30 images will be used for creating the templates.10 images corresponding to every emotion respectively.

Input Samples from different Users for different Classes of expression:

Class 1: Happiness

(a) Input set for Class 1

Class 2: Sadness

(b) Input set for User 2

Class 3: Surprise

(c) Input set for User 3

Figure 3.3 Input images from different users for different classes of expressions.

The training stage has the following steps:

1. Acquire the image

Acquire the required image from the computer by finding it in the required directory.Or click a picture using a webcam and select it as input to the software.

2. Crop Image

The image may have various background data which needs to be eliminated. Thus, the image is cropped for the relevant data. The algorithm used will scan every horizontal line for a lighter pixel. When it detects a lighter pixel, it will crop everything above that horizontal line. Similarly for left and right borders cropping is done. Thus the grayscaled image is cropped.

Figure 3.4 Cropping of the input image.

3. Use the LBP operator

The original LBP operator was introduced by Ojala et al, and was proved a powerful means of texture description. The operator labels the pixels of an image by thresholding a 3 x 3 neighborhood of each pixel with the center value and considering the results as a binary number and the256-bin histogram of the LBP labels computed over a region is used as a texture descriptor. The derived binary numbers (called Local Binary Patterns or LBP codes) codify local primitives including different types of curved edges, spots, flat areas, etc., so each LBP code can be regarded as a micro-texton. The limitation of the basic LBP operator is its small 3 x 3 neighborhood which cannot capture dominant features with large scale structures. Hence the operator later was extended to use neighborhood of different sizes. Using circular neighborhoods and bilinearly interpolating the pixel values allow any radius and number of pixels in the neighborhood. See Fig. 5 for examples of the extended LBP operator, where the notation (P, R) denotes a neighborhood of P equally spaced sampling points on a circle of radius of R that form a circularly symmetric neighbor set.

The LBP operator $LBP_{P,\ R}$ produces 2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighbor set. It has been shown that certain bits contain more information than others. Therefore, it is possible to use only a subset of the 2^P Local Binary Patterns to describe the texture of images. Ojala et al. called these fundamental patterns as uniform patterns. A Local Binary Pattern 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. For example, 00000000, 001110000 and 11100001 are uniform patterns. It is observed that uniform patterns account for nearly90% of all patterns in the (8, 1) neighborhood and for about 70%in the (16, 2) neighborhood in texture images. Accumulating the patterns which have more than 2 transitions into a single bit yields an LBP operator, denoted $LBP^{u2}_{P,R}$, with less than 2^P bins. For example, the number of labels for a neighborhood of 8 pixels is 256 for the standard LBP but 59 for LBP^{u2}.

After labeling an image with the LBP operator, a histogram of the labeled image f(x, y) can be defined as:

$$H_i=\Sigma I(f(x,y)=i), \quad i= 0,1,2,3,\ldots,n-1$$

where n is the number of different labels produced by the LBP operator; and

$$I(A) = \begin{cases} 1 & A \text{ is true} \\ 0 & A \text{ is false} \end{cases}$$

This LBP histogram contains information about the distribution of the local micro-patterns, such as edges, spots and flat areas, over the whole image, so can be used to statistically describe image characteristics.

Face images can be seen as a composition of micro-patterns which can be effectively described by the LBP histograms. Therefore, it is intuitive to use LBP features to represent face images. A LBP histogram computed over the whole face image encodes only the occurrences of the micro-patterns without any indication about their locations. To also consider shape information of faces, face images were equally divided into small regionsR_0, R_1, . . . ,R_m to extract LBP histograms. The

LBP features extracted from each sub-region are concatenated into a single, spatially enhanced feature histogram.

The extracted feature histogram represents the local texture and global shape of face images. Some parameters can be optimized for better feature extraction. One is the LBP operator, and the other is the number of regions divided. Following the setting in, we selected the 59-bin LBP^{u2}8;2 operator, and divided the110 x 150 pixels face images into 18 x 21 pixels regions, giving a good trade-off between recognition performance and feature vector length. Thus face images were divided into 42(6 x 7) regions as shown in Fig 7, and represented by the LBP histograms with the length of 2478(59x42).

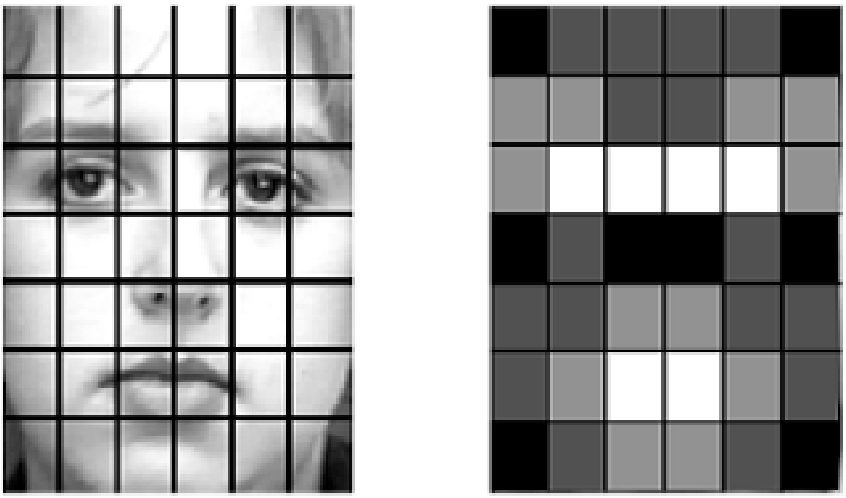

Figure 3.5 (Left) A face image divided into 6 x 7 sub-region. (Right) The weights set for weighted dissimilarity measure. Black squares indicate weight 0.0, dark grey 1.0, light grey 2.0 and white 4.0.

