VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Jnana Sangama, Belgaum-590014, Karnataka, INDIA

A PROJECT REPORT
ON
"Efficient Eye Blink Detection for Disabled: Assisting System for Paralyzed"

Submitted in partial fulfillment of the requirements as prescribed for the award of

BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING

For the Academic year
2018-2019
Submitted by

ASWIN G	**1MV14IS009**
BHUVANA B A	**1MV15IS007**
POOJA SHIVAJI SHANBHAG	**1MV15IS031**
GIRISH SAGAR D S	**1MV16IS403**

Under the Guidance of
Mrs. Shanta H Biradar
Assistant Professor, Dept. of ISE
Sir MVIT, Bangalore-562157

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
HUNASAMARANAHALLI, BANGALORE -562157

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
HUNASAMARANAHALLI, BANGALORE -562157

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled "**Efficient Eye Blink Detection for Disabled: Assisting System for Paralyzed"** is a bonafide work carried out by **Aswin G (1MV14IS009), Bhuvana B A (1MV15IS007),Pooja Shivaji Shanbhag (1MV15IS031),Girish Sagar D S (1MV16IS403)** in partial fulfillment for the award of Degree of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University**,** Belgaum during the year 2018-2019. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signatures:

Project Guide	Prof and Head Dept of ISE	Principal
Mrs. Shanta H Biradar.	**Dr. P. Vijayakarthik**	**Dr. V.R.Manjunath**
Asst. Professor, Dept. of ISE	HOD, Dept. of ISE,	Sir MVIT, Bengaluru
Sir MVIT, Bengaluru	Sir MVIT, Bengaluru	

External Viva

Name of the examiners: **Signature with Date:**

1. ____________________ ____________________

2. ____________________ ____________________

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned my effort with success.

First, we would like to thank and express our heartiest regards and gratitude to our honourable principal **Dr. V R Manjunath** for facilitating us with all the necessary amenities and kept on inspiring us to get the best knowledge not only about the topics but also the subject, as a whole.

We are also extremely grateful to our Project Coordinator, **Mrs. Shanta H Biradar.,** Assistant Professor, Department of Information Science and Engineering, Sir MVIT, Bengaluru, for her constant support and advice throughout the course of preparation of this document.

We are greatly thankful to **Dr. P.Vijayakarthik**, Prof and Head, Department of Information Science and Engineering, Sir MVIT, Bengaluru, for his able guidance, regular source of encouragement and assistance throughout this project.

We gratefully acknowledge the help lent out to us by all the faculty members of the Department of Information Science and Engineering, Sir MVIT, Bengaluru, at all difficult times. We would also take this opportunity to thank our college management for the facilities provided during the course of the project. Furthermore, we acknowledge the support and feedback of our parents and friends.

ASWIN G (1MV14IS009)
BHUVANA B A (1MV15IS007)
POOJA SHIVAJI SHANBHAG (1MV15IS031)
GIRISH SAGAR DS (1MV16IS403)

DECLARATION

We hereby declare that the entire project work embodied in this dissertation was carried out by us and no part has been submitted to any degree or diploma institution previously.

Place: Bangalore

Date:

ASWIN G
(1MV14IS009)

BHUVANA B A
(1MV15IS007)

POOJA SHIVAJI SHANBHAG
(1MV15IS031)

GIRISH SAGAR D S
(1MV16IS403)

ABSTRACT

A real-time algorithm to detect eye blinks in a video sequence from a standard camera is proposed. Recent landmark detectors trained on in-the wild datasets exhibit excellent robustness against a head orientation with respect to a camera, varying illumination and facial expressions. We show that the landmarks are detected precisely enough to reliably estimate the level of the eye opening. The proposed algorithm therefore estimates the landmark positions, extracts a single scalar quantity – eye aspect ratio (EAR) – characterizing the eye opening in each frame. Finally, an SVM classifier detects eye blinks as a pattern of EAR values in a short temporal window. The simple algorithm outperforms the state-of-the-art results on two standard datasets.

