## CHAPTER 1

**INTRODUCTION**

* 1. **Statement of the problem**

In this project, the exact facial Eye Movement stress is identified from a fuzzy domain. Identification of the exact facial Eye Movement from a blurred facial image is not an easy task. Second, segmentation of a facial image into regions of interest is difficult, particularly when the regions do not have significant differences in their imaging attributes. Third, unlike humans, machines usually do not have visual perception to map facial Eye Movement stresss into emotions.

## Scope of the problem

This project also proposes a scheme for controlling emotion by judiciously selecting appropriate audiovisual stimulus for presentation before the subject. The selection of the audiovisual stimulus is undertaken using fuzzy logic. Experimental results show that the proposed control scheme has good experimental accuracy and repeatability. Experimental results show that the detection accuracies of emotions for adult male, adult female, and children of 8–12 years are as high as 88%, 92%, and 96%, respectively, outperforming the percentage accuracies of the existing techniques .

## Aim of the project

Currently available human–computer interfaces do not take complete advantage of these valuable communicative media and thus are unable to provide the full benefits of natural interaction to the users. Human–computer interactions could significantly be improved if computers could recognize the emotion of the users from their facial Eye Movement stresss and hand gestures, and react in a friendly manner according to the users’ needs. This project aims to recognize emotions in human subjects on a computer, whose facial Eye Movement stresss are analyzed by segmenting and localizing the individual frames into regions of interest.

## Plans for delivering the project goals

The project will involve 4stages:analysis and requirements, design, implementation and evaluation.

### Analysis and requirements

During this stage a study will be conducted on fuzzy logic and PYTHON and Python technology with the following objectives:

* To obtain an understanding of fuzzy logic.
* To learn the PYTHON and Python technology with the view of using it for emotion recognition.
* To develop a specification of fuzzy logic in emotion recognition.
* To critically examine previous work conducted on integrating PYTHON and Python technology.

### Design

During this stage, a design will be developed for the specification conceived in the previous stage. In particular, the following outcomes of this stage are:

* Identification of architecture for the system required by the application.
* Assumptions are identified.
* Alternatives for the system components researched and evaluated.
* Documentation of a conceptual design for all system components and overall design for integration of these components. This may include documentation of design methodologies.

### Implementation

During this stage, the actual implementation of the system design will take place. Problems in implementing the original design will be identified and design modifications conceived and tested. A working implementation that meets the specification will hopefully be the outcome of this stage.

### Evaluation

In this stage, the finished system will be evaluated for the value brought to the user. In addition, the design methodologies will be evaluated as to there effectiveness in delivering the required outcomes. Finally the project will be evaluated as to how well the goals of the project were achieved.

## CHAPTER 2

**LITERATURE SURVEY**

* 1. **Fuzzy Logic**

Fuzzy logic is a form of [many-valued logic](http://en.wikipedia.org/wiki/Many-valued_logic); it deals with [reasoning](http://en.wikipedia.org/wiki/Reasoning) that is fluid or approximate rather than fixed and exact. In contrast with "crisp logic", where [binary](http://en.wiktionary.org/wiki/binary) sets have [two-valued logic:](http://en.wikipedia.org/wiki/Two-valued_logic) true or false, fuzzy logic variables may have a [truth value](http://en.wikipedia.org/wiki/Truth_value) that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false.Furthermore, when [linguistic v](http://en.wikipedia.org/wiki/Linguist)ariables are used, these degrees may be managed by specific functions.

Fuzzy logic began with the 1965 proposal of [fuzzy set theory](http://en.wikipedia.org/wiki/Fuzzy_set_theory) by [Lotfi Zadeh](http://en.wikipedia.org/wiki/Lotfi_Zadeh). Fuzzy set theory defines fuzzy operators on fuzzy sets. The problem in applying this is that the appropriate fuzzy operator may not be known [2].

**1.1 PYTHON**

Python is a **high-level, interpreted**, **interactive** and **object-oriented scripting** **language**. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

1

## 1.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## 1.3 Python Features

Python's features include:

* **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
* **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

2

* **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases:** Python provides interfaces to all major commercial databases.
* **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable:** Python provides a better structure and support for large programs than shell scripting.

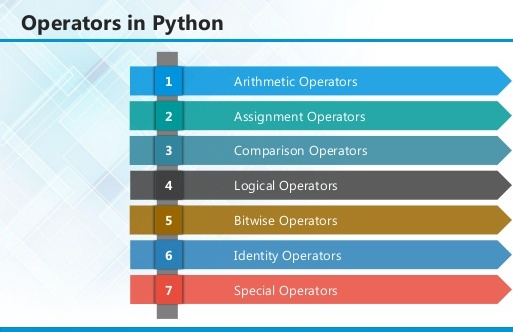
Python has a big list of good features:

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3

**Chapter 2**

**OPERATORS**



4

**2.1 ARITHMETIC OPERATORS**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + Addition | Adds values on either side of the operator. | a + b = 30 |
| - Subtraction | Subtracts right hand operand from left hand operand. | a – b = -10 |
| \* Multiplication | Multiplies values on either side of the operator | a \* b = 200 |
| / Division | Divides left hand operand by right hand operand | b / a = 2 |
| % Modulus | Divides left hand operand by right hand operand and returns remainder | b % a = 0 |
| \*\* Exponent | Performs exponential (power) calculation on operators | a\*\*b =10 to the power 20 |
| // | Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity): | 9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0 |

