DETECTION OF PSYCHOLOGICAL ABNORMALITIES AND CRIMINAL BEHAVIOUR INTEGRATING FACIAL EMOTION WITH MACHINE LEARNING TECHNIQUE

A PROJECT REPORT

Submitted by

K.GAYATHIRI (312416104037)
ISWARIYA RAMU (312416104037)

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

St. JOSEPH'S INSTITUTE OF TECHNOLOGY

CHENNAI -119

ANNA UNIVERSITY: CHENNAI - 600025

MARCH -2020

ANNA UNIVERSITY : CHENNAI 600 025

PROGRESS THROUGH KNOWLEDGE

BONAFIDE CERTIFICATE

Certified that this project report "**DETECTION OF PSYCHOLOGICAL ABNORMALITIES AND CRIMINALBEHAVIOURINTEGRATING FACIAL EMOTION WITHMACHINE LEARNING TECHNIQUE"** is the bonafide work of "**GAYATHIRI.K(312416104037)" and "ISWARIYA RAMU(312416104044)"** who carried out the project work under my supervision.

SIGNATURE

Dr.J DAFNI ROSE ME, Ph.D,

HEAD OF THE DEPARTMENT

Computer Science and
Engineering
St.Joseph's Institute of
technology,
Old Mamallapuram Road,
Chennai- 600 019.

SIGNATURE

MR.C.GOWTHAM ME,

ASSISTANT PROFESSOR

Computer Science and
Engineering
St.Joseph's Institute of
technology,
Old Mamallapuram Road,
Chennai- 600 019.

ABSTRACT

The human face is an important part of an individual's body and plays an important role in knowing the individual's mood. The face is where a human expresses all his basic emotions.

In existing system, we examine the criminal mental disorders manually by assessing them but which has many disadvantages like we cannot predict any accurate solutions based on the assessment score because we might be not sure what kind of emotions the human user would be all time. To overcome this problem and suggest an effective solution for Criminal rehabilitation, we propose a hybrid architecture invoking facial based emotion sequence, PEN test, IQ test. By consistent monitoring of a human's emotion and subjecting to PEN and IQ test, the human's mental state is routed. Combination of above three techniques provides promising results for Criminal rehabilitation and self-control.The processing of emotional facial expressions is modulated by personality; for example, neuroticism, a dimension of the five-factor model of personality has been found to play a role in this regard. Neuroticism is characterized by the tendency to be anxious, nervous, and hostile. Neuroticism is also considered to be a risk predictor for depression.Cognitive and executive functions are often impaired in different clinical disorders, such as dementia, attention deficit hyperactivity disorder, schizophrenia and obsessive compulsive disorder. In schizophrenia, cognitive impairments are often found in multiple areas, including visual information processing; attention; working memory; short-term memory and learning; executive functioning; speed of processing; reasoning and problem solving

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Facial expressions can be considered not only as the most natural form of displaying human emotions but also as a key non-verbal communication technique. If efficient methods can be brought about to automatically recognize these facial expressions, striking improvements can be achieved in the area of human computer interaction. Research in facial emotion recognition has being carried out in hope of attaining these enhancements. In fact, there exist other applications which can benefit from automatic facial emotion recognition. Artificial Intelligence has long relied on the area of facial emotion recognition to gain intelligence on how to model human emotions convincingly in robots. Recent improvements in this area have encouraged the researchers to extend the applicability of facial emotion recognition to areas like chat room avatars and video conferencing avatars. The ability to recognize emotions can be valuable in face recognition applications as well. Suspect detection systems and intelligence improvement systems meant for children with brain development disorders are some other beneficiaries.

The processing of emotional facial expressions is modulated by personality; for example, neuroticism, a dimension of the five-factor model of personality has been found to play a role in this regard. Neuroticism is characterized by the tendency to be anxious, nervous, and hostile. Neuroticism is also considered to be a risk predictor for depression.

Cognitive and executive functions are often impaired in different clinical disorders, such as dementia, attention deficit hyperactivity disorder, schizophrenia and obsessive compulsive disorder. In schizophrenia, cognitive impairments are often found in multiple areas, including visual information processing; attention; working memory; short-term memory and learning; executive functioning; speed of processing; reasoning and problem solving; context processing and social perception and cognition. The impairment of executive functions called "dysexecutive syndrome", are common in neurological patients and are related to brain dysfunction specifically in the prefrontal cortex. Individuals who have an impairment of executive functions show problems of starting and stopping activities, a difficulty in mental and behavioral shifts, an increased distractibility and difficulties in learning new tasks. The executive functions, defined as higher order cognitive functions needed for performing complex tasks, are often impaired also in patients with Obsessive Compulsive Disorder (OCD) which are characterized by the impairment of several skills such as attention, planning, problem-solving and behavioral control. Further, OCD is often associated with impairments of visuospatial skills, and of memory functioning, including visual, verbal, and numerical. Therefore, no single profile of cognitive deficits has been found to characterize all patients; the majority have impaired ability in at least one area of functioning and a standardized platform for assessing neurocognitive functioning is an important aspect of comprehensive treatment and research for this and other conditions.

1.2 DOMAIN-IMAGE PROCESSING

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually **Image Processing** system includes treating images as two dimensional signals while applying already set signal processing methods to them. It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing includes the following steps:

Importing the image with optical scanner or by digital photography.

Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.

Output is the last stage in which result can be altered image or report that is based on image analysis

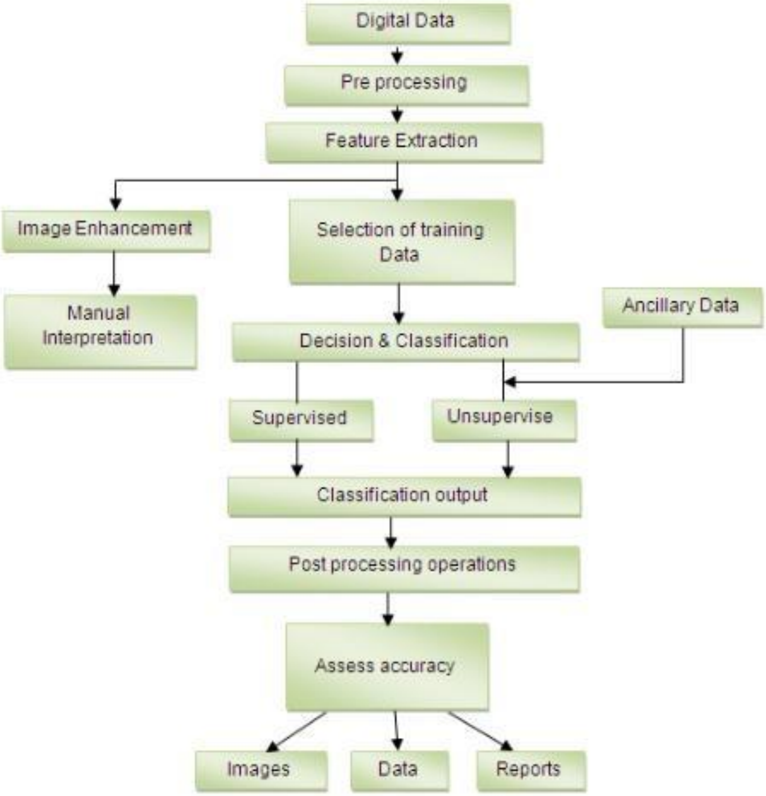
1.2.1 PURPOSE OF IMAGE PROCESSING

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

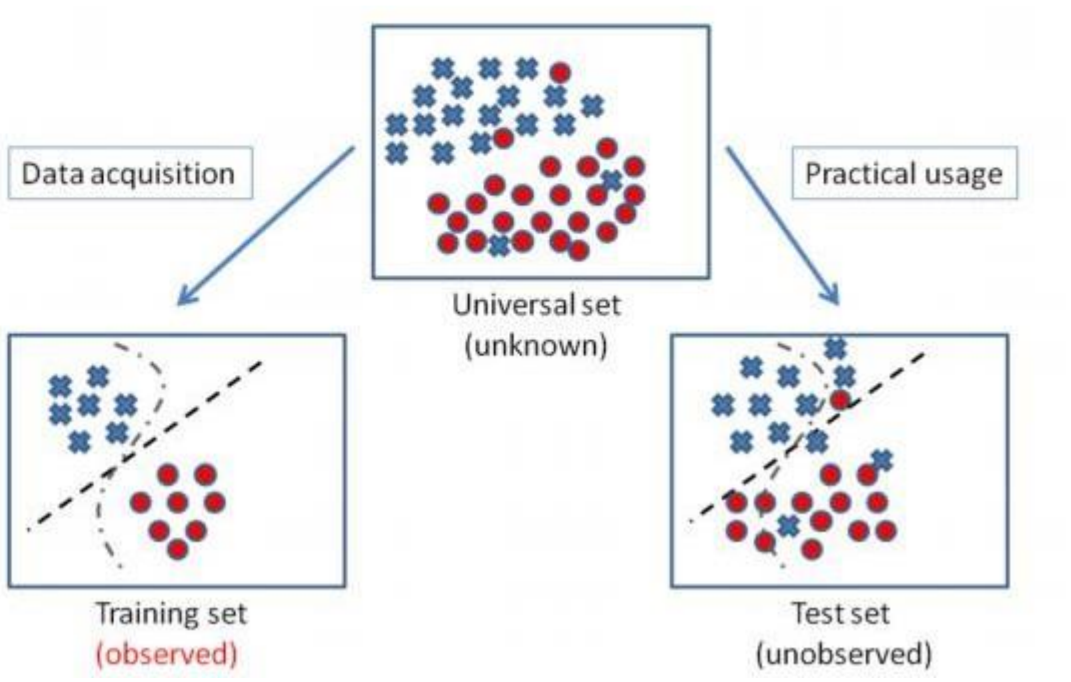
1.2.2 TYPES

The two types of **methods used for Image Processing** are **Analog and Digital** Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.



1.3 MACHINE LEARNING:

“Optimizing a performance criterion using example data and past experience”, said by E. Alpaydin, gives an easy but faithful description about machine learning. In machine learning, data plays an indispensable role, and the learning algorithm is used to discover and learn knowledge or properties from the data. The quality or quantity of the dataset will affect the learning and prediction performance. The textbook (have not been published yet) written by Professor Hsuan-Tien Lin, the machine learning course instructor in National Taiwan University (NTU), is also titled as “Learning from Data”, which emphasizes the importance of data in machine learning. Fig. 1 shows an example of two-class dataset.Training Set and Test Set In machine learning, an unknown universal dataset is assumed to exist, which contains all the possible data pairs as well as their probability distribution of appearance in the real world. While in real applications, what we observed is only a subset of the universal dataset due to the lack of memory or some other unavoidable reasons. This acquired dataset is called the training set (training data) and used to learn the properties and knowledge of the universal dataset. In general, vectors in the training set are assumed independently and identically sampled (i.i.d) from the universal dataset. In machine learning, what we desire is that these learned properties can not only explain the training set, but also be used to predict unseen samples or future events. In order to examine the performance of learning, another dataset may be reserved for testing, called the test set or test data. For example, before final exams, the teacher may give students several questions for practice (training set), and the way he judges the performances of students is to examine them with another problem set (test set).



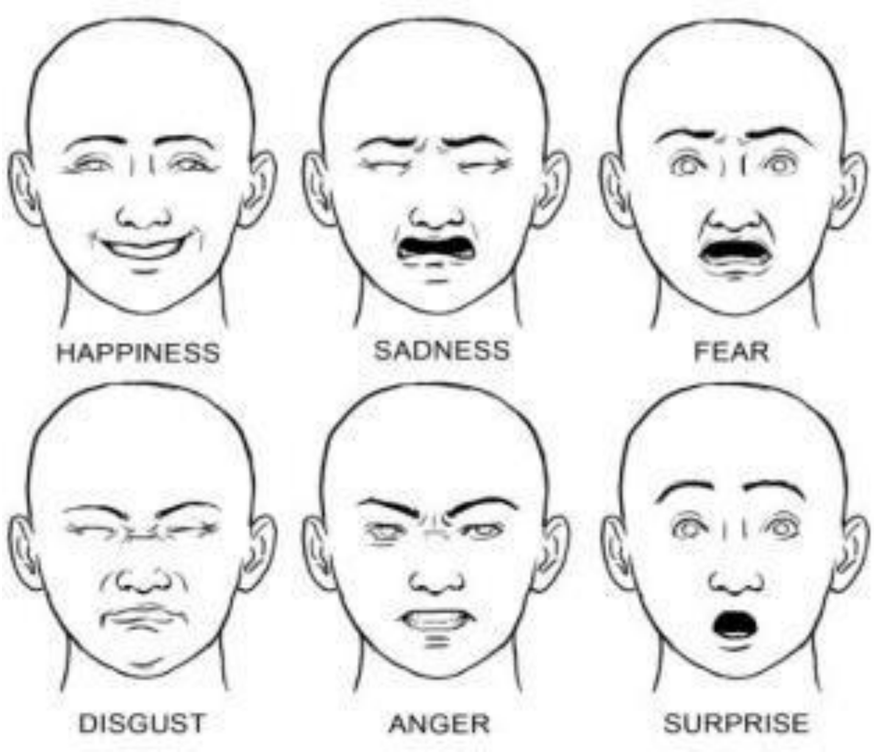
1.3 EXISTING METHOD

Existing systems are highly complex in terms of time and storage for recognizing user behavior. Existing system focus on only manual assessment which takes time and doesn't provide promising results as the user emotions keeps varying based on personal, situational problems. Existing system doesn't invoke biometric based emotion recognition of the users. Existing system doesn't focus on user facial expression with cycometric test based automatic individual criminal behavior prediction.

1.4 PROPOSED SYSTEM

Mental Illness has a profound impact on people's functioning, health and quality of life. Detecting early warnings of depression or any other mental illness is challenging. The proposed system provides a hybrid architecture invoking facial based emotion sequence, PEN test, IQ test. By consistent monitoring of a human's emotion and subjecting to PEN and IQ test, the human's mental state is routed. Combination of above three techniques provides promising results for Criminal rehabilitation and self-control.