A Haar Cascade Classifier Algorithm is applied for face and eye detection for getting eye and facial axis information. In addition, the same classifier is used based on Haar- like features to find out the relationship between the eyes and the facial axis for positioning the eyes.

An efficient eye tracking method is proposed which uses the position of detected face. Finally, an eye blinking detection based on eyelids state (close or open) is used for controlling android mobile phones. The method is used with and without smoothing filter to show the improvement of detection accuracy.

The application is used in real time for studying the effect of light and distance between the eyes and the mobile device in order to evaluate the accuracy detection and overall accuracy of the system. Test results show that our proposed method provides a 98% overall accuracy and 100% detection accuracy for a distance of 35 cm and an artificial light.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Overview

In these days electronic devices are improving day by day and their demand is also improving. Smart phones, tablets are example of this. The system detects the eye blink and differentiates between an intentional long blink and a normal eye blink. Tetraplegia is a condition where people cannot move parts below neck. The proposed system can be used to control and Communicate with other people. In the recent years due to the rapid advancement in the technology there has been a great demand of human computer or mobile interaction (HCI or HMI). Eye blink is a quick action of closing and opening of the eyelids. Blink detection is an important enabling component in various domains such as human computer interaction, mobile interaction, health care, and driving safety. For example, blink has been used as an input modality for people with disabilities to interact with computers and mobile phones.

In Viola the chain of single-feature filters, Haar Cascade Classifier for identifying sub-region image is used. With the fast calculation of integral image technique, it can work in real time. Eye tracking provides an almost seamless form of interaction with the modern graphical user interface, representing the fastest non-invasive method of measuring user interest and attention. While the mouse, keyboard, and other touch-based interfaces have long reigned as the primary mediums associated with the field of human computer interaction, as advances continue to improve the cost and accuracy of eye tracking systems, they stand poised to contend for this role. An open and close eye template for blink pattern decisions based on correlation measurement is used in. The method was specifically useful for people with severely paralyzed. A real-time eye blinking detection was proposed based on SIFT feature tracking with GPU based implementation.

An efficient method is proposed in. A method is based on image processing techniques for detecting human eye blinks and generating inter-eye-blink intervals. A Haar Cascade Classifier and Camshaft algorithms for face tracking and consequently are applied for getting facial axis information. Adaptive Haar Cascade Classifier from a cascade of boosted classifiers based on Haar-like features using the relationship between the eyes and the facial axis applied for positioning the eyes. The algorithm results show that the proposed method can work efficiently in real-time applications.

An EyePhone application which is developed in is a system that capable of driving mobile applications/functions using only the user's eyes movement and actions (e.g. Wink). EyePhone tracks the user's eye movement across the phone's display using the camera mounted on the front of the phone. The results indicate that EyePhone is a promising approach to driving mobile applications in a hand-free manner.

An efficient eye tracking system is presented in having a feature of blink detection for controlling an interface that provides an alternative way of Communication for the people who are suffering from severe physical disabilities the proposed system uses pupil portion for tracking the movement of eyes.

1.2 Problem Statement

Paralyzed people lack the ability to control muscle function in one or more muscle group. The condition can be caused by strokes, ALS, multiple sclerosis, and many other diseases. Locked in Syndrome (LIS) is a form of paralysis where a patient has lost in control of nearly all voluntary muscles. These people are unable to control any part of their body, besides eye moment and blinking. Due to their condition these people are unable to talk, text, and communicate in general. Even through people that have LIS are cognitively aware, their thoughts and ideas are locked inside of them. These people depend on eye blinks to communicate. They rely on nurse and caretakers to interpret and decode their blinking. Whenever patients do not have a person to read their Eyeblink, they have no means of self-expression.

1.3 The solution

BlinkToSpeak offers a form of independence to paralyzed people. The software platform converts eye blinks to Speak. Every feature of the software can be controlled by eye movement. Thus, the software can be independently operated by paralyzed people. Using the software, patients can record messages, recite those messages aloud, and send the messages to others. The software can be run on any low-end computer, from a Raspberry Pi to an IBM ThinkPad. The software uses computer vision and Haar cascades to detect eye blinking and convert the motion into text. The program uses language modelling to predict the next words that the user might blink. The software can be easily customized for each patient as well. Blink to Text is free open source software. It is distributed under the MIT Permissive Free Software License.