5

**2.2ASSIGNMENT OPERATOR**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = | Assigns values from right side operands to left side operand | c = a + b assigns value of a + b into c |
| += Add AND | It adds right operand to the left operand and assign the result to left operand | c += a is equivalent to c = c + a |
| -= Subtract AND | It subtracts right operand from the left operand and assign the result to left operand | c -= a is equivalent to c = c - a |
| \*= Multiply AND | It multiplies right operand with the left operand and assign the result to left operand | c \*= a is equivalent to c = c \* a |
| /= Divide AND | It divides left operand with the right operand and assign the result to left operand | c /= a is equivalent to c = c / ac /= a is equivalent to c = c / a |

**6**

|  |  |  |
| --- | --- | --- |
| %= Modulus AND | It takes modulus using two operands and assign the result to left operand | c %= a is equivalent to c = c % a |
| \*\*= Exponent AND | Performs exponential (power) calculation on operators and assign value to the left operand | c \*\*= a is equivalent to c = c \*\* a |
| //= Floor Division | It performs floor division on operators and assign value to the left operand | c //= a is equivalent to c = c // a |

**2.3 IDENTITY OPERATOR**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| is | Evaluates to true if the variables on either side of the operator point to the same object and false otherwise. | x is y, here **is** results in 1 if id(x) equals id(y). |
| is not | Evaluates to false if the variables on either side of the operator point to the same object and true otherwise. | x is not y, here **is not** results in 1 if id(x) is not equal to id(y |

**7**

**2.4 COMPARISON OPERATOR**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| & Binary AND | Operator copies a bit to the result if it exists in both operands | (a & b) (means 0000 1100) |
| | Binary OR | It copies a bit if it exists in either operand. | (a | b) = 61 (means 0011 1101) |
| ^ Binary XOR | It copies the bit if it is set in one operand but not both. | (a ^ b) = 49 (means 0011 0001) |
| ~ Binary Ones Complement | It is unary and has the effect of 'flipping' bits. | (~a ) = -61 (means 1100 0011 in 2's complement form due to a signed binary number. |
| << Binary Left Shift | The left operands value is moved left by the number of bits specified by the right operand. | a << 2 = 240 (means 1111 0000) |
| >> Binary Right Shift | The left operands value is moved right by the number of bits specified by the right operand. | a >> 2 = 15 (means 0000 1111) |

**8**

**2.5 LOGICAL OPERATOR**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| and Logical AND | If both the operands are true then condition becomes true. | (a and b) is true. |
| or Logical OR | If any of the two operands are non-zero then condition becomes true. | (a or b) is true. |
| not Logical NOT | Used to reverse the logical state of its operand. | Not(a and b) is false. |

## 2.6 Membership Operators

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| in | Evaluates to true if it finds a variable in the specified sequence and false otherwise. | x in y, here in results in a 1 if x is a member of sequence y. |
| not in | Evaluates to true if it does not finds a variable in the specified sequence and false otherwise. | x not in y, here not in results in a 1 if x is not a member of sequence y. |

**9**

## Python Operators Precedence

|  |  |
| --- | --- |
| **Operator** | **Description** |
| \*\* | Exponentiation (raise to the power) |
| ~ + - | Complement, unary plus and minus (method names for the last two are +@ and -@) |
| \* / % // | Multiply, divide, modulo and floor division |
| + - | Addition and subtraction |
| >> << | Right and left bitwise shift |
| & | Bitwise 'AND' |  |
| ^ | | Bitwise exclusive `OR' and regular `OR' |  |
| <= < > >= | Comparison operators |  |
| <> == != | Equality operators |  |
| = %= /= //= -= += \*= \*\*= | Assignment operators |  |
| is is not | Identity operators |  |
| in not in | Membership operators |  |
| not or and | Logical operators |  |

**10**

**Chapter 3**

## CHAPTER 3

**SYSTEM REQUIREMENTS AND SPECIFICATION**

Requirement specification is the activity of translating the information gathered during analysis into a requirement document.

### Classification

* + - User Requirements
    - System Requirements

#### User Requirements

User requirements is abstract statements of the system requirements for the customer and end user of the system who do not have a detailed technical knowledge of the system.

* + - * The device should provide option for selecting the company code.
      * Provision should be provided to save the current values.
      * The real time values with respect to the company code should be displayed from the various share sites
      * The page should be refreshed every 30 seconds
      * The alerts should be provided based on the values matched

#### System Requirements

A set of system services and constraints in detail, the system requirements are the more detailed specification of the user requirements it some times serves as a contract between the user and the developer

#### SOFTWARE REQUIREMENTS

* Microsoft Python framework 2.0
* Anaconda IDE
* Python 3.7
* Jupyter Notebook

#### HARDWARE REQUIREMENTS

* Processor : Pentium IV
* Monitor : SVGA
* RAM : 128MB(minimum)
* Speed : 500MHZ
* Secondary Device : 10GB

#### FUNCTIONAL REQUIREMENTS

These are the statements of services the system should provide, how the system should react for a particular inputs and how the system should behave in the particular situations.

#### NON-FUNCTIONAL REQUIREMENTS

* These are define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
* Process requirements may also be specified mandating a particular CASE system, programming language or development method
* Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless
* Typically they are:
  + Reliability
  + Security
  + Availability
  + Performance.