In our proposed system, the emotions are continuously monitored based on which the information for the classification of mental illness of the person is obtained. Further, using the information obtained, it conducts a psychology test to diagnose the severity of the mental condition. It combines these outputs with a psychometric study which consists of an IQ test and a Personality test. The output of the combination of these three parameters are classified to determine the probabilities of criminal potential a person can hold.



CHAPTER 2

LITERATURE SURVEY

Olga Mich, Andrzej Burda, Krzysztof Pancerz and Jerzy Gomuła designed the The Knowledge Base for Computer-Aided Diagnosis of Mental Disorders Based on Psychometric Tests (2014). The knowledge base is a key element of many decision support systems. In the paper, we describe, the knowledge base implemented in the Copernicus system - a tool for computer-aided diagnosis of mental disorders based on data coming from psychometric tests, in the current version, coming from the Minnesota Multiphasic Personality Inventory (MMPI) test. This tool uses a variety of classification ways for differential inter profile diagnosis. Therefore, the knowledge base embodied in the tool is of different character. The special attention is focused on new elements added in the last version of the Copernicus system.

Filippo La Paglia, Caterina La Cascia, Pietro Cipresso, Rosalinda Rizzo, Antonio Francomano, Giuseppe Riva, and Daniele La Barbera designed the Psychometric Assessment Using Classic Neuropsychological and Virtual Reality Based Test: A Study in Obsessive-Compulsive Disorder (OCD) and Schizophrenic Patients (2014) .Assessment of neurocognitive functioning is a critical task in clinical settings. In many disorders, cognitive impairment precedes the onset of behavioral symptoms, and cognitive decline is a major factor contributing to functional disability.

Robert D. Gibbons, David J. Weiss, Ellen Frank, and David Kupfer designed the Computerized Adaptive Diagnosis and Testing of Mental Health Disorders (2016) In this review we explore recent developments in computerized adaptive diagnostic screening and computerized adaptive testing for the presence and severity of mental health disorders such as depression, anxiety, and mania. The statistical methodology is unique in that it is based on multidimensional litem response theory (severity) and random forests (diagnosis) instead of traditional mental health measurement based on classical test theory (a simple total score) or unidimensional item response theory. We show that the information contained in large item banks consisting of hundreds of and the information contained in these large item banks can be precisely extracted using adaptive administration of a small set of items for each individual. In terms of diagnosis, computerized adaptive diagnostic screening can accurately track an hour-long face-to-face clinician diagnostic interview for major depressive disorder (as an example) in less than a minute using an average of four questions with unprecedented high sensitivity and specificity.

CHAPTER 3

SYSTEM DESIGN

In system design a general description about the module and various components used along with their corresponding diagrammatic representation using UML are discussed.

3.1 DESIGN REPRESENTATION

The design can be represented by means of UML diagrams. In this, We are going to represent the design by using unified modeling language.

3.1.1 UNIFIED MODELLING LANGUAGE

UML is a standard language for specifying, visualizing and documenting of software system and created by object management group (OMG) in 1997. There are three types of modelling : structural model, behavioural model, and architectural model. To model a system the most important aspect is to capture the dynamic behaviour which has some internal or external factors for making the interactions. These internal or external agents are known actors. It consists of actors, use cases and their relationship among several other useful and important features. The UML diagrams are,

- Use Case Diagram
- Flow diagram
- Activity Diagram
- Sequence diagram
- Collaboration diagram

3.1.2 USE CASE DIAGRAM

A use case illustrates a unit of functionality provided by the system. The main purpose of use case diagram is to help development teams visualize the functional requirements of a system, including the relationship of “actors” to essential processes, as well as the relationship among different use cases.

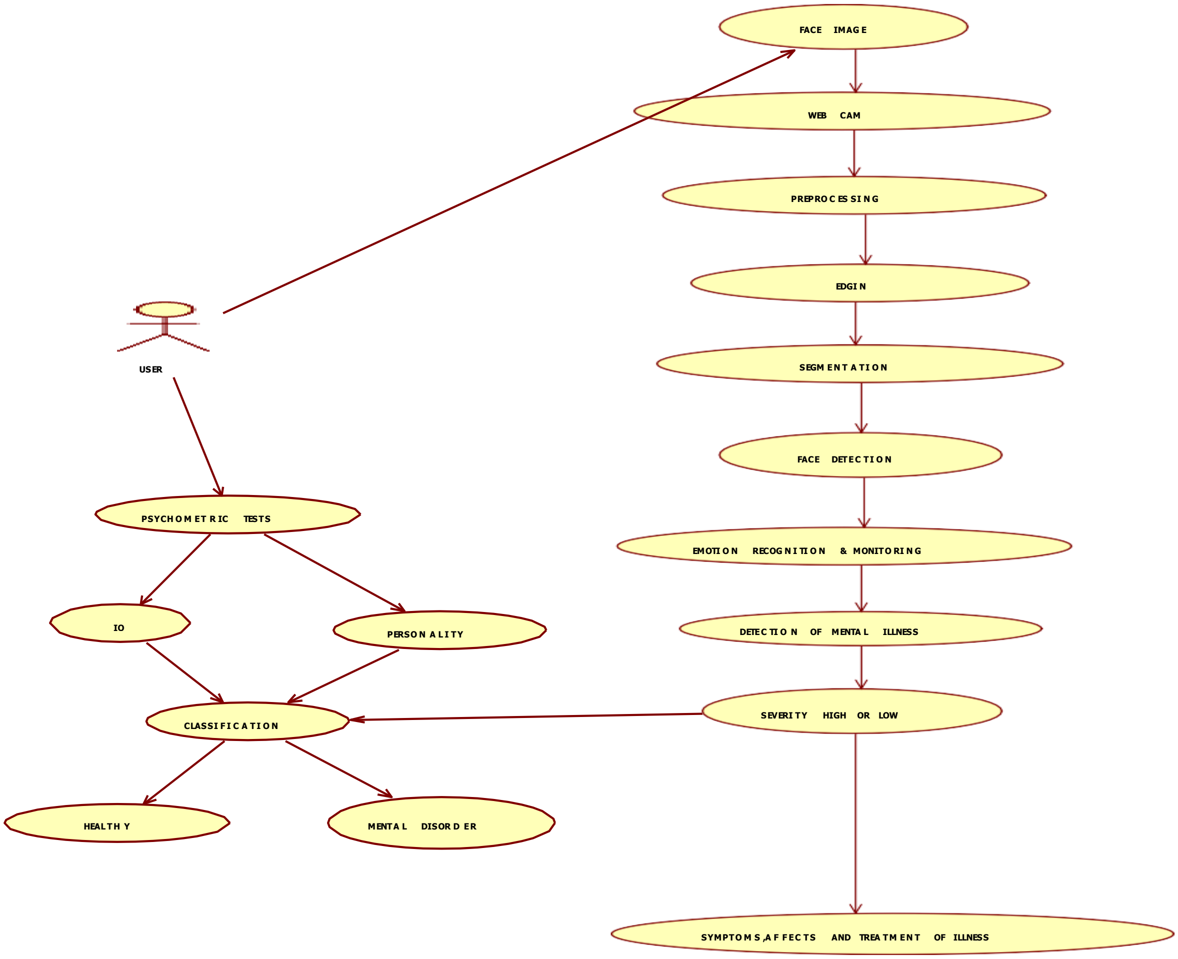


Figure 3.1 Use case diagram for emotion recognition

The user and the system are the two actors used in the figure 3.1,use case diagram .The user will initially undergo the process of image processing and then undergo various kind of test to help get the accurate result. . The user will do the process of capturing the image from the dataset then it will scan the image. The scanned image will get filtered. The filtered image will get enhanced.

3.1.3 FLOW DIAGRAM

Users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented is represented in figure 3.2 Flow diagram is a collective term for a diagram representing a flow or set of dynamic relationships in a system.

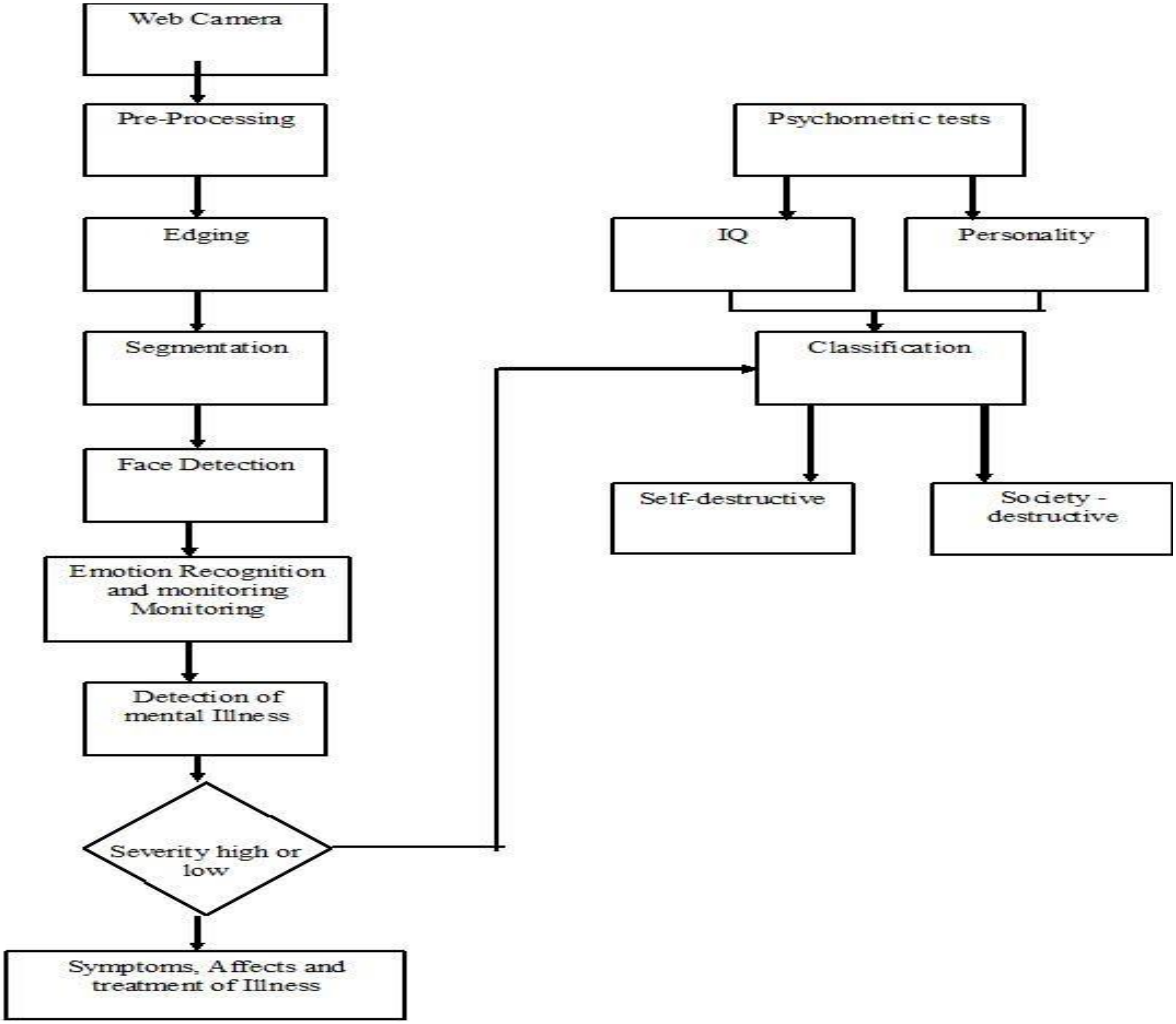


Figure 3.2 Flow diagram for face recognition

The term flow diagram is also used as a synonym for flowchart, and sometimes as a counterpart of the flowchart.

3.1.4 ACTIVITY DIAGRAM

A activity diagram displays the structural relationship of components of a software system.

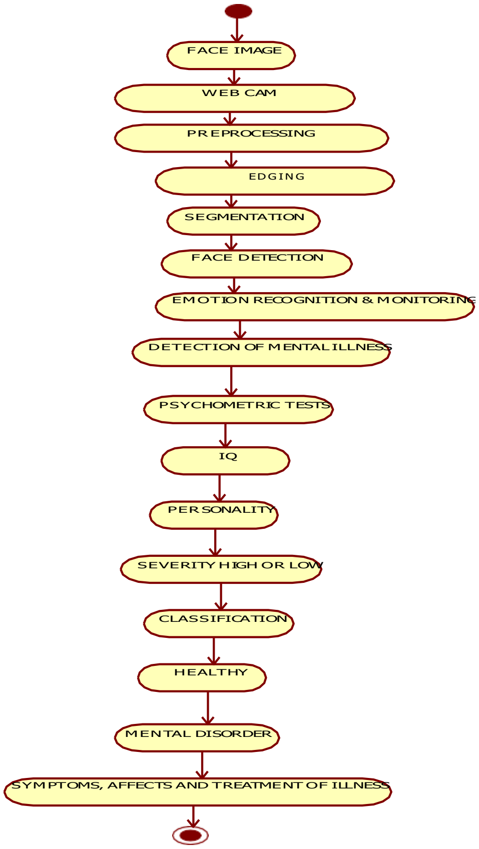


Figure 3.3 diagram for face recognition

The start and the end circles indicate the beginning and ending of the entire process of facical emotion detection.This helps to understand the process flow of the system in detail.