1.4 Key Objectives

- **Allow paralysis victims to communicate independently.**

Many paralysis victims already use eye blinks as a form of communication. It is common for nurses and caretakers to read a patient's eye blinks and decode the pattern. The ALS association even offers a communication guide that relies on eye blinks. BlinkToSpeak automates this task. The software reads a person's eye blinks and converts them into text. A key feature of the software is that it can be started, paused, and operated entirely with eye blinks. This allows patients to record their thoughts with complete independence. No nurses or caretakers are required to help patients express themselves. Not only does this reduce the financial burden on paralysis patients, but this form of independence can be morally uplifting as well.

- **Be accessible to people with financial constraints.**

Many companies are developing technologies that are controlled by eye movement. These technologies rely on expensive hardware to track a user`s eyes. While these devices can absolutely help LIS victims, they are only available to people that can afford the technology. BlinkToSpeak focuses on a different demographic that are often ignored. Twenty-eight percent of U.S. households with a person who is paralyzed make less than $15,000 per year. BlinkToSpeak is free and open source. The software runs on wide variety of low-end computers. The only required peripheral is a basic webcam. Not only is this software accessible to paralyzed people, but paralyzed people of almost all financial classes as well.

1.5 OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. It is designed for computational efficiency with strong focus on real-time application.

This document outlines the specifics of how to test OpenCV that has been released within Processor SDK. This release is based off OpenCV 3.1.

1.6 Facial Landmarks

Eye blinks can be detected by referencing significant facial landmarks. Many software libraries can plot significant facial features within a given region of interest. Python’s dlib library uses Kazemi and Sullivan’s One Millisecond Face Alignment with an Ensemble of Regression Trees to implement this feature.
The program uses a facial training set to understand where certain points exist on facial structures. if they exist, the program plots the same points on regions of interest in other images. The program uses priors to estimate the probable distance between key points.

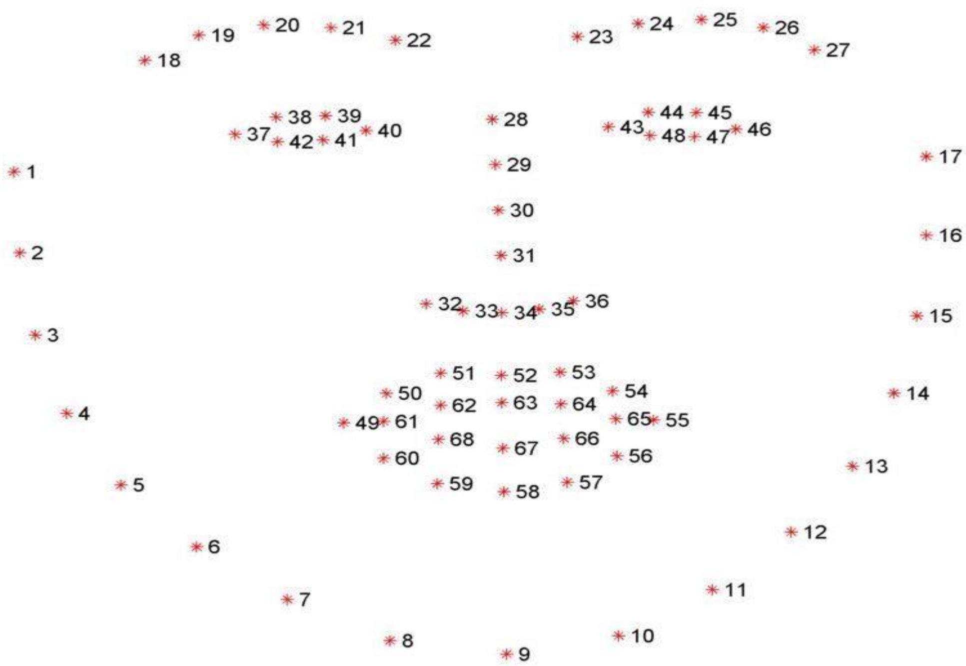


Fig 1.6 Facial Landmark

1.7 Objectives

To make cost effective:

The main objective of developing algorithm of a real time video Oculography system is that to provide cost effective for those people who cannot afford. The existing technique for such patients to communicate is too costly.