## CHAPTER 4

* 1. **Existing System**

**SYSTEM ANALYSIS**

Very few works on human emotion detection have so far been reported in the current literature on machine intelligence.Some of the researchers have proposed the following schemes,but they have not yet been implemented.Researchers such as Ekman and Friesen proposed a scheme for the recognition of facial Eye Movements from the movements of cheek, chin, and wrinkles. They observed the movement of facial muscles as shown in figure 3.1 to recognize emotions.

#### Figure 4.1: Emotion Recognition from chin,cheek and wrinkles

Yamada proposed a new method of recognizing emotions through the classification of visual information. Cohen considered temporal variations in facial Eye Movements, which are displayed in live video to recognize emotions. She proposed a new architecture of hidden Markov models to automatically segment and recognize facial Eye Movement stresss[7].

* + 1. **Limitations of Existing System**

Currently available human–computer interfaces do not take complete advantage of the valuable communicative media and thus are unable to provide the full benefits of natural interaction to the users.Human–computer interactions could significantly be improved if computers could recognize the emotion of the users from their facial Eye Movement stresss.The existing systems does not have a good classification accuracy. The exact emotion was not detected.There is no system to help people suffering with neurodevelopment disorder as shown in Figure 3.2. Children with the neurodevelopmental disorder known as [Autism](http://en.wikipedia.org/wiki/Autism) often have difficulty with social

interaction, in part due to an impaired ability to intuit the emotional state of other people.



#### Figure 4.2: People suffering from Autism Disorder

* 1. **Proposed system**

The Proposed System provides an alternative scheme for human emotion recognition from facial images, and its control, using fuzzy logic. Convolutional neural network (CONVOLUTIONAL NEURAL NETWORK) clustering is used for the segmentation of the facial images into three important regions containing mouth, eyes, and eyebrows. The exact emotion is extracted from fuzzified emotions by a denormalization procedure similar to defuzzification. The proposed scheme is both robust and insensitive to noise because of the nonlinear mapping of image attributes to emotions in the fuzzy domain. Experimental results show that the detection accuracies of emotions for adult male, adult female, and children of 8–12 years are as high as 88%, 92%, and 96%, respectively, outperforming the percentage accuracies of the existing techniques.

### Advantages of Proposed System

Emotion recognition and control can be applied in system design for two different problem domains. First, it can serve as an intelligent layer in the next generation human– machine interactive system. Such a system would have extensive applications in the frontier technology of pervasive and ubiquitous computing. Second, the emotion monitoring and control scheme would be useful for psychological counseling and therapeutic applications.

### Applications of the Proposed System

* + - * Proposed system helps people suffering from Autism.These people can’t understand the emotions of surrounding people and others can’t understand their emotions.Thus this system helps in physiological counseling and therapy.
      * It helps in the detection of criminal and antisocial motives.Here by looking at the criminal faces,we can find out whether the criminal has done the crime for gaining money or for fame.

## Feasibility Study

In the feasibility study, we studied various feasibility studies performed i.e technical feasibility whether existing equipment, software were sufficient for completing the project. The economic feasibility determines whether the doing of project is economically beneficial. This seems to be beneficial because the company need not spend any amount on the project. Trainees because they work at a less amount and only machine time are burden.The outcome of first phase was that the request and the various studies were approved and it was decided that the project taken up will serve the end user. On developing and implementation this software saves a lot of amount and Sharing of valuable company time.

The key considerations involved in the feasibility analysis are

* Economical feasibility
* Technical feasibility
* Social feasibility

#### Economical feasibility

This study is carried out to check the economic impact that the system will have on the organization.th e amount of fund that the company can pour into research and development of the system is limited . the expenditure must be justified.

#### Technical feasibility

This is carried out to check the technical feasibility ,that is,the technical requirements of the system.any system developed must not have a high demand on the available technical resources.this will lead to high demands on the available technical resourses.the developed system must have a modest requirements. And are required for implementing this system.

#### Social feasibility

The aspect of study is to check the level of acceptance of the system by the user.This includes the process of training the user to use the system efficiently.The user must not be threatned by the system.His level of confidence must be increased so that he is able to make some constructive criticism which is welcomed[9].

## CHAPTER 5

**DESIGN**

A software design is description of the structure of the software to be implemented, the data which is part of the system, the interfaces between the system components and sometimes the algorithms used. Designers do not arrive at a finished design immediately but develop the design iteratively through a number of different versions. The design process involves adding formality and detail as the design is developed with constant backtracking to correct earlier designs.

## Design process

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase.

Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified.

Design will explain software components in detail. This will help the implementation of the system. Moreover, this will guide the further changes in the system to satisfy the future requirements.

### Form design

Form is a tool with a message; it is the physical carrier of data or information.

The user interface form provides a user to select a workgroup, find the active peers, type any message to send to an active peer.

### Input design

Inaccurate input data is the most common case of errors in data processing. Errors entered by data entry operators can control by input design. Input design is the process of converting user-originated inputs to a computer-based format. Input data are collected and organized into group of similar data.

The specific design process activities are:

* + - * **Architectural design:** The sub-system making up the system and their relationships are identified and documented.
      * **Object oriented design:** In Object oriented design we thought of “things” instead of operations and functions, the executing system is made up of interacting objects that maintain their local state and operation.
      * **Real time software design:** One way of looking at a real time system is as a stimulus response system. Given a particular input stimulus, the system must produce some corresponding response.
      * **User interface Design:** Good user interface design is critical to the success of the system, an interface that is difficult to use will, a best, result in a high level of user errors.