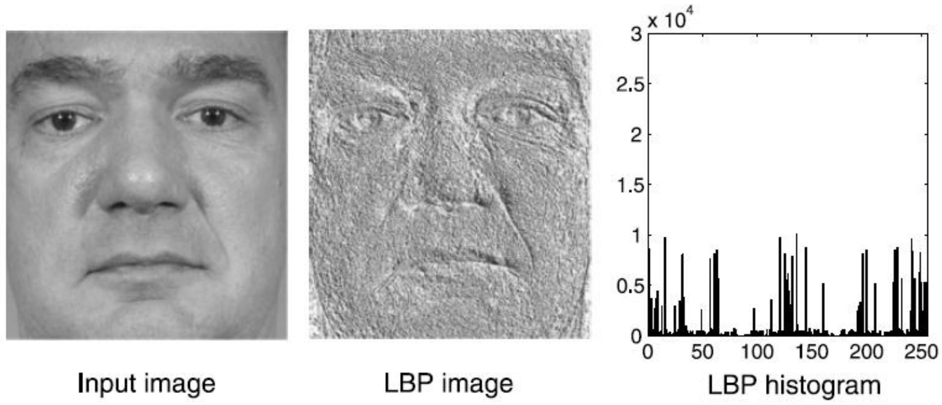

Figure 3.6 An illustration of the basic LBP operator.

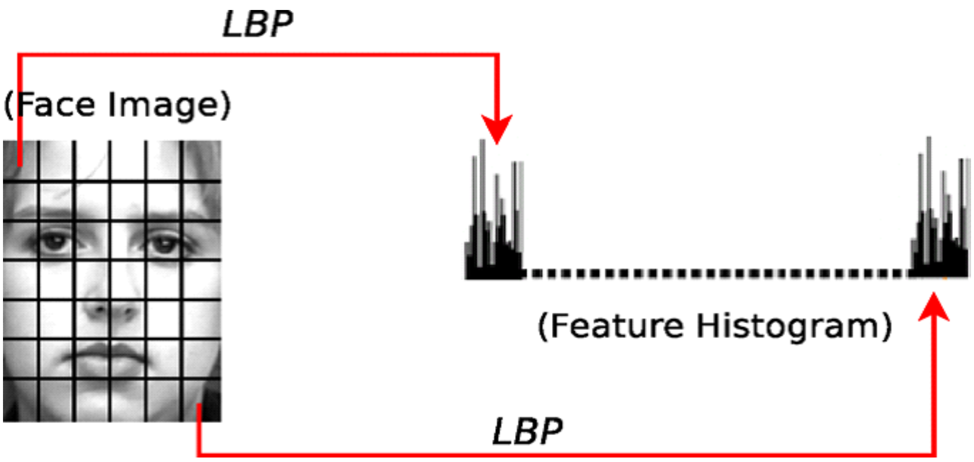

Figure 3.7 A face image is divided into small regions from which LBP histograms are extracted and concatenated into a single, spatially enhanced feature histogram.

6. Creating the class template

Template matching was used in to perform face recognition using the LBP-based facial representation: a template is formed for each class of face images, then a nearest-neighbor classifier is used to match the input image with the closest template. Here we first adopted template matching to classify facial expressions for its simplicity. In training, the histograms of expression images in a given class were averaged to generate a template for this class.

Following, we also selected the Chi square statistic as the dissimilarity measure for histograms:

$$\chi^2 = \Sigma w_j \, (S_{ij} - M_{ij})^2 \, / \, S_{ij} + M_{ij}$$

where S and M are two LBP histograms, and w_j is the weight for region j.

The template matching achieved the generalization performance of 79.1% for the 7-class task and 84.5% for the 6-class task. We compared the results with that reported in [11], where Cohen et al. adopted Bayesian network classifiers to classify 7-class emotional expressions based on the tracked geometric facial features(eyebrows, eyelids and mouth). They carried out 5-fold cross-validation on a subset of 53 subjects from the Cohn–Kanade database, and obtained the best performance of 73.2% by using Tree-Augmented-Naive Bayes (TAN) classifiers. We can observe that Joy and Surprise can be recognized with high accuracy (around 90–92%), but Anger and Fear are easily confused with others.

3.2.2.2 Recognition Stage

The testing image is captured using a webcam. The Recognition stage has the same steps as Training Stage except for the classification step which is explained below.

Classification Parameter using the Templates

The feature histogram of the input images is compared with the templates of various classes. The template matching method uses nearest-neighbor matching. It uses the chi square parameter for dissimilarity measure.

$$\chi^2 = \Sigma w_j \, (S_{ij} - M_{ij})^2 \, / \, S_{ij} + M_{ij}$$

3.2.3 ACTIVITY DIAGRAM

Figure 3.8 shows the activity diagram of the Facial Expression Recognition System. The blocks in the diagram represent the modules in the Facial Expression Recognition System.

The main functions of these blocks are as given below:

- *Pre-processing* – To pre-process the images and make them ready for LBP operator.
- *Use of LBP operator* – To create LBP features which are used to create feature histogram.
- *Template creation* – Average the histograms of all the images belonging to a particular class and obtain an average template.
- *Recall* – To test the performance of the system when the data provided as input is the same as on which the templates are made.
- *Generalization of the EPBTA Neural Network* – To test the performance of the system when the data provided as input is other than that on which the template was made.