Thus, it is necessary to design a system which is affordable to common people which includes cost effective components for designing.

- To design a system which can assist paralyzed people in terms of communication
- face detection
- landmark estimation
- Voice Conversion

CHAPTER 2
LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

Various system and methods are involved while doing the work on communication model that is for paralyzed patient. Augmentative and Alternative communication device by using morse code for paralyzed patient.

2.1 Atish udayshankar, Amit R Kaushik-Assistance for the Paralyzed using Eye Blink Detection || IEEE paper Fourth International Conference on Digital Home 2012

In this system, the device that uses the signals from patient and convert it into some form of data that is for communication but this system is very expensive, so that they have developed extremely low priced device that read and convert eyes blinking of patient to universally accepted code that is morse code. 2.2 Using eyes blink detection assistance for the paralyzed person.

2.2 Assi.prof. Aree A. Mohammed, Suleiman, Shereen – Efficient Eye Blink Detection Method for disabled helping domain. International Journal of Computer science and Application.

In this method, they have designed a real-time interactive system that can help the paralyzed to control the appliances such as fan, lights etc. through the pre-recorded sample of eyes blinking. Detection of eyes blinking by using video camera with region of interest.

2.3 Kristen Grauman Margrit Betke James Gips Gary R. BradskiVision Communication via Eye Blinks Detection and Duration Analysis in Real Time Interface Group Image &amp.

Video Computing Eagle Eyes Visual Interactivity Group MIT AI Lab Boston University Boston College Intel Corporation 2001.

In this study, the patient eyes blinking is detected through the visual sensing system which is real-time subtraction of an image with the help of CCD camera. Eye blink detection method to control mobile phone.

2.4 Kingshek Mukherji, Debdatta Chattergi Augmentative and Alternative Communication Device Based on Eye-Blink Detection and Conversion to Morse-Code to Aid Paralyzed Individuals. International Conferencing Communication, information & amp.

Computing Technology (ICCICT), Jan. 16-17

In this study the motivation of this research is those people are physically disable or who cannot control the human mobile calls for interaction without using hands. For eyes and face detection they have used the haar cascade classifier.

2.5 Augmentative and Alternative Communication device based on Eye-Blink detection and Conversion to Morse- code aid paralyzed individuals.

There are several medical disorders that can lead to an individual becoming paralyzed or having motor speech disorders that inhibits speech or voice production. Conditions such as Locked-In Syndrome (LIS) or motor neuron diseases such as Amyotrophic Lateral Sclerosis (ALS) and Cerebral Palsy are among the common diseases that affect speech. In all or most such cases, the patient loses the ability to communicate with the external world in an effective manner even though his intelligence is mostly unaffected. Not only does that cause extreme distress to that individual, but also to his family and friends. Some customized Augmentative and Alternative Communication (AAC) devices have been developed that uses signals from the patient and converts them into some form of data that can be communicated but such devices are very expensive and are practically out of reach for most people. In this document, we have designed an extremely low-priced device that reads and converts eye-blinks from the patient to a universally accepted communication code-The Morse code.

Date Added to IEEE Xplore: 23 February 2015

2.6 Study and Development of Support Tool with Blinks for Physically Handicapped Children

In this study, we try to develop a new application for physically handicapped children to communicate with others by a blink. Because of limited body movements and mental disorders, many of them cannot communicate with their families or caregivers. We think if they can use application in smart phones by a blink, it will be big help for them to tell caretakers what they really need or want to tell. First, we try to detect an eye area by using OpenCv. Then we develop the way to detect opening and closing of eyes. We combine the method using situation and using complexity of image to get more accurate results to detect a blink. The level of handicapped is very varied in children.

Date Added to IEEE Xplore: 10 February 2014.