3.1.5 SEQUENCE DIAGRAM

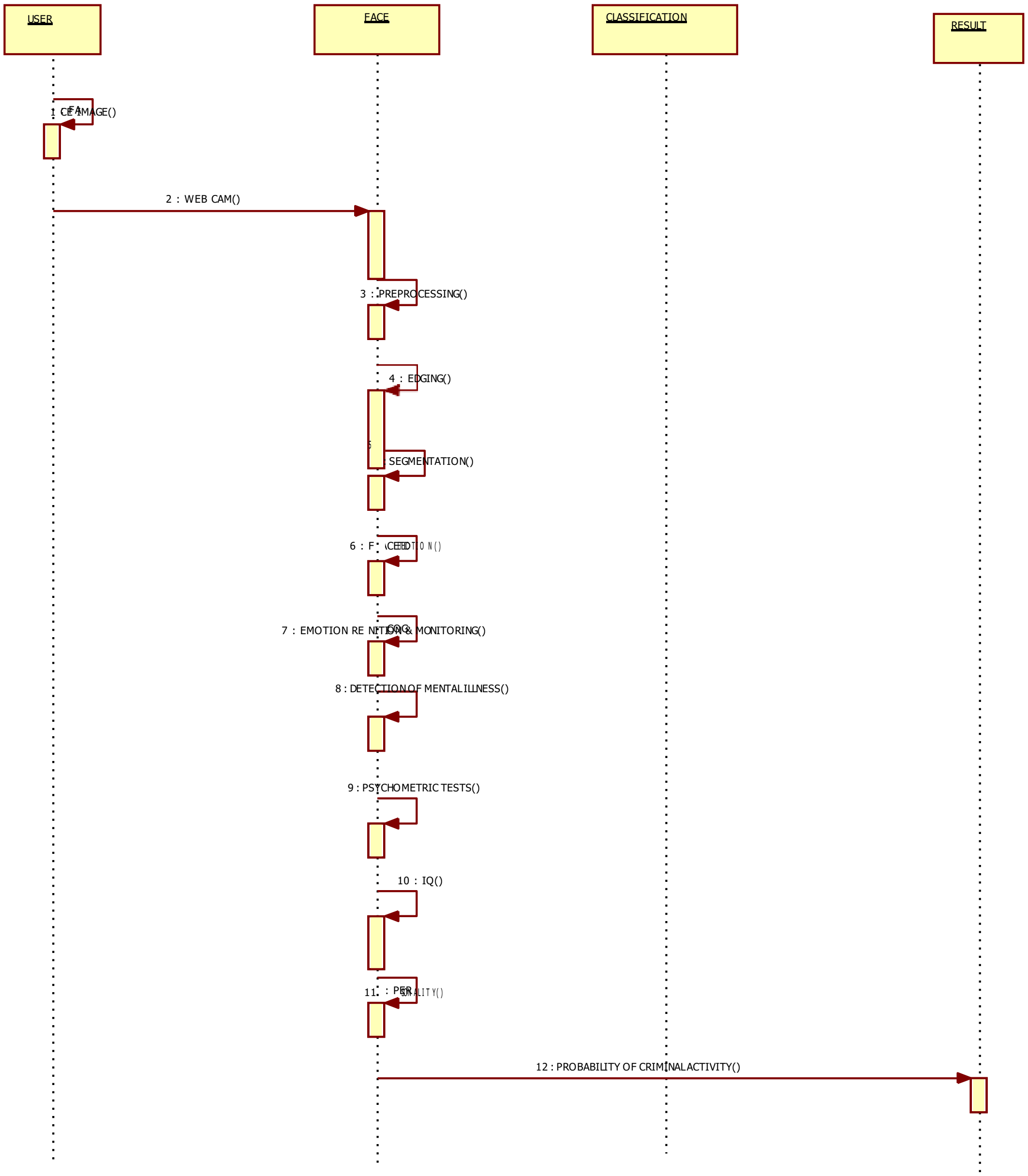


Figure 3.4 Sequence diagram for facial emotion recognition.

The above figure 3.4 is sequence diagram that explain the sequence of activity that is done during the processs of facial emotion recognition.This also helps in understanding the sequence flow.

3.1.6 COLLABORATION DIAGRAMS

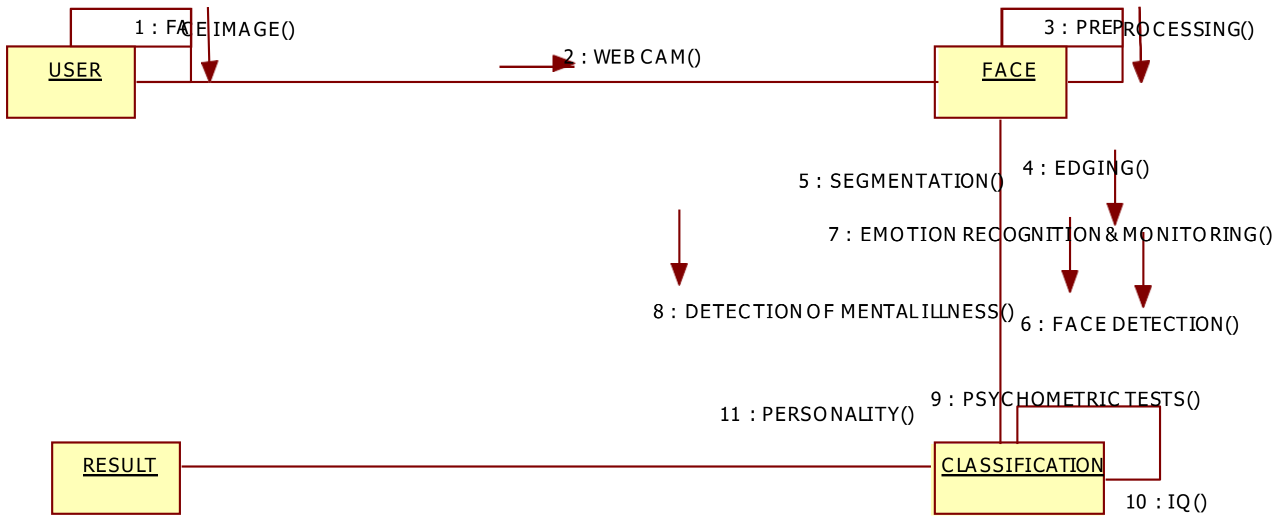


Figure 3.5 Collaboration Diagram for facial emotion recognition.

The above figure 3.5 shows the collaboration diagram for facial emotion recognition.This helps to understand the collaboration diagram for the entire process of the system. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

3.2 EMOTION DATA FLOW DIAGRAMS

LEVEL 0

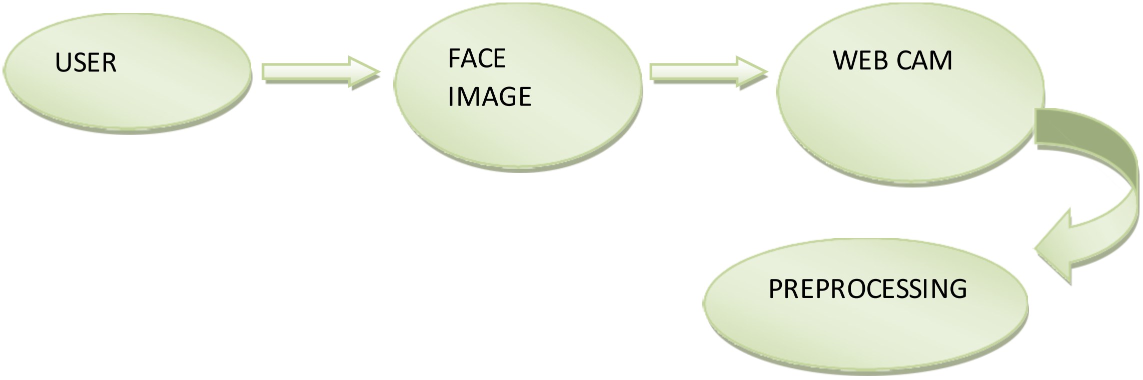


Figure 3.6 Level 0 diagram of emotion data flow diagrams.

This above diagrams shows the level 0 of the data flow diagram , Where the users face is captured and preprocessed during this level.

LEVEL 1

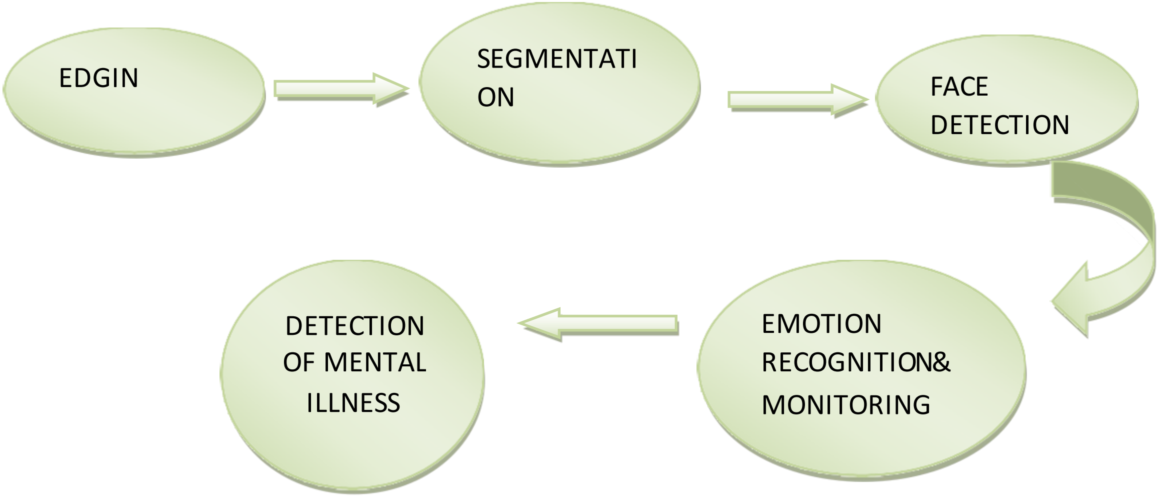


Figure 3.7 Level 1 of emotion data flow diagrams

This above diagram shows the level 1 of the data flow diagram, here the preprocessed image is segmented for face detection process to retrieve the mental illness of the user.

LEVEL 2

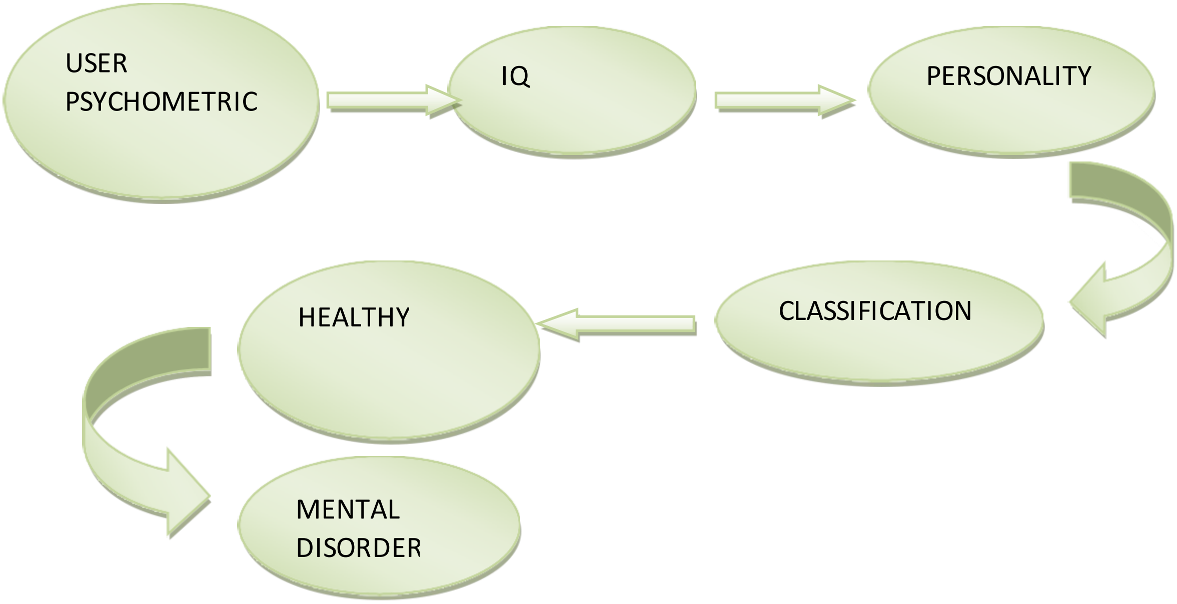


Figure 3.8 level 2 of emotion data flow diagram.

This above diagram shows the level 2 of the data flow diagram ,Where the previous result is combined with the IQ and Personality test to get the accurate result.

LEVEL 3

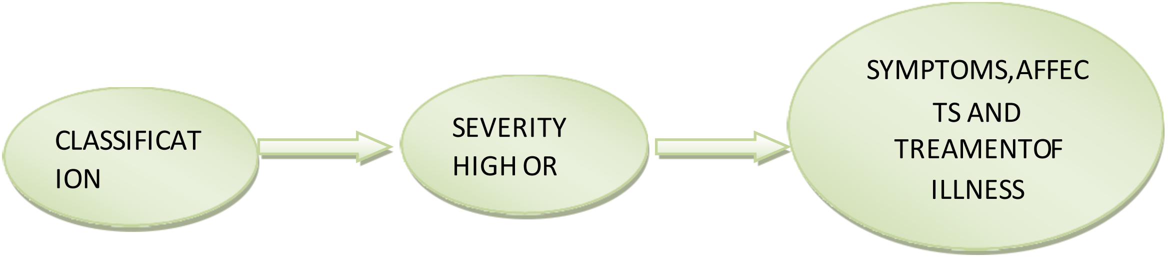


Figure 3.9 level 3 of emotion data flow diagram

The above diagram shows the level 3 of the emotion data flow diagram ,where the above classification results is checked for severity high or low and the symptoms are shown.