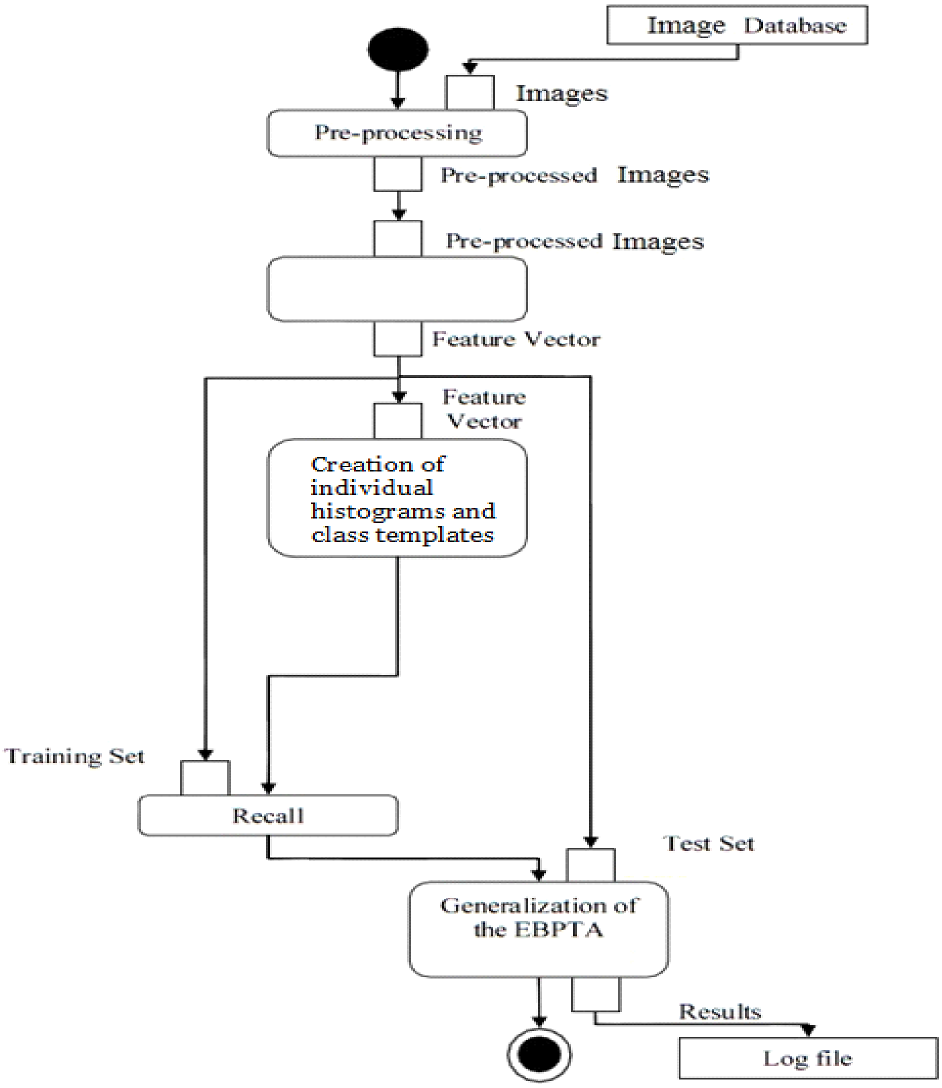

Figure 3.8 Activity Diagram of Facial Expression Recognition System

3.3 GUI DESIGN

A simple GUI for feeding an image and showing the output is designed as shown in Figure 3.9. The user need not know the entire process of the recognition. The image capture is done as input and the recognized expression is displayed as output. All the other processes are done internally.