## Modules

1. Face Detection from input image.
2. Segmentation & Determination of the Mouth Region.
3. Segmentation & Determination of the Eye Region.
4. Emotion Detection.

### Face Detection From Input Image

For face detection, first we convert binary image from RGB image. For converting binary image, we calculate the average value of RGB for each pixel and if the average value is below than 110, we replace it by black pixel and otherwise we replace it by white pixel. By this method, we get a binary image from RGB image as shown in Figure 5.1.

Then, we try to find the forehead from the binary image. We start scan from the middle of the image, then want to find a continuous white pixels after a continuous black pixel. Then we want to find the maximum width of the white pixel by searching vertical both left and right site. Then, if the new width is smaller half of the previous maximum width, then we break the scan because if we reach the eyebrow then this situation will arise. Then we cut the face from the starting position of the forehead and its high will be

1.5 multiply of its width as shown in Figure 5.2.



#### Figure 5.1:converting RGB image to Binary image



**Figure 5.2: Face calculation**



**Figure 5.3: finding the middle position of face**

* + 1. **Segmentation & Determination of the Mouth Region**

This module is used the mouth region, we first represent the image in the L \* a \* b space from its conventional red–green–blue (RGB) space. The L \* a \* b system has the additional benefit of representing a perceptually uniform color space. It defines a uniform matrix space representation of color so that a perceptual color difference is represented by the Euclidean distance. The color information, however, is not adequate to identify the lip region. The position information of pixels together with their color would be a good feature to segment the lip region from the face. The Fuzzy C-means clustering algorithm that we employ to detect the lip region is supplied with both color and pixel-position

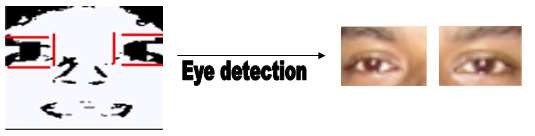
information of the image. This module use in image segmentation in general and lip region segmentation in particular is a novel area of research.

Determination of MO in a black and white image is easier because of the presence of the white teeth. A plot of the average intensity profile against the MO reveals that the curve has several minima, out of which the first and third correspond to the inner region of the top lip and the inner region of the bottom lip, respectively. The difference between the preceding two measurements along the Y-axis gives a measure of the MO[6].

### Segmentation & Determination of the Eye Region and Eyebrows

The eye region in a monochrome image has a sharp contrast to the rest of the face. Consequently, the thresholding method can be employed to segment the eye region from the image. Images grabbed at poor illumination conditions have a very low average intensity value. Segmentation of the eye region in these cases is difficult because of the presence of dark eyebrows in the neighborhood of the eye region. To overcome this problem, we consider images grabbed under good illuminating conditions.

After segmentation of the image, we need to localize the left and right eyes on the image. For eyes detection, we convert the RGB face to the binary face. Now, we consider the face width by W. We scan from the W/4 to (W-W/4) to find the middle position of the two eyes as shown in figure 5.3. The highest white continuous pixel along the height between the ranges is the middle position of the two eyes



#### Figure 5.4: Segmentation of Eye region

Then we find the starting high or upper position of the two eyebrows by searching vertical. For left eye, we search w/8 to mid and for right eye we search mid to w – w/8. Here w is the width of the image and mid is the middle position of the two eyes. There may be some white pixels between the eyebrow and the eye. To make the eyebrow and eye connected as shown in figure 5.4, we place some continuous black pixels vertically from eyebrow to the eye. For left eye, the vertical black pixel-lines are placed in between

mid/2 to mid/4 and for right eye the lines are in between mid+(w-mid)/ 4 to mid+3\*(w- mid)/ 4 and height of the black pixel-lines are from the eyebrow starting height to (h- eyebrow starting position)/4. Here w is the width of the image and mid is the middle position of the two eyes and h is the height of the image. Then we find the lower position of the two eyes by searching black pixel vertically. For left eye, we search from the mid/4 to mid - mid/4 width. And for right eye, we search mid + (w-mid)/ 4 to mid+3\*(w- mid)/ 4 width from image lower end to starting position of the eyebrow. Then we find the right side of the left eye by searching black pixel horizontally from the mid position to the starting position of black pixels in between the upper position and lower position of the eye br. And left side for right eye we search mid to the starting position of black pixels in between the upper position and lower position of right eye. The left side of the left eye is the starting width of the image and the right side of the right eye is the ending width of the image. Then we cut the upper position, lower position, left side and the right side of the two eyes from the RGB image.

In a facial image, eyebrows are the second darkest region after the hair region.

The eye regions are also segmented by thresholding[10].