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 ARCHITECTURE DIAGRAM

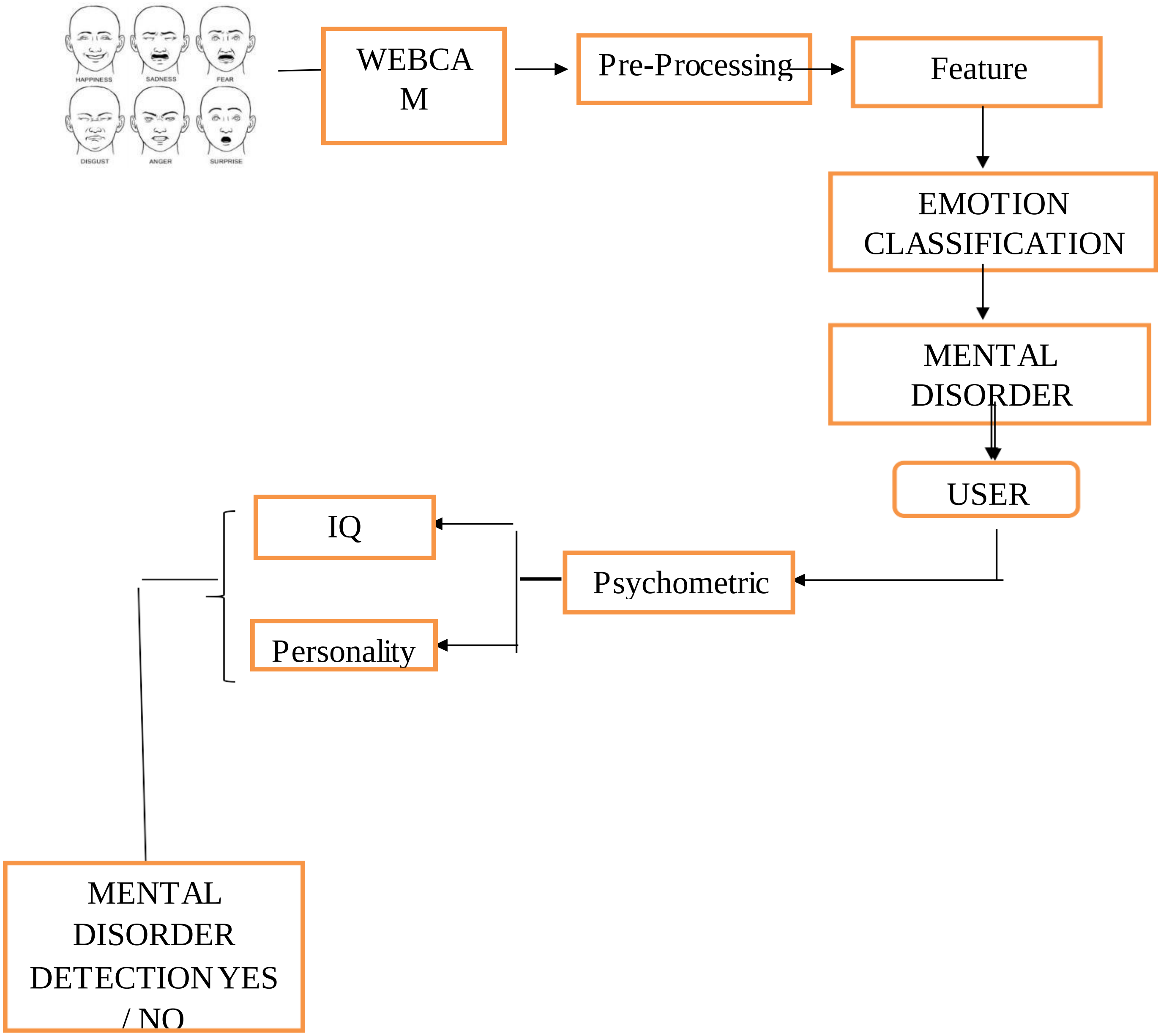


Figure 4.1 Architecture diagram for emotion recognition

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 ALGORITHM

5.1.1 POINT DETECTION ALGORITHM

A naive approach

Apply a mask over the image.

Apply thresholding

If |R| > T, then possible discontinuity

Depending on the value of T we can get:

4 points ($0 < T \leq 9$)

1 point ($9 < T \leq 72$)

0 points ($T > 7$)

A better approach

- Let us assume that f x and f y denote the partial derivatives of the image f .

- Let us assume a corner p and a neighborhood Q of p (e.g., 3x3, 5x5. etc.)

- The following matrix C is very important for deciding the presence of

$$C = \begin{bmatrix} \sum_Q f_x^2 & \sum_Q f_x f_y \\ \sum_Q f_x f_y & \sum_Q f_y^2 \end{bmatrix} = \sum_Q \begin{bmatrix} f_x \\ f_y \end{bmatrix} [f_x \quad f_y]$$

- One way to examine the presence of a corner is to look the eigenvalues of *C*

(1) if the area is a region of constant intensity, both eigenvalues will be very small.

(2) if it contains an edge, there will be one large and one small eigenvalue (the eigenvector associated with the large eigenvalue will be parallel to the image gradient).

(3) if it contains edges at two or more orientations (i.e., a corner), there will be Two large eigenvalues (the eigenvectors will be parallel to the image gradients).

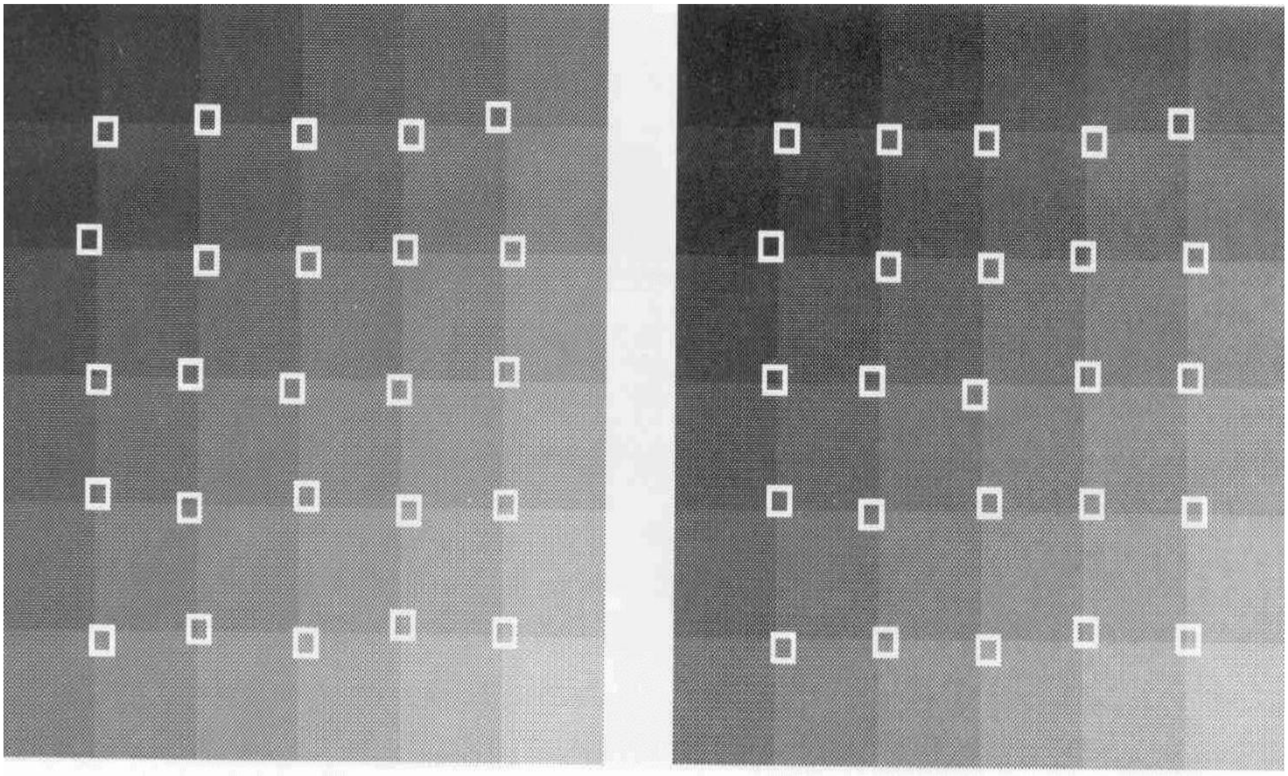


Figure 5.1 Example diagram 1

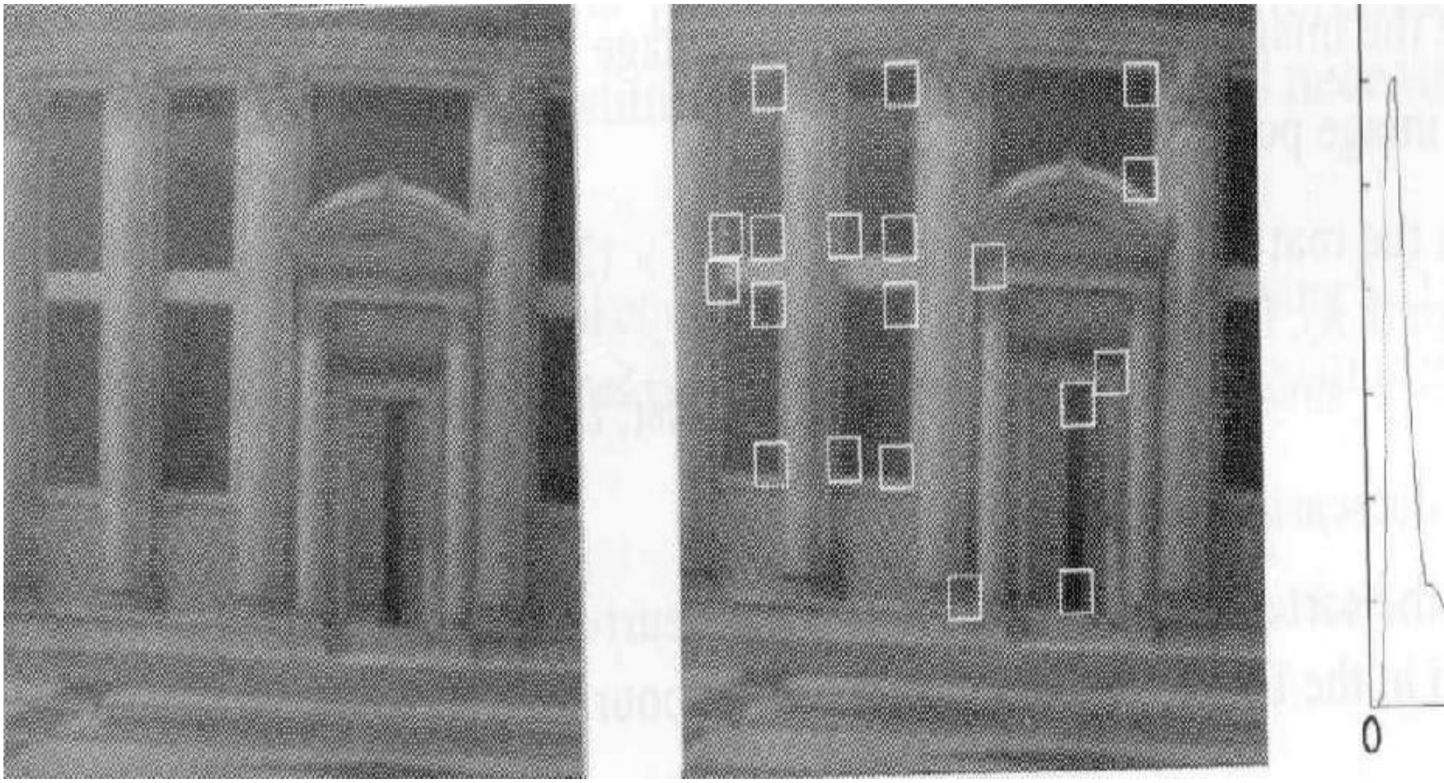


Figure 5.2 Example diagram 2

- How should we compare the eigenvalues?

1. find locations where the smaller eigenvalue is greater than some threshold.
2. consider the ratio of the two eigenvalues.

Algorithm

Input: image *f* , threshold *t* for 2 , size of *Q*

(1) Compute the gradient over the entire image *f*

(2) For each image point *p*:

(2.1) form the matrix *C* over the neighborhood *Q* of *p*

(2.2) compute 2, the smaller eigenvalue of *C*

(2.3) if 2 >*t*, save the coordinates of *p* in a list *L*

(3) Sort the list in decreasing order of 2

(4) Scanning the sorted list top to bottom: delete all the points that appear in the list that are in the same neighborhood *Q* with *p*

5.1.2 CLASSIFICATION- OPEN CV:

OpenCV (Open Source Computer Vision) is a library that can be imported in almost all computer languages like python, C, Java etc. It contains optimized image processing tools. Using OpenCV in python boosts its abilities by incorporating numpy (Numerical Python). In image processing, images are dealt as large 3D arrays and numpy serves as a robust tool for numerical array computations. The commands used for installing OpenCV, Matplot library and numpy in raspberry Pi are “sudo apt-get install python-opencv”, “sudo apt-get install pythonmatplotlib”, ”sudo apt-get install python-numpy”. The concept of using Matplot library in python is to plot a graphical representation as well as drawing steps.