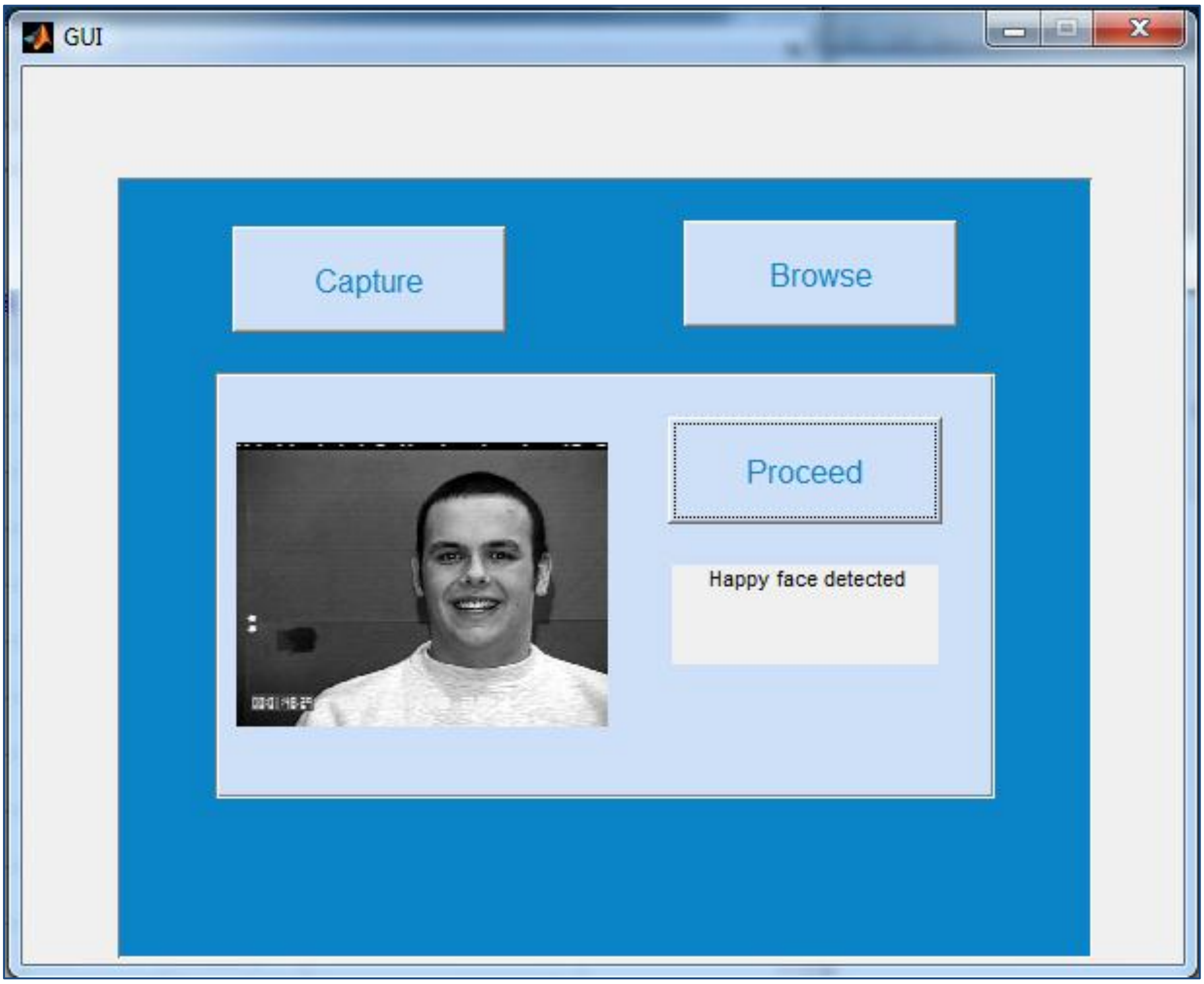

Figure 3.9 GUI design.

4. IMPLEMENTATION AND RESULTS

This chapter includes the details related to implementation and explains the results produced after the implementation of the project. This chapter also focuses on the major implementation issues which had been encountered during the implementation and their remedies.

4.1 IMPLEMENTATION DETAILS

In this section, the steps performed and the algorithms used in each of the module of the project have been discussed.

4.1.1 Steps for Pre-Processing

Input: Image of resolution 640*480 obtained from the webcam or the directories.

Output: Grayscale images of a cropped face and features of size 60*30.

Step 1: Accept the images from the directory or the webcam.

Step 2: If it is an rgb image, convert it to grayscale by using rgb2gray() function.

Step 3: Detect the face using the vision toolbox.

Step 4: Crop the face and store it in a folder called croppedface.

Step 5: Extract features like eyes and lips.

Step 6: Resize these facial feature images to 60 x 30.

Step 4: Store the eyes in a folder called cropped eyes and lips in a folder called cropped lips.

4.1.2 Algorithm for use of LBP operator

Input: Grayscale image of size 60*30.

Output: LBP feature.

Step 1: Read the detected image

Step 2: From the matrix values of the binary image label() uses the 8 connectivity to find the connected neighbours.

Step 3: Perform thresholding with 3 x 3 matrix using the enter vale of the pixel.

Step 4: All the neighbouring values greater than the center value are set to 1 while the rest are made 0.

Step 5: Each 3 x 3 matrix give a binary LBP label.

4.1.3 Algorithm of feature histogram generation

Input: LBP image with LBP labels.

Output: Feature histogram.

Step 1: Read the LBP image

Step 2: Generate histogram of the image using imhist (img).

4.1.4 Algorithm of template generation

Input: Feature histogram of a particular class

Output: Template of a class

Step 1: Read all the histograms of a particular class.

Step 2: Create a template by averaging these histograms.

4.1.4 Algorithm for Recognition

Input: Image.

Output: Expression recognized.

Step 1: Pre-process the image an apply LBP operator and generate histograms

Step 2: Compare it with the templates of various classes.

Step 3: Display the result.

4.2 RESULTS

In this section the results of accepting the image, cropping, use of LBP operator and creation of template and Output Display. The results obtained are as follows.

4.2.1 Result of acquiring the image

This subsection displays the output of the image it has been acquired and converted to grayscale.

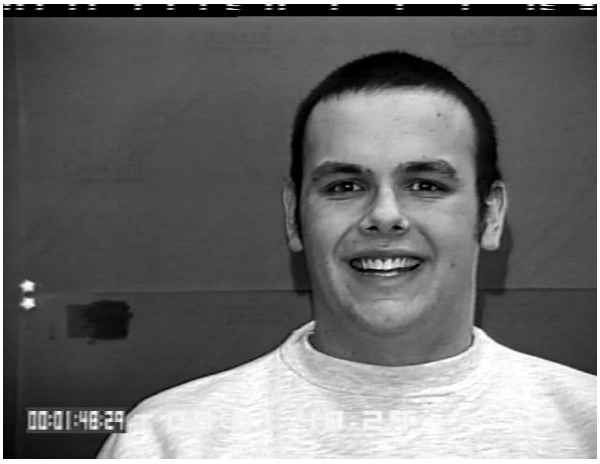

Figure 4.1 Input image

4.2.1 Result of cropping the face

This subsection provides the output of the image after the face has been detected and cropped so as to eliminate unwanted details in the image.

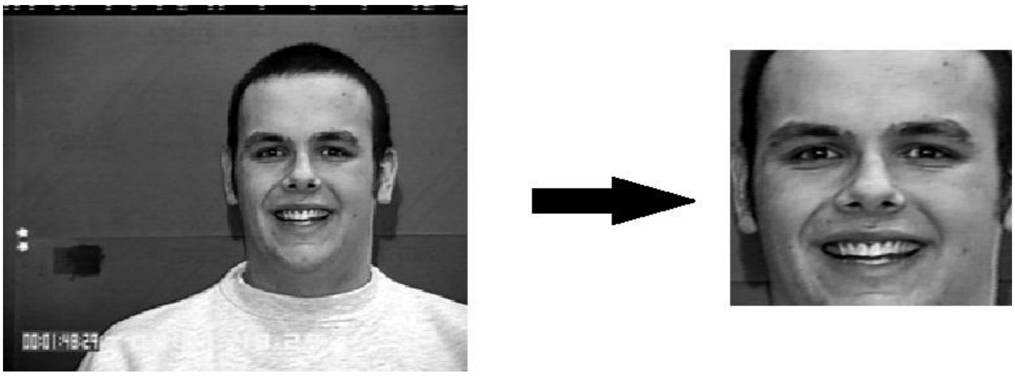

Figure 4.2 Cropped image

4.2.3 Result of cropping the face

This subsection provides the output of the image after essential features like eyes and lips have been extracted.