### Emotion Detection

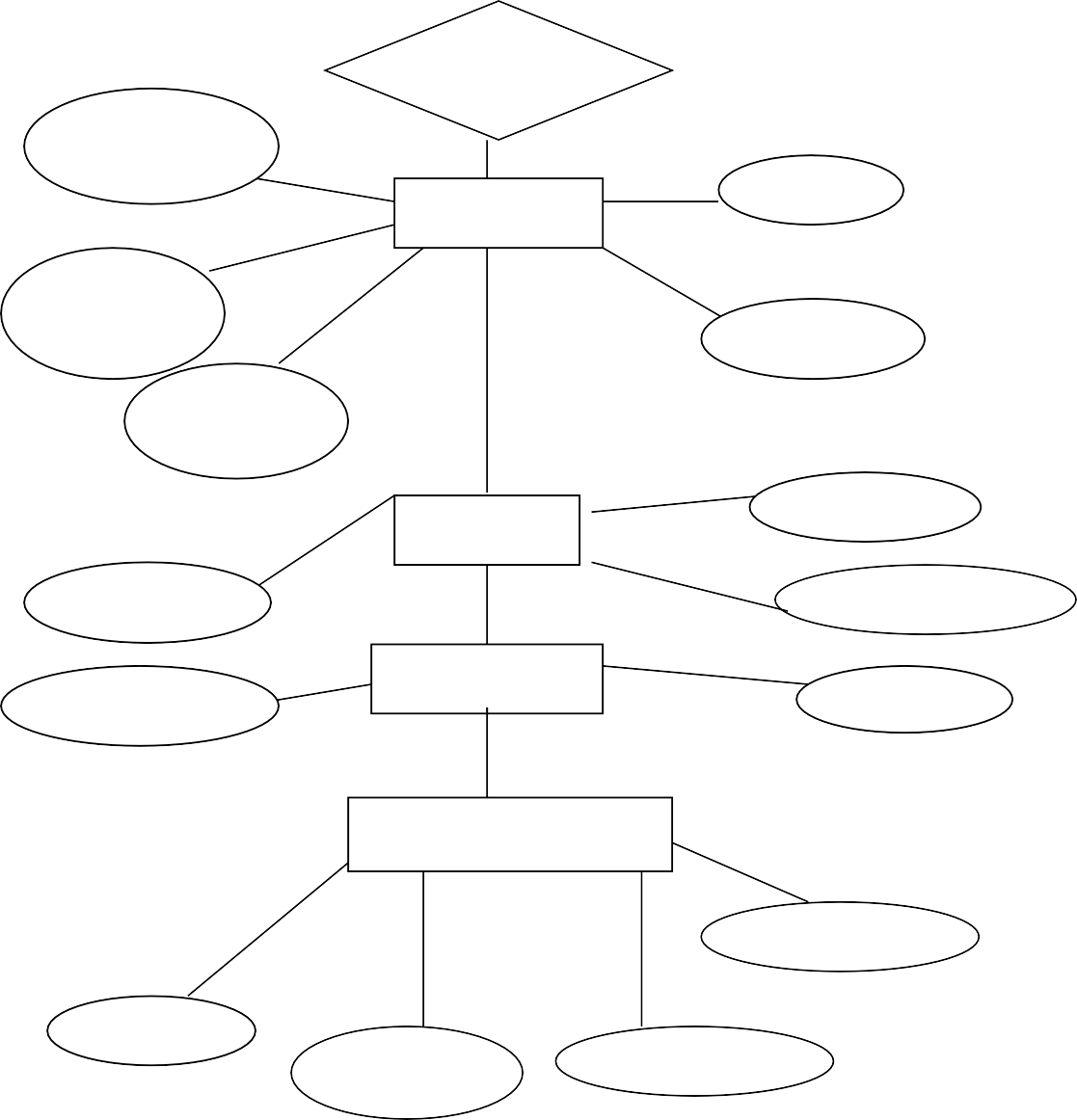
For emotion detection of an image, we have to find the Bezier curve of the lip, left eye and right eye. Then we convert each width of the Bezier curve to 100 and height according to its width. If the person’s emotion information is available in the database, then the program will match which emotion’s height is nearest the current height and the program will give the nearest emotion as output.

If the person’s emotion information is not available in the database, then the program calculates the average height for each emotion in the database for all people and then get a decision according to the average height.