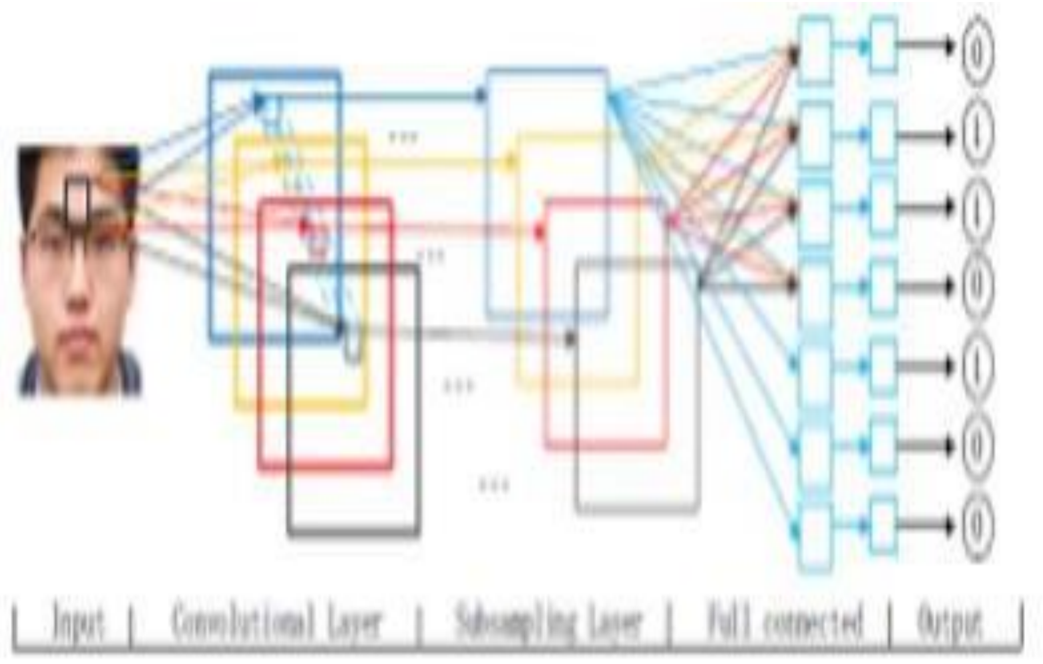


Fig.3 The model of CNN

Figure 5.3 Example diagram 3.

5.2 MODULEDESCRIPTION

5.2.1Facial Expression Recognition:

The thresholding technique is the simplest method used in the segmentation process. The process collects all the pixels with a certain threshold and rejects other pixels which have values less than the threshold. After the thresholding procedure is applied the stroke region will be isolated from the brain tissue. Color, intensity, orientation are been analyzed in the segmentation session. The stroke regions will be more clearly visible in the output images.

The input image to the system can be captured using a web cam or can be acquired from the hard disk. This image undergoes image enhancement, where tone mapping is applied to images with low contrast to restore the original contrast of the image.

Binarization :

All RGB and gray scale images are converted into a binary image. This preprocessed image is fed into the face detection block.

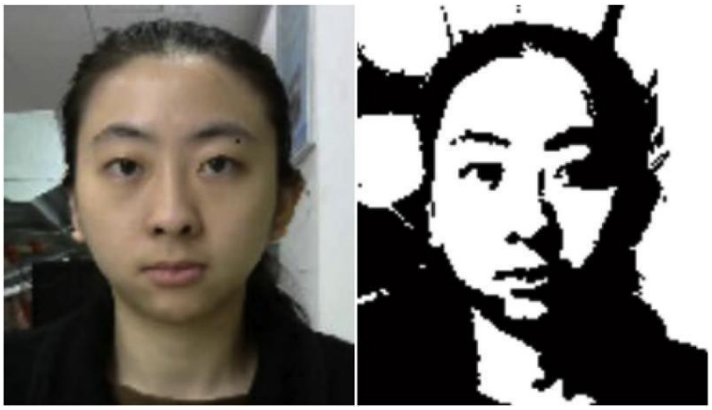


Figure 5.1 Binarization image of a given sample

5.2.2 Segmentation:

A bounding box is formed over every feature of the face that contributes to an emotion. Each bounding box is derived using face coordinates. These boxes are thus segmented and studied further to derive an emotion

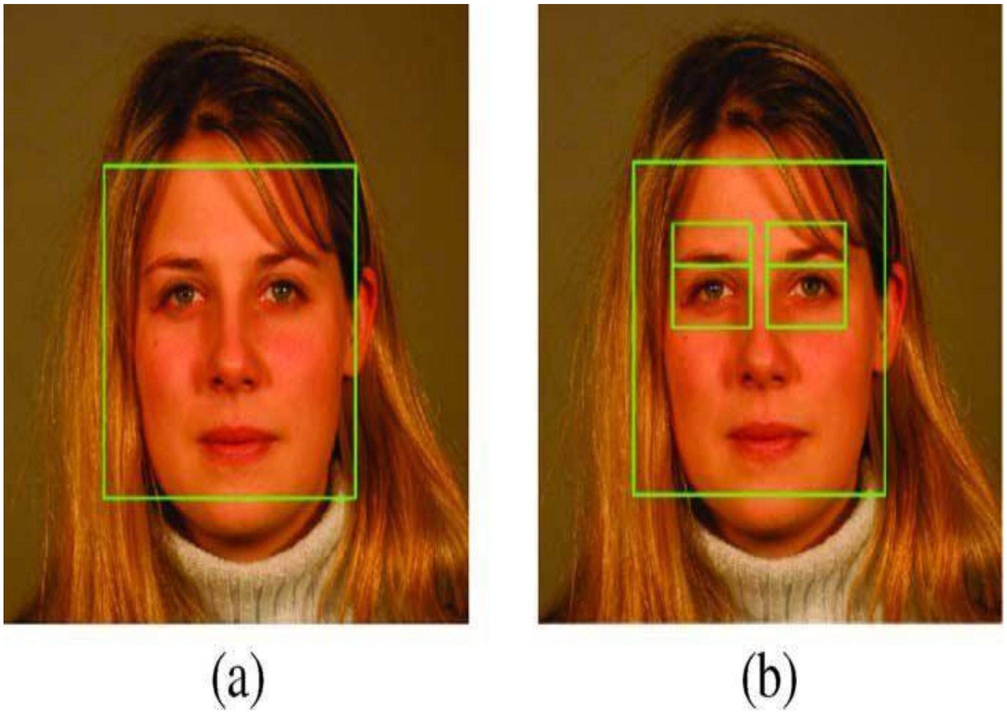


Figure 5.2 Segmentation image of a given sample

The segmentation step helps in recognizing the accurate emotion by separating each part of the human face. It partitions an image into distinct regions containing each pixels with similar attributes. To be meaningful and useful for image analysis and interpretation, the regions should strongly relate to depicted objects or features of interest. Meaningful segmentation is the first step from low-level image processing transforming a greyscale or colour image into one or more other images to high-level image description in terms of features, objects, and scenes. The success of image analysis depends on reliability of segmentation, but an accurate partitioning of an image is generally a very challenging problem.

5.2.3 Feature extraction:

The facial image obtained from the face detection stage forms an input to the feature extraction stage. To obtain real time performance and to reduce time complexity, for the intent of expression recognition, only eyes and mouth are considered. The combination of two features is adequate to convey emotions accurately.

Finally, a corner point detection algorithm was used to obtain the required corner points from the feature regions.

Eye Extraction

The eyes display strong vertical edges (horizontal transitions) due to its iris and eye white. Thus, the Sobel mask is applied to an image and the horizontal projection of vertical edges can be obtained to determine the Y coordinate of the eyes.

Eyebrow Extraction

Two rectangular regions in the edge image which lies directly above each of the eye regions are selected as the eyebrow regions. The edge images of these two areas are obtained for further refinement. Now sobel method was used in obtaining the edge image since it can detect more edges than roberts method. These obtainededge images are then dilated and the holes are filled.

Mouth Extraction:

The top, bottom, right most and left most points of the mouth are been extracted and the centriod of the mouth is calculated.

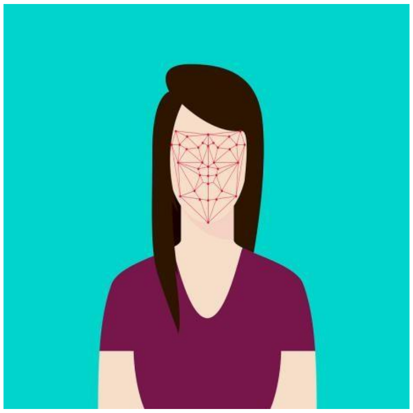


Figure 5.3 Diagram of mouth extraction.

5.2.4 Facial Emotion Recognition:

- An emotion matrix is constructed with discrete set of values.
- The rows and columns represent the extent of each motion in the person
- These values set a range that determines the permutations of different emotions

The accuracy of emotion recognition is usually improved when it combines the analysis of human expressions from multimodal forms such as texts, physiology, audio, or video .Different emotion types are detected through the integration of information from facial expression, body movement and gestures, and speech. The technology is said to contribute

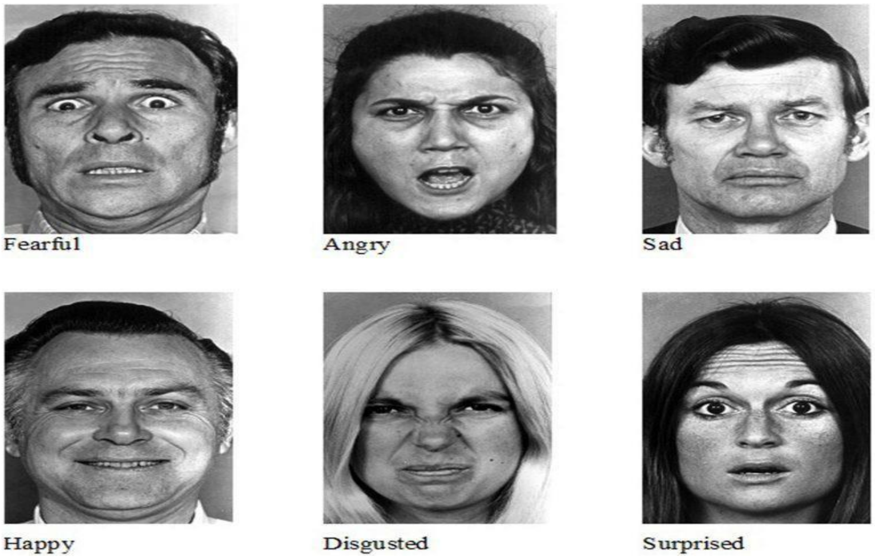
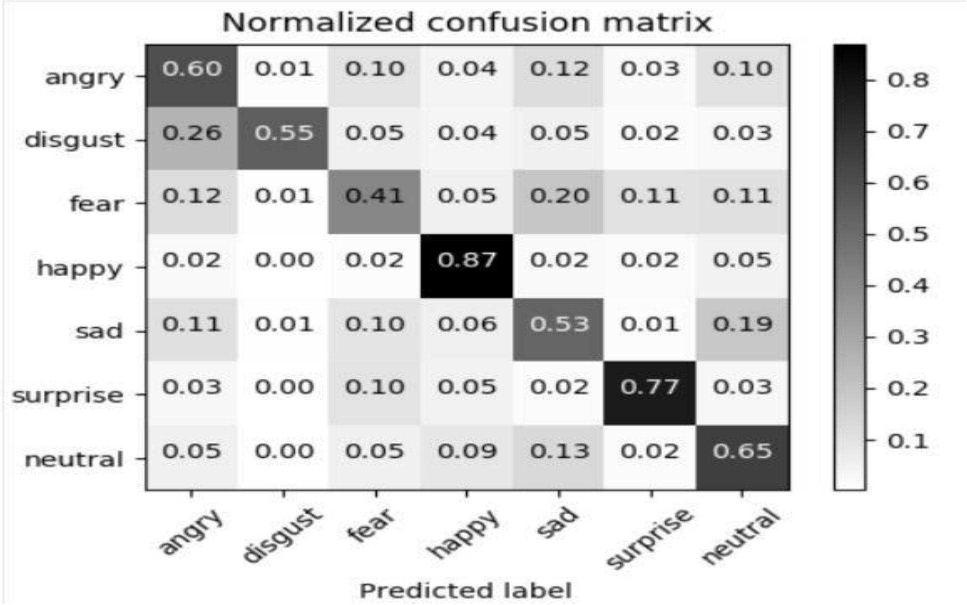


Figure 5.4 Diagram for facial expression recognition.



	angry	disgust	fear	happy	sad	surprise	neutral
angry	0.60	0.01	0.10	0.04	0.12	0.03	0.10
disgust	0.26	0.55	0.05	0.04	0.05	0.02	0.03
fear	0.12	0.01	0.41	0.05	0.20	0.11	0.11
happy	0.02	0.00	0.02	0.87	0.02	0.02	0.05
sad	0.11	0.01	0.10	0.06	0.53	0.01	0.19
surprise	0.03	0.00	0.10	0.05	0.02	0.77	0.03
neutral	0.05	0.00	0.05	0.09	0.13	0.02	0.65

Figure 5.5 Normalized confusion matrix

Mental State Detection

The emotions recognized are stored in a data frame and monitored over a period of time. Depending on the persistence and repetition of the emotions, mental state of the person is detected. Further a psychology test is taken up to evaluate the severity of the clinical condition.

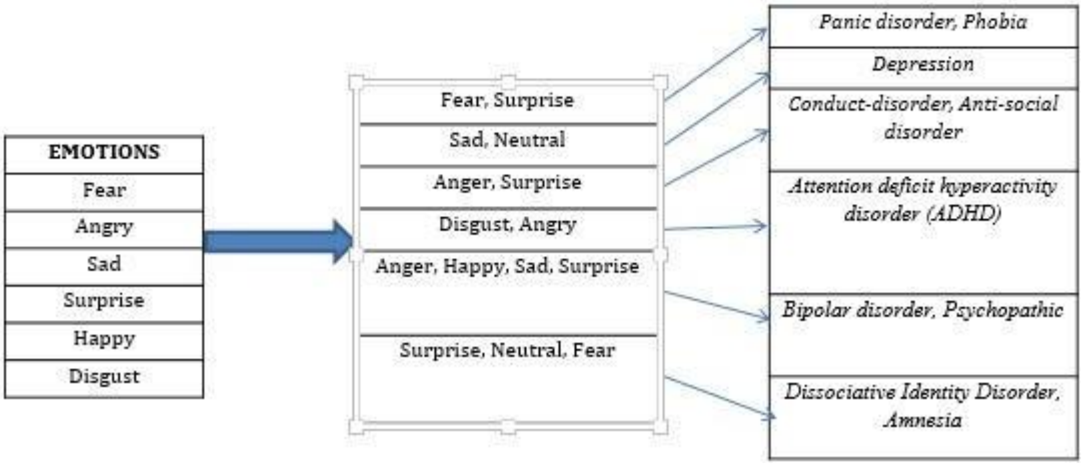

EMOTIONS
Fear
Angry
Sad
Surprise
Happy
Disgust
Fear, Surprise
Sad, Neutral
Anger, Surprise
Disgust, Angry
Anger, Happy, Sad, Surprise
Surprise, Neutral, Fear
Panic disorder, Phobia
Depression
Conduct-disorder, Anti-social disorder
Attention deficit hyperactivity disorder (ADHD)
Bipolar disorder, Psychopathic
Dissociative Identity Disorder, Amnesia

Figure 5.6 Mental state detection diagram.