2.7 Wink to grasp” - comparing eye, voice & EMG gesture control of grasp with soft-robotic gloves

The ability of robotic rehabilitation devices to support paralysed end-users is ultimately limited by the degree to which human-machine-interaction is designed to be effective and efficient in translating user intention into robotic action. Specifically, we evaluate the novel possibility of binocular eye-tracking technology to detect voluntary winks from involuntary blink commands, to establish winks as a novel low-latency control signal to trigger robotic action. We compare our novel approach to two conventional approaches for controlling robotic devices based on Electromyo-Graphy (EMG) and speech-based human-computer interaction technology. We use a soft-robotic SEM glove (Bio servo Technologies AB, Sweden) to evaluate how the 3 modalities, support the performance and subjective experience of the end-user when movement assisted. All 3 modalities are evaluated in streaming, closed-loop control operation for grasping physical objects. We find that wink control shows the lowest error rate mean with lowest standard deviation of (0.23 ± 0.07, mean ± SEM) followed by speech control (0.35 ± 0.13) and EMG gesture control (using the Myo armband by Thalamic Labs), with the highest mean and standard deviation (0.46 ± 0.16).

2.8 System and method for detecting gaze direction

We try to develop a support application for physically handicapped children to communicate with others by blinks. First, we try to detect an eye area by using OpenCv. Then we develop the way to detect opening and closing of eyes. We combine the method using saturation and using complexity of image to get more accurate results to detect blinks. Then we develop the technique into a communication application that has the accurate and high-precision blink determination system to detect letters and put them into sound. Then we applied this highly precise blink detection method to developing the non-contact communication support tool which judges the eye direction from ocular movement. We combine eyes direction and blink to choose a letter. This blink determination and gaze detraction detecting made it possible to choose letters remarkably fast. Furthermore, we digitized blurring of the vibration of the centre point of the right and left eyes by the comparison with the afterimage and replaced middle point of the amount of change with scatter diagram and distinguished the state of the subject. We will find association with fatigue degree, sleep shortage and an intensive degree and the blurring of the eyeball vibration of right and left using neural network in future.

Date Added to IEEE Xplore: 21 January 2016.

2.9 Development of communication support application with blinks

In this study, we try to develop a new application for physically handicapped children to communicate with others by a blink. Because of limited body movements and mental disorders, many of them cannot communicate with their families or caregivers. We think if they can use applications of smart phones by a blink, it will be big help for them to tell caretakers what they really need or want to say. First, we try to detect an eye area by using OpenCv. Then we develop the way to detect opening and closing of eyes. We combine two methods, using saturation and using complexity of image, to get more accurate results of detecting a blink.

Date Added to IEEE Xplore: 18 August 2014.

2.10 Eye-Fi: integrating optical data communication into intraocular lines

The human eye is a compelling host site for wearable health-care electronics. Smart intraocular lenses can be used both to receive external optical data and to convey information to the user, along with monitoring one's health indicators and transmitting the results to a central unit. This is coined as Eye-Fi technology and can be deployed to a lens where commands to an embedded microelectronics chip are modulated either in a simple pattern of eye blinks, or in the light itself. In this paper, we demonstrate an integrated circuit meant to be implanted in an IOL, as a first stage to enable the lens to later perform intraocular health monitoring of the user.

Date of Conference: 8-9 Feb. 2018

2.11 Eye Blink Detection for the Implantable System for Functional Restoration of Orbicularis Oculi Muscle

In this work, the blink detection algorithm for an embedded closed-loop system for a restoration of facial muscle functions in the case of unilateral facial nerve paralysis is proposed. The algorithm is based on a real-time assessment of a spectral power of the electromyographic signal, which is recorded from the orbicularis oculi muscle with implanted needle electrodes on the healthy side of the patient's face. Once a blink is detected, the stimulation command is sent to the module of the functional electrical stimulation, which elicits the contraction of muscles on the paralyzed side of the face. The algorithm was tested on the electromyographic signals, which were recorded from the orbicularis oculi muscle of the healthy subject performing daily routine activity tasks (head movements, vocalization, etc). The resulting 93 % sensitivity and 63 % precision values are obtained.