## Architecture Diagram



Main Page



Contains

Input Image

**File**

Exit

Restore

Image

Save

Camera Image

Pre-Process

Next

Skin Color

Connected

Binary Image

Segmentation

Face

Segmentation Process

Result

Mouth

Eye

EyeBrow

Contrast

Preview

Sharpen Image

Brightness

Options

#### Figure 5.5: Architecture diagram

Help

* 1. **Data Flow Diagram and Use case Diagram**

Main Application

start

Launch

Login Page

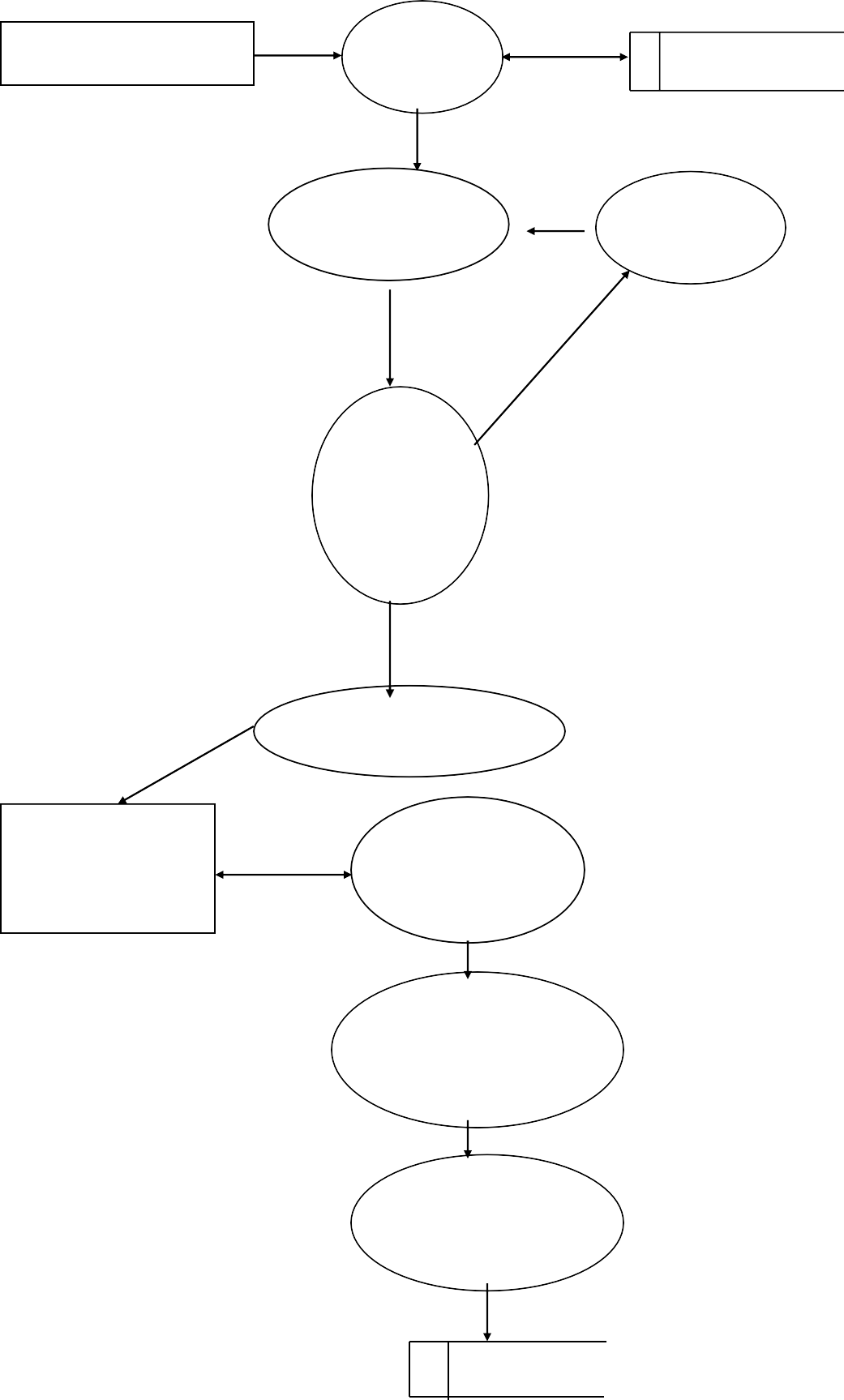
Login Fail

Check User name& Password

Login Successfully

After Successfully Login in Main Page

Home Page

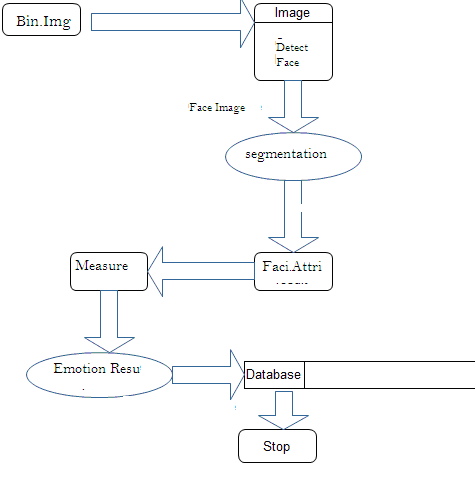


Select Images (.jpeg,.bng,.tiff etc)

Convert to Binary Image

Return Binary Images

#### Figure 5.6: DFD of Login module



**Figure 5.7: DFD of segmentation and emotion result**



**Registration**

**UserLogin**

**User**

**Select Image**

**Pre Process**

**Segmentation**

#### Figure 5.8: Use case Diagram

* 1. **Class Diagram and ER diagram:**

User:

)

+result

+ color\_ segmentation

ion

+emotion ()

+Calculate Distance()

+compare Image()

+bezier\_posit

+displayResult()

+Databsae image

+Connection

**Pre Process**

**Emotion Result**

+ eyelocal ()

+ range ()

+ black\_white (

+Upload Image()

+Image

+Open File dialog

**Segmentation**

**Select Image**

+Login()

+UserName

+Password

**User Login**

+registation()

+Login()

+Registration()

+Detail

+details

**User**

**Registation**

#### Figure 5.9: class diagram

|  |  |  |
| --- | --- | --- |
| **Position** | | |
|  | Id |  |
|  | lip1\_x |  |
|  | lip1\_y |
|  | lip2\_x |
|  | lip2\_y |
|  | lip3\_x |
|  | lip3\_y |
|  | lip4\_x |
|  | lip4\_y |
|  | lip5\_x |
|  | lip5\_y |
|  | lip6\_x |
|  | lip6\_y |
|  | left\_eye1\_x |
|  | left\_eye1\_y |
|  | left\_eye2\_x |
|  | left\_eye2\_y |
|  | left\_eye3\_x |
|  | left\_eye3\_y |
|  | left\_eye4\_x |
|  | left\_eye4\_y |
|  | left\_eye5\_x |
|  | left\_eye5\_y |
|  | left\_eye6\_x |
|  | left\_eye6\_y |
|  | right\_eye1\_x |
|  | right\_eye1\_y |
|  | right\_eye2\_x |
|  | right\_eye2\_y |
|  | right\_eye3\_x |
|  | right\_eye3\_y |
|  | right\_eye4\_x |
|  | right\_eye4\_y |
|  | right\_eye5\_x |
|  | right\_eye5\_y |
|  | right\_eye6\_x |
|  | right\_eye6\_y |
|  | lip\_h1 |
|  | lip\_h2 |
|  | lip\_h3 |
|  | lip\_h4 |
|  | left\_eye\_h1 |
|  | left\_eye\_h2 |
|  | left\_eye\_h3 |
|  | left\_eye\_h4 |
|  | right\_eye\_h1 |
|  | right\_eye\_h2 |
|  | right\_eye\_h3 |  |
|  | right\_eye\_h4 |