Evaluation of Psychometrics

The term psychometrics refers to the design and interpretation of tests that measure psychological values such as aptitude, ability, personality, memory, happiness and intelligence. These tests are designed for, and given in, mental health, education and employment settings.

In this project, we measure the IQ value and the personality traits of the person. Hans Eysenck's personality test is used to measure personality based on three dimensions psychoticism, extraversion and neuroticism (PEN). Personality disorders like depression, anxiety, dissociation, positive and negative, Agreeableness and Openness, Neuroticism, Psychoticism etc can be detected and preventive measure can be addr4essed at the early stage itself.

Classification of criminal behavior / mental disorder:

By considering the output of Psychometrics and mental state as parameters, the probability and the nature of criminal behavior / mental disorder is classified as self-destructive and harmful to the society.

CHAPTER 6

RESULTS AND CODING

6.1 SOFTWARE REQUIREMENTS

Operating system	:	Windows 7 Professional.
Coding Language	:	Python.
Backend	:	SQLite
Frontend	:	HTML

6.2 HARDWARE REQUIREMENTS

Raspberry Pi Controller

System	:	Pentium IV 2.4 GHz
Hard Disk	:	40 GB
Monitor	:	15 VGA Color
Mouse	:	Logitech
Ram	:	512 MB.

6.3 SOFTWARE ENVIRONMENT

PyCharm is a Python IDE for professional developers created by JetBrains. It provides developers with all the tools they need to increase their coding productivity and also offers them smart code assistance.

With PyCharm, developers are able to improve their code's quality. The developing tool does this with quality checks, testing aid, intelligent refactoring, and framework support among others. With this, code writers can spend more time with additional lines instead of debugging.

Moreover, PyCharm has a user-definable interface so they are free to customize their tools. On top of that, there are also more than fifty plugins available that developers can utilize to enhance their PyCharm experience.

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django.

PyCharm is a Python IDE with complete set of tools for Python development. In addition, the IDE provides capabilities for professional Web development using the Django framework. Code faster and with more easily in a smart and configurable editor with code completion, snippets, code folding and split windows support.

Smart Python

With PyCharm, developers are assisted by an intelligent platform that aids them when it comes to code completions, inspections, error pinpointing, quick fixes, and more. This enables them to bolster their productivity as their codes can be automatically completed. Plus, they do not need to waste time scouring their codes for errors to rectify.

PyCharm Features

- **Project Code Navigation -** Instantly navigate from one file to another, from method to its declaration or usages, and through classes hierarchy. Learn keyboard shortcuts to be even more productive
- **Code Analysis -** Take advantage of on-the-fly code syntax, error highlighting, intelligent inspections and one-click quick-fix suggestions to make code better

- **Python Refactoring -** Make project-wide code changes painlessly with rename, extract method/superclass, introduce field/variable/constant, move and pull up/push down refactorings
- **Web Development with Django -** Even more rapid Web development with Django framework backed up with excellent HTML, CSS and JavaScript editors. Also with CoffeeScript, Mako and Jinja2 support
- **Google App Engine Support -** Develop applications for Google App Engine and delegate routine deployment tasks to the IDE. Choose between Python 2.5 or 2.7 runtime
- **Version Control Integration -** Check-in, check-out, view diffs, merge — all in the unified VCS user interface for Mercurial, Subversion, Git, Perforce and other SCMs
- **Graphical Debugger -** Fine-tune Python or Django applications and unit tests using a full-featured debugger with breakpoints, stepping, frames view, watches and evaluate expressions
- **Integrated Unit Testing -** Run a test file, a single test class, a method, or all tests in a folder. Observe results in graphical test runner with execution statistics
- **Customizable & Extensible -** Bundled Textmate, NetBeans, Eclipse & Emacs keyboard schemes, and Vi/Vim emulation plugin

Features:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: including rename, extract method, introduce variable,

introduce constant, pull up, push down and others

- Support for web frameworks: Django, web2py and Flask
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with changelist and merge.

SYSTEM TESTING

This testing approach document is designed for Information and Technology Services' upgrades to PeopleSoft. The document contains an overview of the testing activities to be performed when an upgrade or enhancement is made, or a module is added to an existing application. The emphasis is on testing critical business processes, while minimizing the time necessary for testing while also mitigating risks. It's important to note that reducing the amount of testing done in an upgrade increases the potential for problems after go-live. Management will need to determine how much risk is acceptable on an upgrade by upgrade basis.

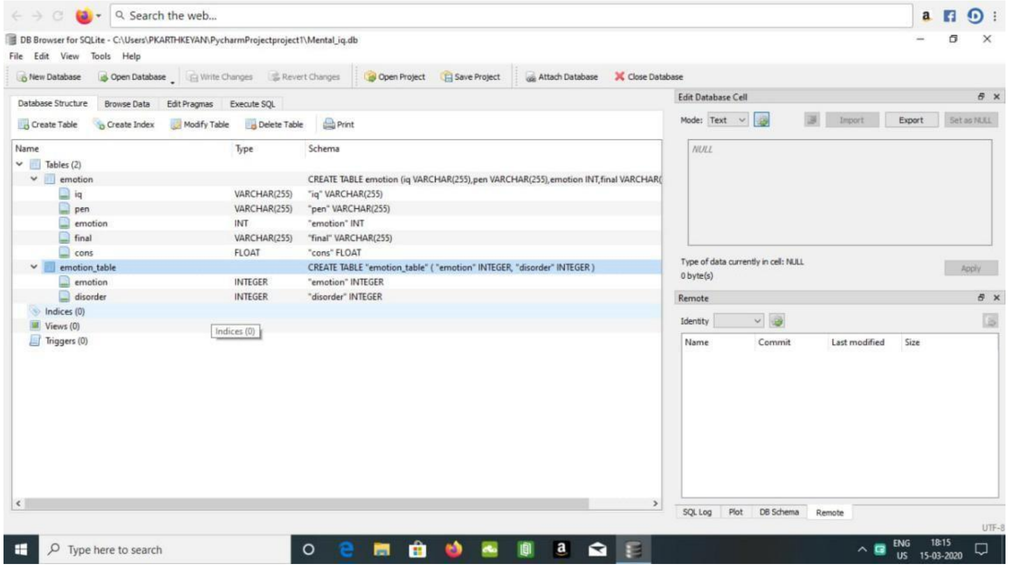
System testing is simply testing the system as a whole; it gets all the integrated modules of the various components from the integration testing phase and combines all the different parts into a system which is then tested. Testing is then done on the system as all the parts are now integrated into one system the testing phase will now have to be done on the system to check and remove any errors or bugs.

Unit Testing or Component Testing

It is the first and the most basic level of Software Testing, in which a single unit (i.e. a smallest testable part of a software) is examined in isolation from the remaining source code. Unit Testing is done to verify whether a unit is functioning properly. In other words, it checks the smallest units of code and proves that the particular unit can work perfectly in isolation. However, one needs to make sure that when these units are combined, they work in a cohesive manner. This directs us to other levels of software testing.

In computer programming, **unit testing** is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use.[1] Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others.

Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.



DB Browser for SQLite - C:\Users\PKARTHKEYAN\PycharmProjectproject1\Mental_iq.db

Name	Type	Schema
Tables (2)		
emotion		CREATE TABLE emotion (iq VARCHAR(255),pen VARCHAR(255),emotion INT,final VARCHAR(
iq	VARCHAR(255)	"iq" VARCHAR(255)
pen	VARCHAR(255)	"pen" VARCHAR(255)
emotion	INT	"emotion" INT
final	VARCHAR(255)	"final" VARCHAR(255)
cons	FLOAT	"cons" FLOAT
emotion_table		CREATE TABLE "emotion_table" ("emotion" INTEGER, "disorder" INTEGER)
emotion	INTEGER	"emotion" INTEGER
disorder	INTEGER	"disorder" INTEGER
Indices (0)		
Views (0)		
Triggers (0)		

Type of data currently in cell: NULL
0 byte(s)

Name	Commit	Last modified	Size

6.4 SAMPLE CODE

Sample code for preprocessing step

```
import numpy as np
from scipy.misc import imread, imresize


def preprocess_input(x, v2=True):
    x = x.astype('float32')
    x = x / 255.0
    if v2:
        x = x - 0.5
        x = x * 2.0
    return x

def _imread(image_name):
        return imread(image_name)

def _imresize(image_array, size):
        return imresize(image_array, size)

def to_categorical(integer_classes, num_classes=2):
    integer_classes = np.asarray(integer_classes,
dtype='int')
    num_samples =  integer_classes.shape[0]
    categorical = np.zeros((num_samples, num_classes))
    categorical[np.arange(num_samples), integer_classes] = 1
    return categorical
```

Class for Start page

```
class StartPage(tk.Frame):
    output=0

    def_init____(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self,
                         text="Mental Health Monitoring
System",
                         font=controller.title_font)
        label.pack(side="top", fill="x", pady=10)
        label = tk.Label(self,
                         text="Psychometric Tests",
                         font=controller.title_font)
        label.pack(side="top", fill="x", pady=50)

        button1 = tk.Button(self, text="IQ TEST",width=19,
height=-11,
                            command=lambda:
controller.show_frame("PageOne"))
        button2 = tk.Button(self, text="PEN PERSONALITY
TEST",
                            command=lambda:
controller.show_frame("PageTwo"))
        button1.pack(side="top", fill="x", pady=20,padx=100)
        button2.pack(side="top", fill="x", pady=20,padx=100)

        b = tk.Button(self, text='submit', command=self.main)
        b.pack(side="top", fill="x", pady=20,padx=100)

        load =  Image.open("./IQ.png")
        load = load.resize((220, 220))
        render = ImageTk.PhotoImage(load)
        img = tk.Label(self, image=render)
        img.image = render
        img.place(x=255, y=380)

        def close_window():
            app.destroy()

        button = tk.Button(text="Good-bye",
command=close_window)
        button.pack()
    def main(self):
        output_list=[]
        USE_WEBCAM = True  # If false, loads video file
source

        # parameters for loading data and images
        emotion_model_path = './models/emotion_model.hdf5'
        emotion_labels = get_labels('fer2018')

        # hyper-parameters for bounding boxes shape
        frame_window = 10
        emotion_offsets = (20, 40)

        # loading models
        face_cascade =
cv2.CascadeClassifier('./models/haarcascade_frontalface_defa
ult.xml')
        emotion_classifier = load_model(emotion_model_path)

        # getting input model shapes for inference
        emotion_target_size =
emotion_classifier.input_shape[1:3]

        # starting lists for calculating modes
        emotion_window = []

        # starting video streaming

        cv2.namedWindow('window_frame')
        cap = cv2.VideoCapture(0)

        while cap.isOpened():  # True:
            ret, bgr_image = cap.read()

            gray_image = cv2.cvtColor(bgr_image,
cv2.COLOR_BGR2GRAY)

            rgb_image = cv2.cvtColor(bgr_image,
cv2.COLOR_BGR2RGB)
```

```
            faces = face_cascade.detectMultiScale(gray_image,
scaleFactor=1.1, minNeighbors=5,

minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)

            for face_coordinates in faces:

                x1, x2, y1, y2 =
apply_offsets(face_coordinates, emotion_offsets)
                gray_face = gray_image[y1:y2, x1:x2]

                try:
                    gray_face = cv2.resize(gray_face,
(emotion_target_size))
                except:
                    continue

                gray_face = preprocess_input(gray_face, True)
                gray_face = np.expand_dims(gray_face, 0)
                gray_face = np.expand_dims(gray_face, -1)
                # cv2.imshow('window1', bgr_image)
                emotion_prediction =
emotion_classifier.predict(gray_face)
                emotion_probability =
np.max(emotion_prediction)
                emotion_label_arg =
np.argmax(emotion_prediction)
                emotion_text =
emotion_labels[emotion_label_arg]
                emotion_window.append(emotion_text)

                if len(emotion_window) > frame_window:
                    emotion_window.pop(0)
                try:
                    emotion_mode = mode(emotion_window)
                except:
                    continue

                if emotion_text == 'angry':
                    if emotion_probability >= 0.48181498 and
emotion_probability <= 0.49352397:

                        cap.release()
                        cv2.destroyAllWindows()

self.controller.show_frame("PageThree")
                        output_list.append(0)


                    color = emotion_probability *
np.asarray((255, 0, 0))
                elif emotion_text == 'sad':
                    if emotion_probability >= 0.32582545 and
emotion_probability <= 0.3314238:
                        cap.release()
                        cv2.destroyAllWindows()

self.controller.show_frame("PageThree")
                        output_list.append(1)
                    #     break
                    color = emotion_probability *
np.asarray((255, 255, 0))
                elif emotion_text == 'happy':
                    if emotion_probability >= 0.9015124 and
emotion_probability <= 0.9188275:
                        cap.release()
                        cv2.destroyAllWindows()

self.controller.show_frame("PageThree")
                        output_list.append(2)
                        # break
                    color = emotion_probability *
np.asarray((0, 255, 0))
                elif emotion_text == 'surprise':
                    if emotion_probability >= 0.7189152 and
emotion_probability <= 0.73248416:
                          cap.release()
                          cv2.destroyAllWindows()
```