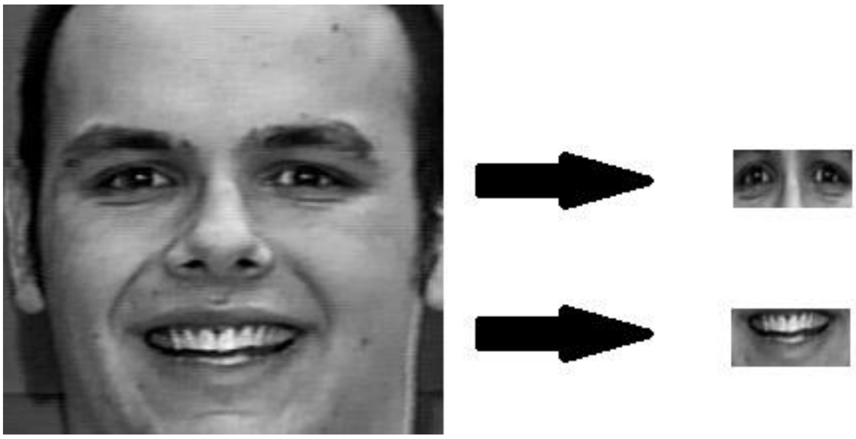

Figure 4.3 An image displaying extracted features

4.2.4 Result of cropping the face

This subsection provides the output of the image after thresholding with the LBP operator.

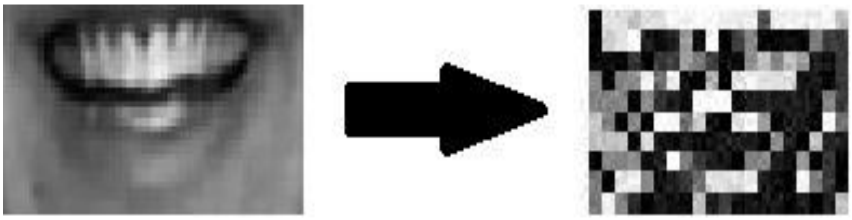

Figure 4.4 An illustration of the the generated LBP

4.2.5 Histogram of the LPB features

This subsection provides the output of the histogram of image after thresholding with the LBP operator.

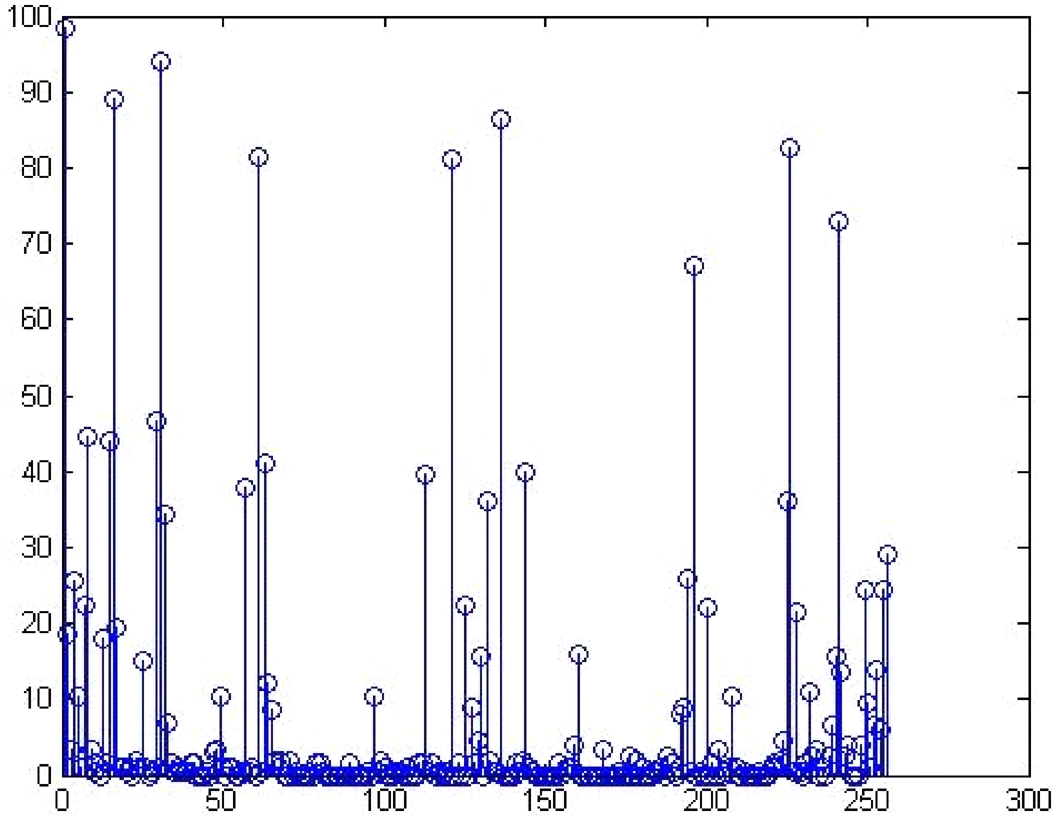

Figure 4.5 Histogram of the LBP features.