**Figure 5.10 : ER diagram**

Sad

Surprise

Normal

Smile

Name

**Person**

**TB\_SourceUser**

RecId UserName Pwd



**CHAPTER 6**

**IMPLEMENTATION**

Implementation is the realization of an application, or execution of a [plan,](http://en.wikipedia.org/wiki/Plan) idea, [model,](http://en.wikipedia.org/wiki/Scientific_modelling) [design,](http://en.wikipedia.org/wiki/Design) [specification,](http://en.wikipedia.org/wiki/Specification) [standard,](http://en.wikipedia.org/wiki/Standardization) [algorithm,](http://en.wikipedia.org/wiki/Algorithm) or [policy.](http://en.wikipedia.org/wiki/Policy)

There are five things in consideration when the project is developed. They are as follows:-

* + - Correction
    - Adaptation
    - Maintenance
    - Change

#### Correction:

The project is corrective to its end and all the validation has been incorporated to software developed so that no further corrective action can be thought of.

#### Adaptation/Enhancement:

In this Project a high performance data synchronization server for mobile device is proposed. For the mobile application system, the information or data (ex. Contacts, Music, Video, Image) sets are usually stored in both the mobile device and system database. After several operations for the mobile system, the data sets between the mobile device and system database may become not identical. In order to keep the consistence of these data sets, the data synchronization plays a key role in such mobile applications **Maintenance:**

The project is to be maintained in the way its accuracy, versatility, working, integrity, corrective ness, etc. are as was proposed and will be as it was made with possibility of enhancement to these properties. This project also has this property that makes it truly maintainable.

#### Change:

Design during maintenance involves redesigning the product to incorporate the desired changes. The changes must then be implemented, nternal documentation of the code must be updated, and new test cases must be designed to access the adequacy of the modification. Also the supporting documents must be updated to reflect the changes.

The modules were implemented as follows:

## Face Detection module

For converting binary image, we calculate the average value of RGB for each pixel and if the average value is below than 110, we replace it by black pixel and otherwise we replace it by white pixel.The code for this is:

public Bitmap black\_and\_white(Image Im)

## CHAPTER 7

**TESTING AND RESULT ANALYSIS**

The purpose of testing is to discover Errors. Testing is the process of trying to discover every conceivable fault or weakness in work product, IT provides a way to check the functionality of components ,sub assemblies ,assemblies and/or finished product. it is the process of exercising software with the intent of ensuring that the software meets its requirements and user expectations and does not fail in an unacceptable manner ..there are various types of test. each test type addresses a specific resting requirement.

## Unit Testing

In this, the programs that made up the system were tested. This is also called as program testing. This level of testing focuses on the modules, independently of one another. The purpose of unit testing is to determine the correct working of the individual modules. For unit testing, we first adopted the code testing strategy, which examined the logic of program. During the development process itself all the syntax errors etc. got rooted out. For this we developed test case that result in executing every instruction in the program or module i.e. every path through program was tested. (Test cases are data chosen at random to check every possible branch after all the loops.).

Unit testing involves a precise definition of test cases, testing criteria, and management of test cases. This level of testing focuses on the modules, independently of one another. Testing means to check whether system meets user requirements about:

### Unit test for face module:

The unit testing for face module is done after the completion of face module. The face module was designed and tested to see if there is any error. Here whether the face region is segmented correctly or not was checked.

**Test Results:** The test cases mentioned above passed successfully. No defect was encountered.

### Unit test for mouth module:

The unit testing for mouth module is done after the completion of mouth module. The mouth module was designed and tested to see if there is any error. Here whether the mouth region is segmented correctly or not was checked.

**Test result:** The test cases mentioned above passed successfully. No defect was encountered.

### Unit test for eye module:

The unit testing for eye module is done after the completion of eye module. The eye module was designed and tested to see if there is any error. Here whether the left eye and right eye region is segmented correctly or not was checked.

**Test result:** The test cases mentioned above passed successfully. No defect was encountered.

### Unit test for emotion detection module:

The unit testing for emotion detection module is done after the completion of emotion module. The emotion module was designed and tested to see if there is any error. Here whether the exact emotion is detected correctly or not was checked.

**Test result:** The test cases mentioned above passed successfully. No defect was encountered.

## Integration Testing

In this the different modules of a system are integrated using an integration plan. The integration plan specifies the steps and the order in which modules are combined to realize the full system. After each integration step, the partially integrated system is tested. The primary objective of integration testing is to test the module interface.

In Main module, all the individual programs are tested first and after having successful results in the individual program testing we moved further for the integration.

We have combined some programs and then tested it, after having good results; we have combined all the programs together and started for system testing.

**Test result:** The test cases mentioned above passed successfully. No defect was encountered

## System Testing

Once we are satisfied that all the modules work well in themselves and there are no problems, we do in to how the system will work or perform once all the modules are put together. At this stage the system is used experimentally to ensure that all the requirements of the user are fulfilled. At this point of the testing takes place at different levels so as to ensure that the system is free from failure.