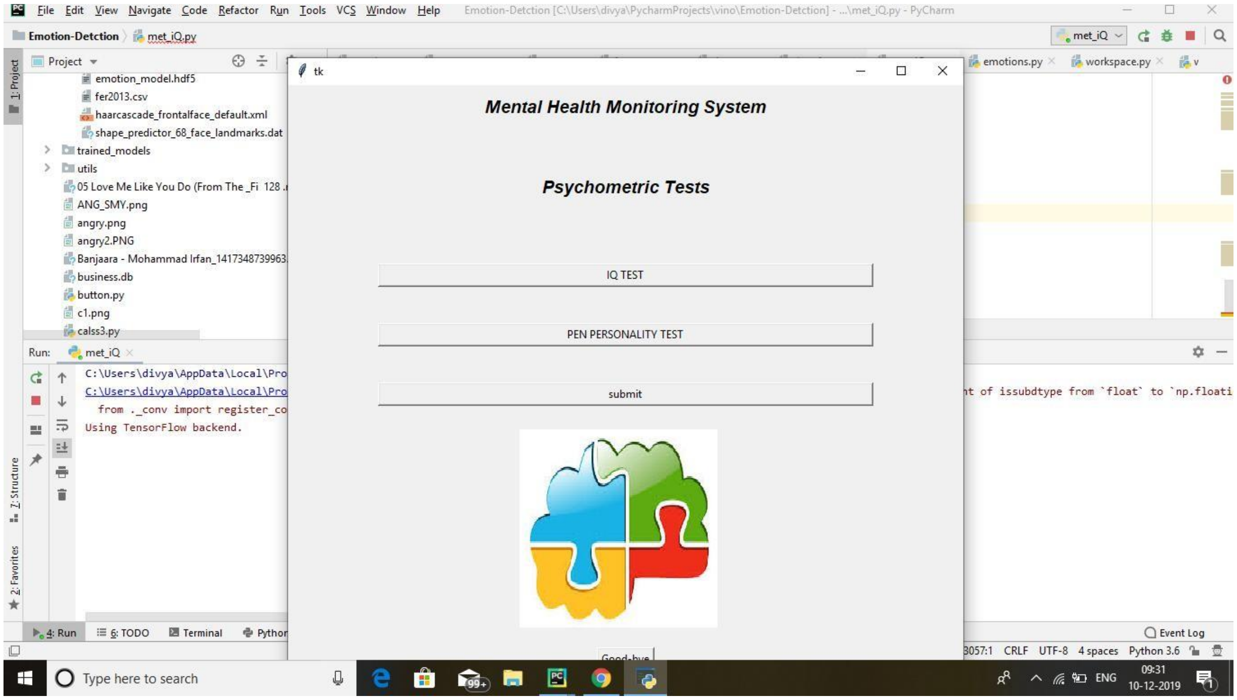
```
self.controller.show_frame("PageThree")
                          output_list.append(3)

                    color = emotion_probability *
np.asarray((255, 255, 255))
                else:
                    color = emotion_probability *
np.asarray((0, 255, 0))

                color = color.astype(int)
                color = color.tolist()

                draw_bounding_box(face_coordinates,
rgb_image, color)
                draw_text(face_coordinates, rgb_image,
emotion_mode,
                          color, 0, -45, 1, 1)

            bgr_image = cv2.cvtColor(rgb_image,
cv2.COLOR_RGB2BGR)

            cv2.imshow('window_frame', bgr_image)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                 self.controller.show_frame("PageThree")
                 break
        StartPage.output=output_list
        cap.release()
        cv2.destroyAllWindows()



# --- put frame in canvas ---

frame = tk.Frame(canvas)
canvas.create_window((50, 1330), window=frame, anchor=tk.W)


v = tk.IntVar()
v1 = tk.IntVar()
```

```
v2 = tk.IntVar()
v3 = tk.IntVar()
v4 = tk.IntVar()
v5 = tk.IntVar()
v6 = tk.IntVar()
v7 = tk.IntVar()
v8 = tk.IntVar()
v9= tk.IntVar()
v10 = tk.IntVar()
v11 = tk.IntVar()
v12 = tk.IntVar()
v13 = tk.IntVar()
v14 = tk.IntVar()


def selected_C():
    mylist_C = list()

    which_button_is_selected = v.get()
    A=0
    if which_button_is_selected == 1:
        A = 0
    elif which_button_is_selected == 2:
        A = 1
    elif which_button_is_selected == 3:
        A = 0
    elif which_button_is_selected == 4:
        A = 0
    mylist_C.append(A)

    B = 0
    which_button_is_selected1 =v1.get()
    if (which_button_is_selected1 == 1):
        B = 0
    elif (which_button_is_selected1 == 2):
        B = 0
    elif (which_button_is_selected1 == 3):
        B = 1
    elif (which_button_is_selected1 == 4
```

6.2 SAMPLE SCREENSHOTS

The below screenshots will explain the flow of output in each process.

6.2.1 Start page



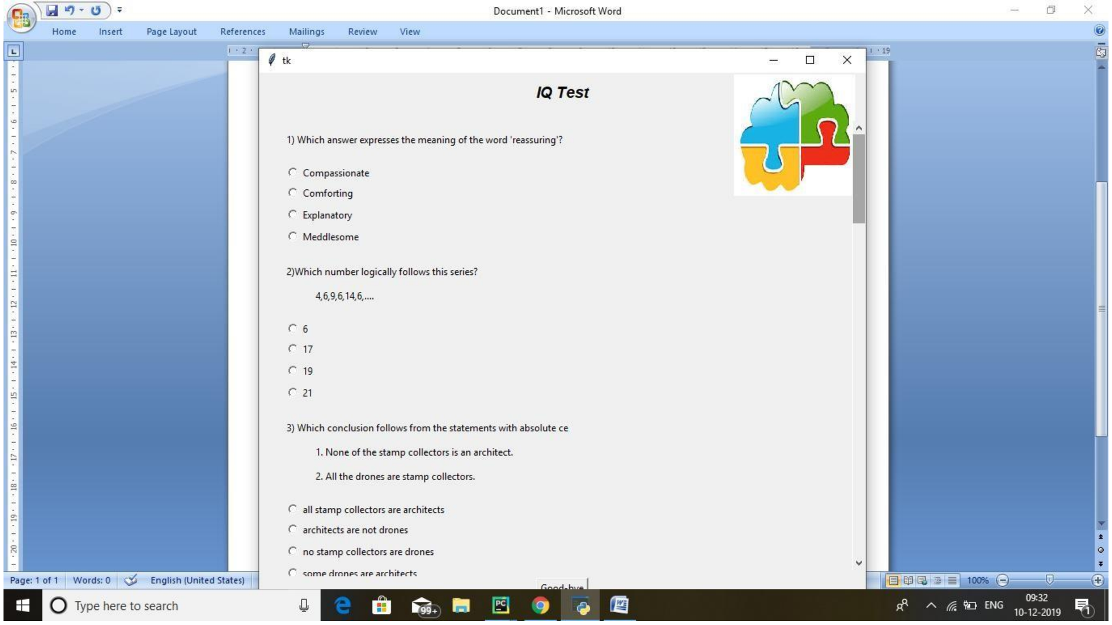
```
C:\Users\divya\AppData\Local\Pro
C:\Users\divya\AppData\Local\Pro
  from ._conv import register_co
Using TensorFlow backend.
```

```
nt of issubdtype from `float` to `np.floati
```

Figure 6.1 Sample screenshot of the start page

This start page forms the user interface of the application as a whole ,which contains various test and at the end the submit and good bye button to navigate.This page makes the user to understand the procees flow and to navigate to the next sections easily.It also contains various tests that is finally combined to get the final output.

6.2.2 IQ Test Page



IQ Test

1) Which answer expresses the meaning of the word 'reassuring'?

○ Compassionate
○ Comforting
○ Explanatory
○ Meddlesome

2)Which number logically follows this series?

4,6,9,6,14,6,....

○ 6
○ 17
○ 19
○ 21

3) Which conclusion follows from the statements with absolute ce

1. None of the stamp collectors is an architect.

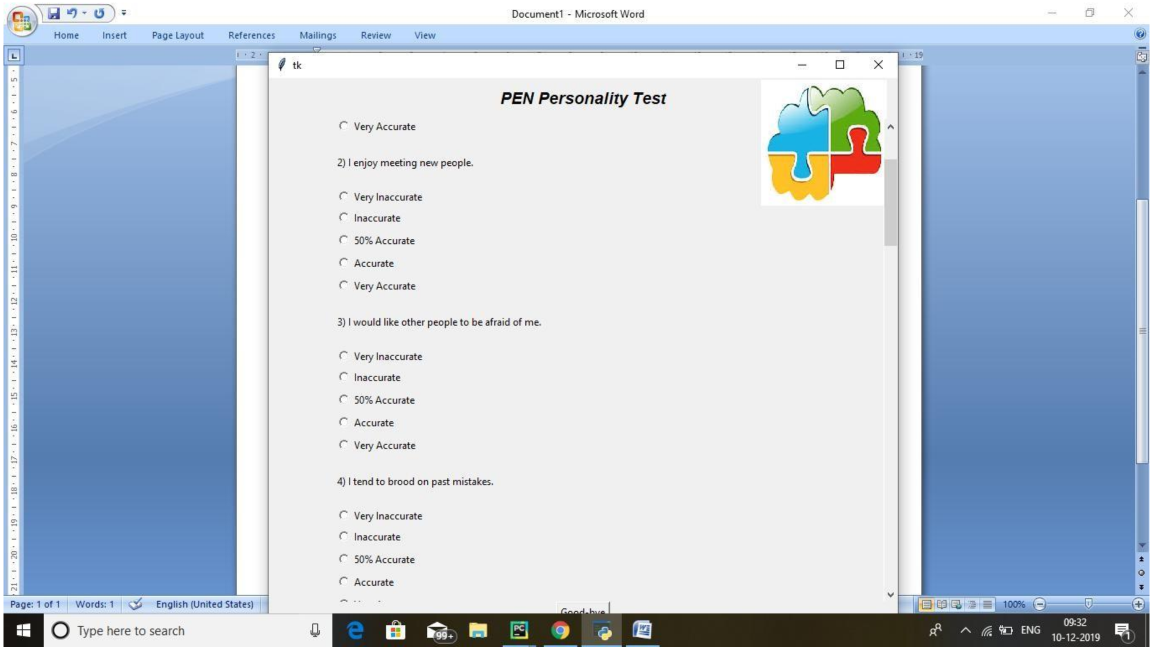
2. All the drones are stamp collectors.

○ all stamp collectors are architects
○ architects are not drones
○ no stamp collectors are drones
○ some drones are architects

Figure 6.2 Sample screenshot of IO test Page.

The above screenshot depicts the IQ test page which contain a series of questions which forms the IQ test.

6.2.3 PEN Test Page



PEN Personality Test

○ Very Accurate

2) I enjoy meeting new people.

○ Very Inaccurate
○ Inaccurate
○ 50% Accurate
○ Accurate
○ Very Accurate

3) I would like other people to be afraid of me.

○ Very Inaccurate
○ Inaccurate
○ 50% Accurate
○ Accurate
○ Very Accurate

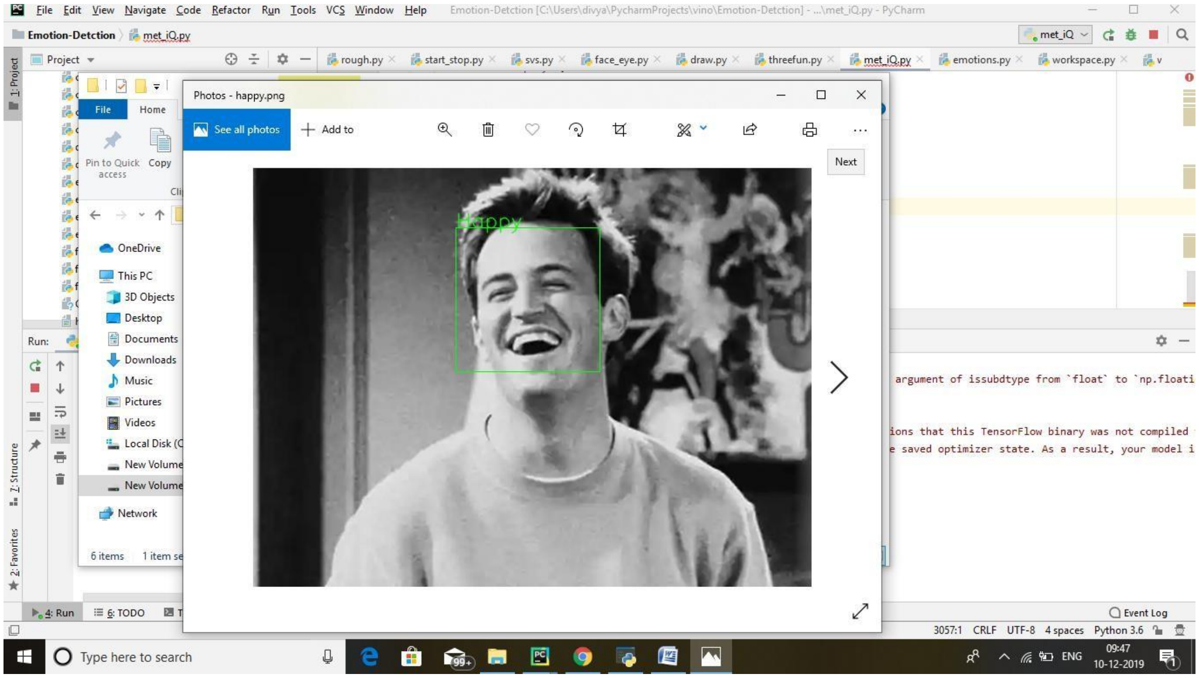
4) I tend to brood on past mistakes.

○ Very Inaccurate
○ Inaccurate
○ 50% Accurate
○ Accurate

Figure 6.3 Sample screenshot of Penetration test page.

The above image shows the series of questions for penetration test.