4.2.4 Result of cropping the face

This subsection provides the output of the system. It tells us what kind of expression is being exhibited by the face in the input image.

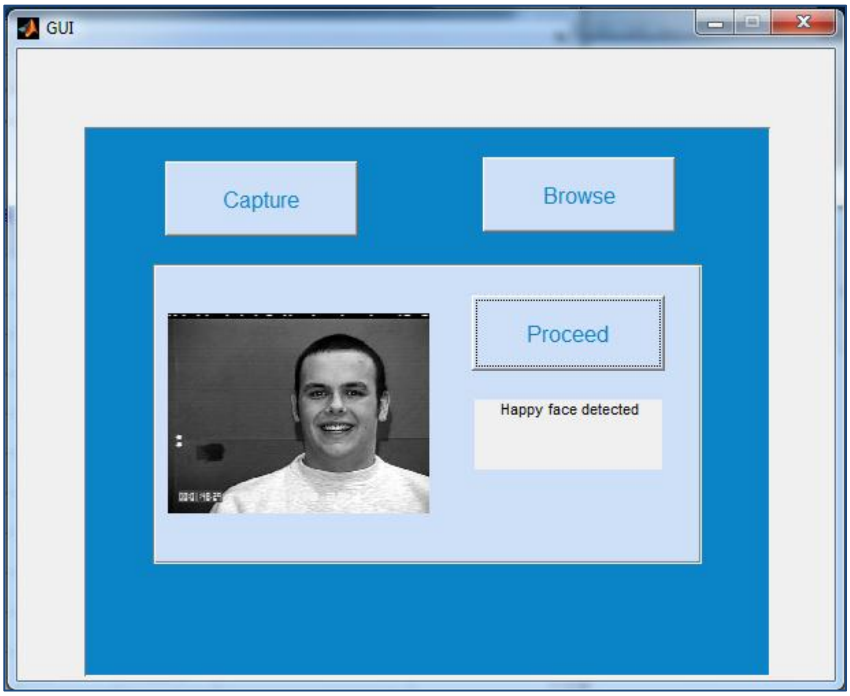

Figure 4.6 Output on the GUI showing detected expression.

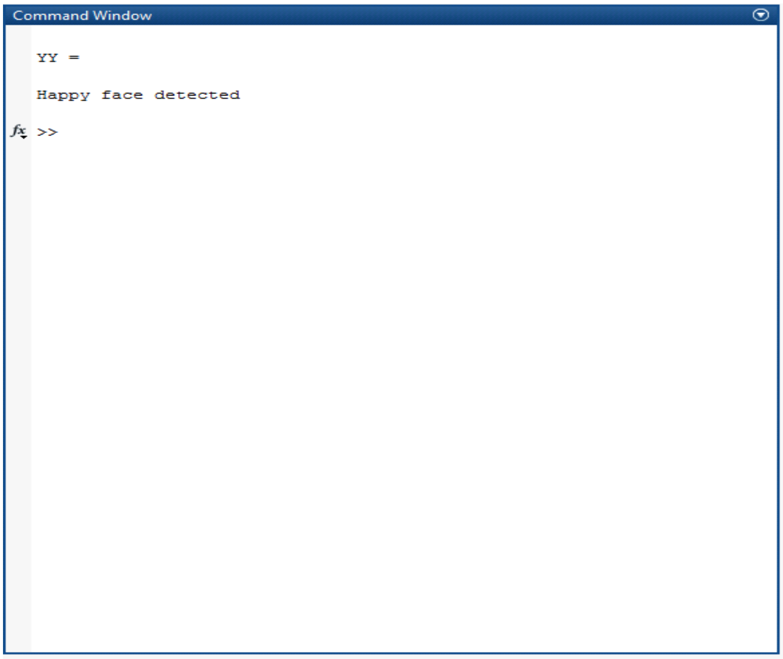

Figure 4.7 Output on the command window showing detected expression.

5. TESTING

This chapter focuses on testing of the Facial Recognition System. This chapter includes the details of the testing phase that followed the implementation. It includes various levels of testing such as unit testing and system testing.

5.1 UNIT TESTING

Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

The following units in the system were tested individually:

5.1.1 Capturing Device

The capturing device or the webcam was tested and ensured of correct image capturing capability with different environments.

5.1.2 Training Data

The training data is checked and verified before integrating the entire system.

5.1.3 Testing and Output

The testing and display is verified separately before the system integration.

5.2 SYSTEM TESTING

System testing tests a completely integrated system to verify that it meets its requirements. The system was tested with the help of the following tests:

5.2.1 Testing to checking if it recognizes images from the training set

The system is tested with images from the training set. The output provided is correct. Thus the system fares well in this testing.

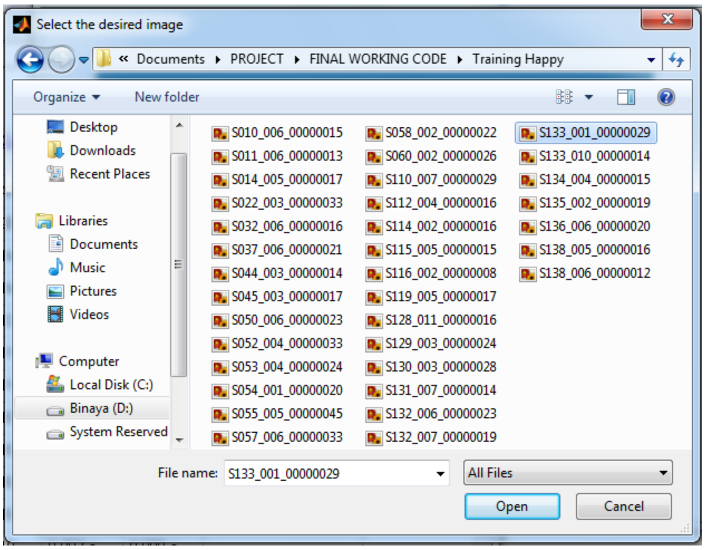

Figure 5.1 Select training images for testing.