The training is given to user about how to make an entry. The best test made on the system was whether it produces the correct outputs. All the outputs were checked out and were found to be correct. Feedback sessions were conducted and the suggested changes given by the user were made before the acceptance test. System tests are designed to validate a fully developed system with a view to assuring that it meets its requirements.

## User Acceptance Testing

Acceptance testing involves planning and execution of functional test, performance tests. This is critical phase of any project and requires significant contribution by end user.

**Test result:** All the test cases passed successfully. No defect was encountered.

|  |  |  |  |
| --- | --- | --- | --- |
| CASE | INPUT | EXPECTED  OUTPUT | RESULT |
| 1 | When a person with Spectacles was  given as input  image. | Correct emotion can’t be detected. | Success |
| 2 | When input image  was animals | It is not a human  face | Success |
| 3 | When invalid password or user-id  was entered | Warning message to be displayed | Success |

## CHAPTER 8

**CONCLUSION AND FUTURE ENHANCEMENTS**

An important aspect of this Project is the design of an emotion control scheme. The accuracy of the control scheme ensures convergence of the control algorithm with a zero error, and repeatability ensures the right selection of audiovisual stimulus. The proposed scheme of emotion recognition and control can be applied in system design for two different problem domains. First, it can serve as an intelligent layer in the nextgeneration human–machine interactive system. Such a system would have extensive applications in the frontier technology of pervasive and ubiquitous computing. Second, the emotion monitoring and control scheme would be useful for psychological counseling and therapeutic applications. The pioneering works on the “structure of emotion” by Gordon and the “emotional control of cognition” by Simon would find a new direction with the proposed automation for emotion recognition and control.

In the course of work, we have identified areas that we need to carry out the further work of the project.

* Our proposed system can be enhanced to be used in next generation human machine interactive system.
* We can use a web camera to capture images of people and detect their emotions.
* It is enhanced to be used in medicine field for physiological counseling.
* Used in emotion recognition of animals.
* Emotions such as anger, disgust etc can be detected in the future.

## CHAPTER 9

**SNAPSHOTS**

# REFERENCES

1. J. C. Bezdek, “Fuzzy mathematics in pattern classification,” Ph.D. dissertation,Appl. Math. Center, Cornell Univ., Ithaca, NY, 1973.
2. B. Biswas, A. K. Mukherjee, and A. Konar, “Matching of digital images using fuzzy logic,” *AMSE Publication*, vol. 35, no. 2, pp. 7–11, 1995.
3. M. T. Black and Y. Yacoob, “Recognizing facial Eye Movements in image sequences using local parameterized models of image motion,” *Int. J.Comput. Vis.*, vol. 25, no. 1, pp. 23–48, Oct. 1997.
4. C. Busso and S. Narayanan, “Interaction between speech and facial gestures in emotional utterances: A single subject study,” *IEEE Trans. Audio,Speech Language Process.*, vol. 15, no. 8, pp. 2331–2347, Nov. 2007.
5. I. Cohen, “Facial Eye Movement stress recognition from video sequences,” M.S.thesis, Univ. Illinois Urbana-Champaign, Dept. Elect. Eng., Urbana, IL,2000.
6. I. Cohen, N. Sebe, A. Garg, L. S. Chen, and T. S. Huang, “Facial Eye Movement stress recognition from video sequences: Temporal and static modeling,”*Comput. Vis. Image Underst.*, vol. 91, no. 1/2, pp. 160–187,Jul. 2003.
7. C. Conati, “Probabilistic assessment of user’s emotions in educational games,” *J. Appl. Artif. Intell., Special Issue Merging Cognition AffectHCT*, vol. 16, no. 7/8, pp. 555– 575, Aug. 2002.
8. G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski,“Classifying facial actions,” *IEEE Trans. Pattern Anal. Mach. Intell.*,vol. 21, no. 10, pp. 974–989, Oct. 1999.
9. P. Ekman and W. V. Friesen, *Unmasking the Face: A Guide to RecognizingEmotions From Facial Clues*. Englewood Cliffs, NJ: Prentice-Hall,1975.
10. I. A. Essa and A. P. Pentland, “Coding, analysis, interpretation and recognition of facial Eye Movement stresss,” *IEEE Trans. Pattern Anal. Mach. Intell.*,vol. 19, no. 7, pp. 757–763, Jul. 1997.
11. W. A. Fellenz, J. G. Taylor, R. Cowie, E. Douglas-Cowie, F. Piat,S. Kollias, C. Orovas, and B. Apolloni, “On emotion recognition of faces and of speech using neural networks, fuzzy logic and the ASSESS systems,”in *Proc. IEEE -INNS-ENNS Int. Joint Conf. Neural Netw.*, 2000,pp. 93–98.
12. J. M. Fernandez-Dols, H. Wa**l** botl, and F. Sanchez, “Emotion category accessibility and the decoding of emotion from facial Eye Movement and context,” *J. Nonverbal Behav.*, vol. 15, no. 2, pp. 107–123,Jun. 1991
13. [www.msdn.microsoft.com](http://www.msdn.microsoft.com/)
14. [www.pentestmonkey.com](http://www.pentestmonkey.com/)
15. [www.testbed.com](http://www.testbed.com/)
16. [www.michaeldaw.org](http://www.michaeldaw.org/)
17. [www.webappsec.com](http://www.webappsec.com/)