6.2.4 Capturing of face using webcam

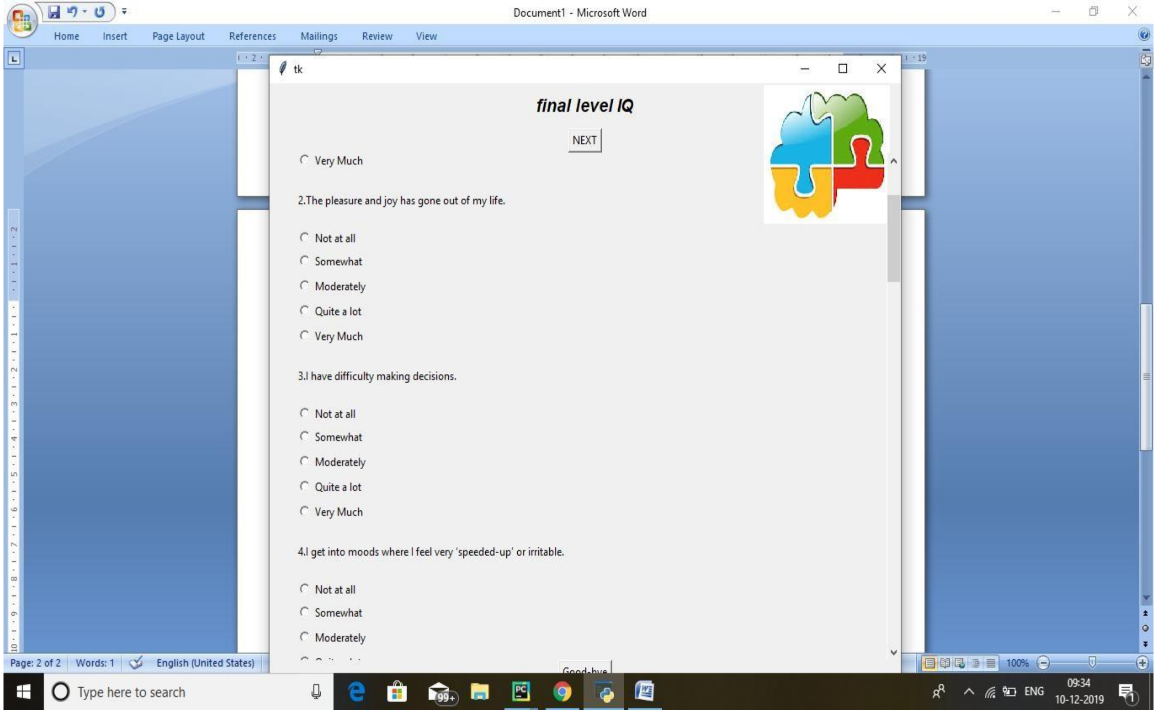


```
argument of issubdtype from `float` to `np.floati
ions that this TensorFlow binary was not compiled
e saved optimizer state. As a result, your model i
```

Figure 6.4 Sample screenshot of capturing faces

This picture shows the process of wab cam capturing the images of the user and process the further steps.

6.2.5 Final IQ Page



final level IQ

○ Very Much

2.The pleasure and joy has gone out of my life.

○ Not at all
○ Somewhat
○ Moderately
○ Quite a lot
○ Very Much

3.I have difficulty making decisions.

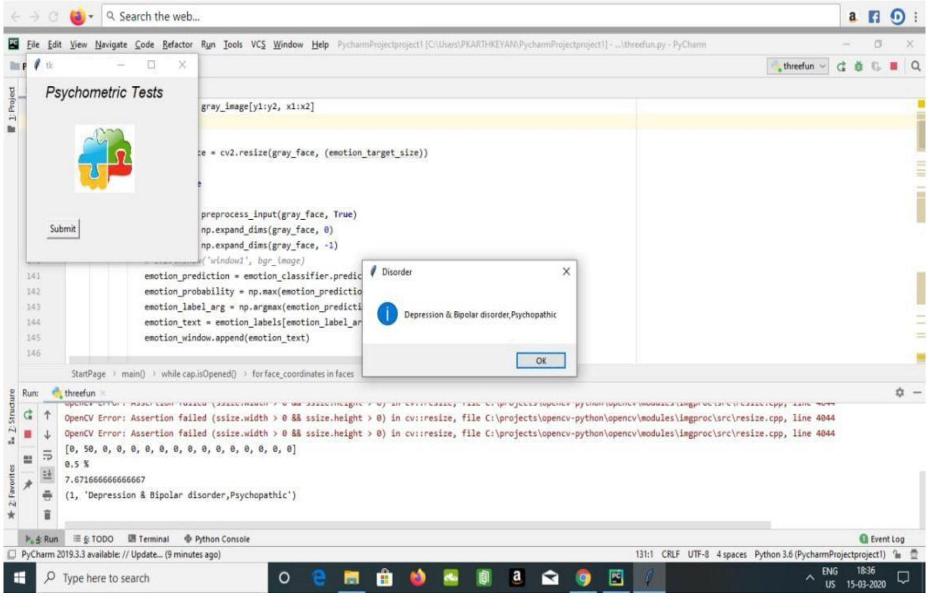
○ Not at all
○ Somewhat
○ Moderately
○ Quite a lot
○ Very Much

4.I get into moods where I feel very 'speeded-up' or irritable.

○ Not at all
○ Somewhat
○ Moderately

Figure 6.5 Sample screenshot of final IQ test

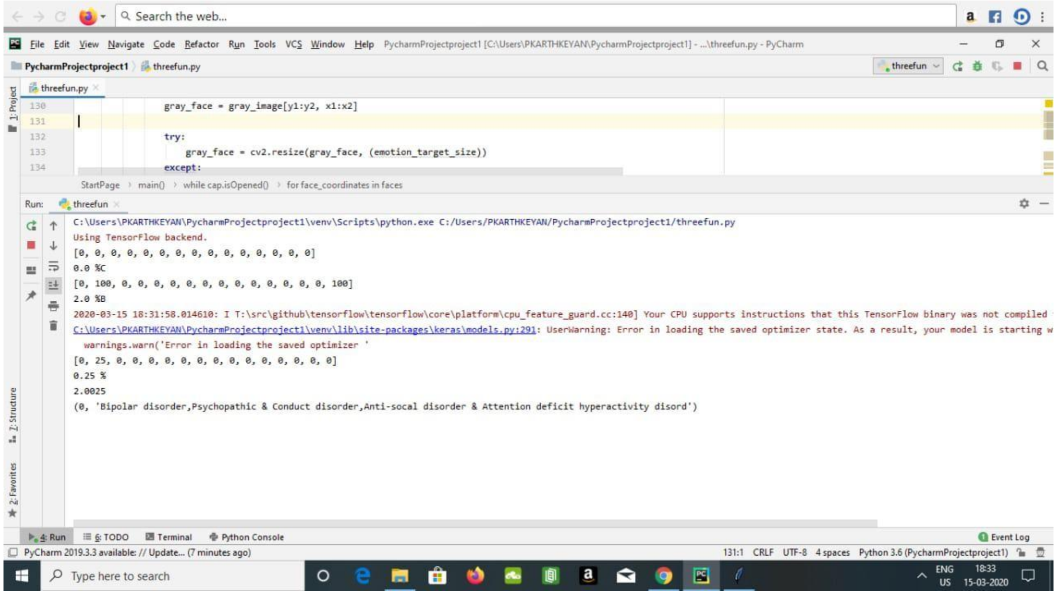
6.3 RESULTS AND GRAPHS



```
gray_image[y1:y2, x1:x2]

e = cv2.resize(gray_face, (emotion_target_size))

preprocess_input(gray_face, True)
np.expand_dims(gray_face, 0)
np.expand_dims(gray_face, -1)
('window1', bgr_image)
emotion_prediction = emotion_classifier.predic
emotion_probability = np.max(emotion_predictio
emotion_label_arg = np.argmax(emotion_predicti
emotion_text = emotion_labels[emotion_label_ar
emotion_window.append(emotion_text)
```

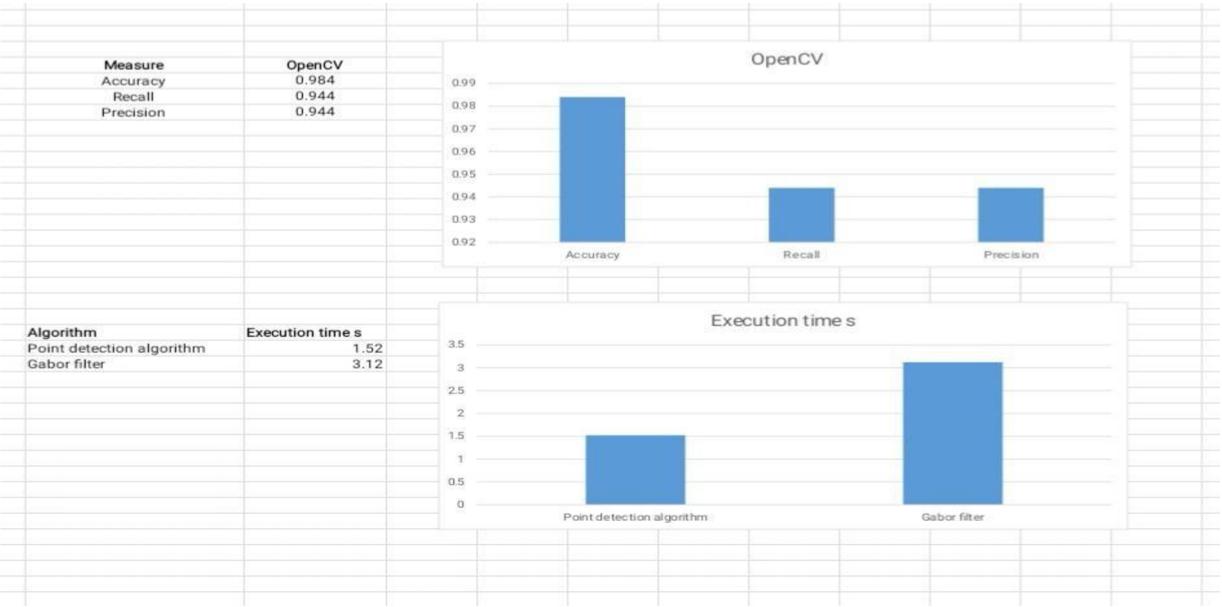
Depression & Bipolar disorder,Psychopathic

```
OpenCV Error: Assertion failed (ssize.width > 0 && ssize.height > 0) in cv::resize, file C:\projects\opencv-python\opencv\modules\imgproc\src\resize.cpp, line 4044
OpenCV Error: Assertion failed (ssize.width > 0 && ssize.height > 0) in cv::resize, file C:\projects\opencv-python\opencv\modules\imgproc\src\resize.cpp, line 4044
[0, 50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
0.5 %
7.671666666666667
(1, 'Depression & Bipolar disorder,Psychopathic')
```



```
gray_face = gray_image[y1:y2, x1:x2]

try:
    gray_face = cv2.resize(gray_face, (emotion_target_size))
except:
```

```
C:\Users\PKARTHKEYAN\PycharmProjectproject1\venv\Scripts\python.exe C:/Users/PKARTHKEYAN/PycharmProjectproject1/threefun.py
Using TensorFlow backend.
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
0.0 %C
[0, 100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100]
2.0 %B
2020-03-15 18:31:58.014610: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled
C:\Users\PKARTHKEYAN\PycharmProjectproject1\venv\lib\site-packages\keras\models.py:291: UserWarning: Error in loading the saved optimizer state. As a result, your model is starting w
  warnings.warn('Error in loading the saved optimizer '
[0, 25, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
0.25 %
2.0025
(0, 'Bipolar disorder,Psychopathic & Conduct disorder,Anti-socal disorder & Attention deficit hyperactivity disord')
```



Measure	OpenCV
Accuracy	0.984
Recall	0.944
Precision	0.944

Algorithm	Execution time s
Point detection algorithm	1.52
Gabor filter	3.12

CHAPTER 7

CONCLUSION AND FUTURE WORK

CONCLUSION

It is essential to get accurate and reliable identification of stress and it requires a valid analysis and experimental methodology framework. The main contribution of the project is developing an experimental model for successfully identifying stress at multiple levels. Our research suggest facial emotion based system along with psychiatric tests with reliably to detect stress. However, to quantify stress into different levels it requires further analysis and validation.Testing on real time data by applying in hospitals .To help doctors, counsellors, therapists to identify and detect stress in patients using the framework . To relive working professionals of stress by identifying on daily basis.

Reference

1.Alschuler, A.W. (2003). The Changing Purposes of Criminal Punishment: A Retrospective on the Last Century and Some Thoughts about the Next. University of Chicago Law Review 70(1): 1–22.

2.Barkley, R.A. (1997). ADHD and the Nature of Self-Control. New York: The Guilford Press.

3.Baroni, A., & Castellanos, F.X. (2015). Stimulants, cognitions and ADHD. Current Opinion in Behavioral Sciences 4: 109–114.

4.Baumeister, R.F., & Exline, J.J. (1999). Virtue, Personality and Social Relations: Self-Control as the Moral Muscle. Journal of Personality 67(6): 1165–1194.

5.Beaver, K.M., DeLisi, M., Vaughn, M.G., & Wright, J.P. (2010). The Intersection of Genes and Neuropsychological Deficits in the Prediction of Adolescent Delinquency and Low Self-Control. International Journal of Offender Therapy and Comparative Criminology 54(1): 22–42.

6.Berman, M.G., Yourganov, G., Askren, M.K., Ayduk, O., Case, B.J., Gotlib, I.H., Kross, E., McIntosh, A.R., Strother, S., Wilson, N.L., Zayas, V., Mischel, W., Shoda, Y., & Jonides, J. (2013). Dimensionality of brain networks linked to life-long individual differences in self-control. Nature Communications 4(1373): 1–7. doi:10.1038/ncomms2374.

7.Burt, C.H., Simons, R.L., & Simons, L.G. (2006). A Longitudinal Test of the Effects of Parenting and the Stability of Self-Control: Negative Evidence for the General Theory of Crime. Criminology 44 (2): 353–396.

8.Caplan, A.L. (2006). Ethical issues surrounding forced, mandated, or coerced treatment. Journal of Substance Abuse Treatment 31(2): 117–120.