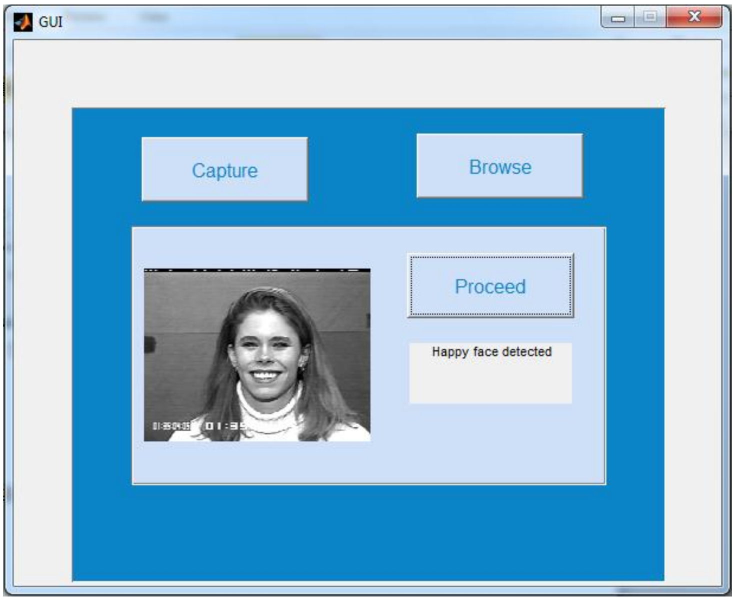

Figure 5.2 Output on the GUI when training image is used for testing.

5.2.1 Testing to checking if it recognizes images not from the training set

The system is tested with an unknown image. This image is not present in the training set. The output provided is correct. Thus the system fares well in this testing.

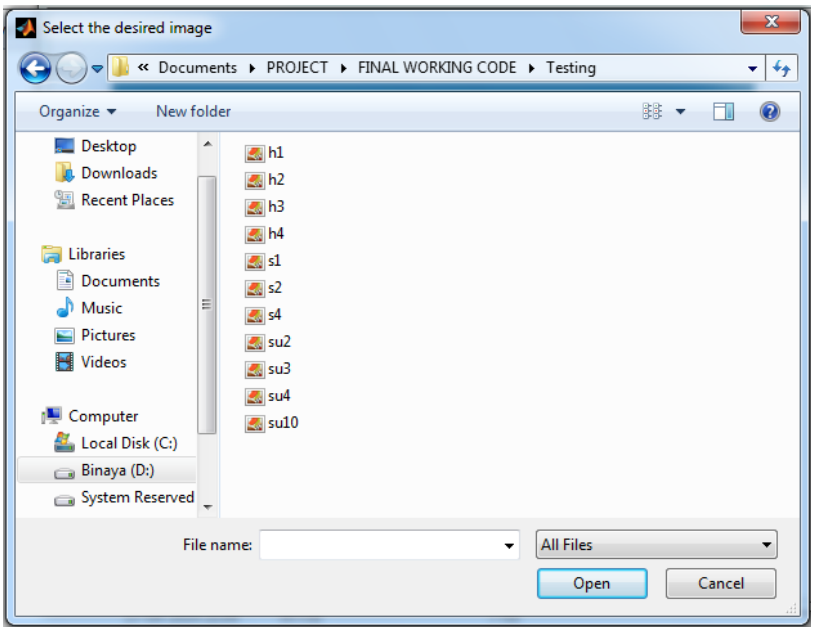

Figure 5.3 Select unknown images for testing.

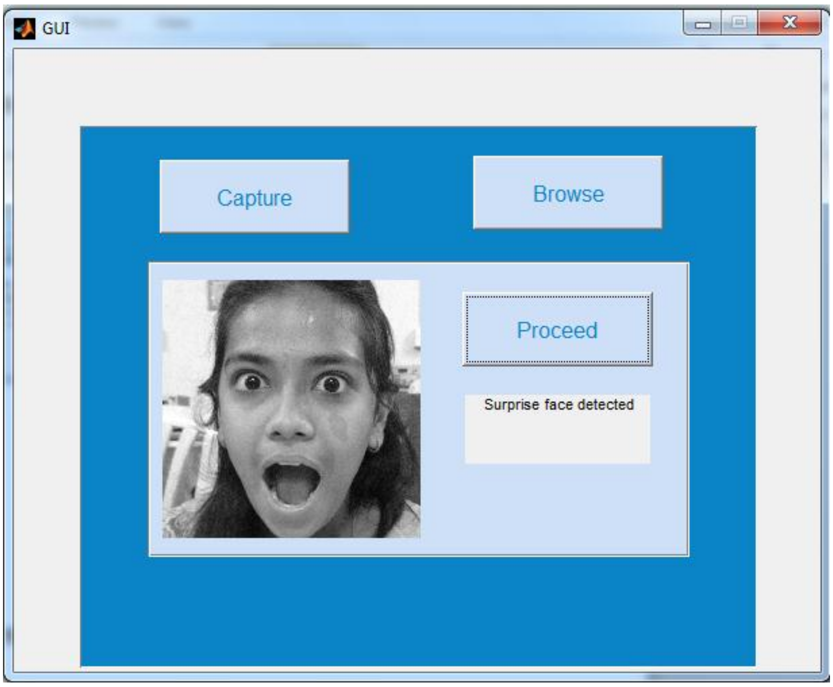

Figure 5.4 Output on the GUI when unknown image is used for testing.

6. CONCLUSION AND FURTHER WORK

This chapter majorly focuses on the conclusion and further work which can be carried out in this domain with a view to enhance this project.

6.1 Conclusion

A Facial Expression Recognition System was designed using Image Processing and Template matching method. LBP was used to detect important features and thereby create histograms which are averaged to create class templates.

Two different image databases were used in this project. The first thirty five face samples (images) of each emotion were taken from thirty five different individuals and used to create the class template. The images showed variations like shape and size of the face, color of the skin, etc varying from person to person. The templates were efficiently created due to the large variation provided.

LBP operator was used to detect micro-patterns in the image. This was done by thresholding the entire image by a 3 x 3 matrix with the central value. The points greater than the central value were set to 1 and the others to 0. This gave a binary label called the LBP labels. We reduced the number of labels by grouping uniform and non-uniform labels thus giving 59 output labels for an 8-neighborhood image.

The system performs fairly well even in low resolution since LBP is independent of illumination intensity. The background of the images is cropped out thus it poses no effect on the overall result

The next one samples of each emotion were taken from 10 different users and used for testing. The accuracy of the system came out to be above 80%.

6.2 Further Work

Even though the system successfully recognizes expressions it is still very far from being complete. It works on still images shot under ideal conditions of lighting.

This technology can be extended by adapting it to recognize expression on images extracted from videos and also to recognize expression in real-time.

Also the software requires full frontal view. Further advancements can be done to recognize expressions in partial frontal view or even side view.

REFERENCES

[1] M. Pantic, L. Rothkrantz, Automatic analysis of facial expressions: the state of art, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1424–1445.

[2] B. Fasel, J. Luettin, Automatic facial expression analysis: a survey, Pattern Recognition 36 (2003) 259–275.

[3] M. Pantic, L. Rothkrantz, Toward an affect-sensitive multimodal human–computer interaction, in: Proceeding of the IEEE, vol. 91, 2003, pp. 1370–1390.

[4] Y. Tian, T. Kanade, J. Cohn, Handbook of Face Recognition, Springer, 2005 (Chapter 11. Facial Expression Analysis).

[5] Y. Yacoob, L.S. Davis, Recognizing human facial expression from long image sequences using optical flow, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (6) (1996) 636–642.

[6] I. Essa, A. Pentland, Coding, analysis, interpretation, and recognition of facial expressions, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 757–763.

[7] M.J. Lyons, J. Budynek, S. Akamatsu, Automatic classification of single facial images, IEEE Transactions on Pattern Analysis and Machine Intelligence 21(12) (1999) 1357–1362.

[8] G. Donato, M. Bartlett, J. Hager, P. Ekman, T. Sejnowski, Classifying facial actions, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (10) (1999) 974–989.

[9] M. Pantic, L. Rothkrantz, Expert system for automatic analysis of facial expression, Image and Vision Computing 18 (11) (2000) 881–905.

[10] Y. Tian, T. Kanade, J. Cohn, Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 97–115.

[11] I. Cohen, N. Sebe, A. Garg, L. Chen, T.S. Huang, Facial expression recognition from video sequences: temporal and static modeling, Computer Vision and Image Understanding 91 (2003) 160–187.

[12] L. Yin, J. Loi, W. Xiong, Facial expression representation and recognition based on texture augmentation and topographic masking, in: ACM Multimedia, 2004.

[13] M. Yeasin, B. Bullot, R. Sharma, From facial expression to level of interests: a spatio-temporal approach, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

[14] J. Hoey, J.J. Little, Value directed learning of gestures and facial displays, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

[15] Y. Chang, C. Hu, M. Turk, Probabilistic expression analysis on manifolds, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

[16] R.E. Kaliouby, P. Robinson, Real-time inference of complex mental states from facial expressions and head gestures, in: IEEE CVPR Workshop on Real-time Vision for Human–Computer Interaction, 2004.

ACKNOWLEDGEMENT

We are thankful to Ms. Urvi Kore, Assistant Professor, Computer Engineering Department, without whose help, guidance and motivation; we could not have proceeded with this project. She was very helpful and took a keen interest in our project topic. We are grateful to her for her help and guidance to make this project report complete.

We would like to thank the H.O.D. of our department, Mr. Jayant Gadge for providing good facilities especially regarding lab work. We value the guidance, feedback and contributions given by the panel of examiners and all those who have directly and indirectly helped us in our project.

Binaya Swain

Arjun Nair

Rishi Rajani

SUMMARY

Facial expression is one of the most powerful, natural and immediate means for human beings to communicate their emotions and intensions. Automatic facial expression analysis is an interesting and challenging problem, and impacts important applications in many areas such as human–computer interaction and data-driven animation. Due to its wide range of applications, automatic facial expression recognition has attracted much attention in recent years. Though much progress has been made, recognizing facial expression with a high accuracy remains difficult due to the subtlety, complexity and variability of facial expressions.

A Facial Expression Recognition System was designed using Image Processing and Template matching method. LBP was used to detect important features and thereby create histograms which contribute to the template of the class.

Two different image databases were used in this project. The first nine face samples (images) of each emotion were taken from nine different individuals and used to create the class template. The images showed variations like shape and size of the face, color of the skin, etc varying from person to person. The templates were efficiently created due to the large variation provided.

LBP operator was used to detect micro-patterns in the image. This was done by thresholding the entire image by a 3 x 3 matrix with the central value. The points greater than the central value were set to 1 and the others to 0. This gave a binary label called the LBP labels. We reduced the number of labels by grouping uniform and non-uniform labels this giving 58 output labels for a 8-neighborhood image.

The system performs fairly well even in low resolution since LBP is independent of illumination intensity. The background of the images is cropped out thus it poses no effect on the overall result.