Date of Conference: 12-14 Sept. 2017

2.12 Efficient Eye Blink Detection Method for disabled helping domain

Facial paralysis makes patients lose their facial movements, which can incur eye damage even blindness since patients are incapable of blinking. We design and implement a pair of smart glasses iBlink to assist facial paralysis patients to blink. Our contributions are: First, we propose an eye-blink detection mechanism based on support vector machine (SVM), which can detect asymmetric blinks of patients under various illumination conditions with an accuracy above 99%. Our eye-image library for training the model is published online for further related studies, which contains more than 30,000 eye images. Second, we design and implement an automatic stimulation circuits to generate electrical impulse for stimulating the patient's facial nerve branches, which can configure operational parameters in a self-adaptive manner for different patients. Third, we implement the entire iBlink system, which integrates the two functions above and a communication function module for tele-medicine applications.

Date of Publication: 03 September 2018

2.13 Towards a wearable device for controlling a smartphone with eye winks

The development of mobile technology over the last years and the consequent boom of available apps has enabled users to migrate a wide range of activities that were traditionally performed on computers to their smartphones. Even though there are voice-control alternatives for operating smartphones, these do not perform well in crowded or noisy environments. In this paper we present Eyewink: an innovative hand- and voice-free wearable device that allows users to operate the smartphones with eye winks. The system records the Electrooculography (EOG) signals on the forehead by means of two facial electrodes. Eye winks are detected by comparing the potentials recorded from the electrodes, which also helps avoid false actuations due to (unavoidable) eye blinks. The user can associate the action to perform with each eye by means of an app installed on the smartphone. The proposed device can be widely used, with customers ranging from runners to people with severe disabilities.

Date Added to IEEE Xplore: 23 November 2015

CHAPTER 3

REQUIRIMENT SPECIFICATIONS

CHAPTER 3

REQUIREMENTS SPECIFICATIONS

3.1 HARDWARE REQUIREMENT

- **Camera Installation**

The software detects the action of eyes with the help of camera. So, the computer needs to be installed with camera. The laptop is equipped with camera, but the desktop computer sometimes doesn't have camera, so an extra camera needs to be installed and linked to the computer.

3.1.1 iBall super-view C8.0

A famous iBall Super-View C8.0 web camera with interpolated 8.0MP Still Image resolution, 4.0MP Video resolution and 5G Wide angle lens provides smooth video and lets you enjoy the clarity of web video.

- High quality still pictures and motion video capture.
- Image sensor: high quality ¼ inch CMOS sensor.
- Effective pixels: 480k pixels (interpolated 8M pixels still image and 4M pixels video).
- High quality 5G wide angle lens for sharp and clear picture.
- 4 LEDs for night vision, with brightness controller.
- LED panel with 12 LEDs and on/Off switch for reading or working at night.
- Highly sensitive USB microphone.
- 4x Digital zoom and Motion tracking.
- 10 Photo frames and 136 Special effects for more fun.
- OS Compatibility : Windows XP / Windows Vista & Windows 7 and above.

Fig: 3.1.1 iBall super-view C8.0

3.2 SOFTWARE REQUIREMENT

3.2.1 OPEN CV

It is a library of programming functions mainly aimed real-time computer vision. Originally developed by Intel and it was later supported by
WILLOW GARAGE and is now maintained by IITSSEZ. The library is the cross platform and free for use under the open source BSD license.
Open CV is written in C++ and it is primarily interface is in C++. There are bindings in python, java and MATLAB/Octave.
Open CV application areas include:

- 2D and 3D features toolkits
- Ego motion estimation
- Facial recognition system
- Gesture Recognition
- Human Computer Interaction
- Mobile Robot
- Segmentation and Recognition
- Augmented reality and Motion Tracking

- **Installation and Usage**
  - run pip install OpenCV-python
  - run pip install OpenCV-contrib-python

- **Import the Packages for OpenCv**

Open Python IDLE and type following codes in Python terminal.

- >>> import cv2
- >>> print(cv. version__)

Fig: 3.2.1 Open CV

3.2.2 PYTHON 3.5.0

It is an interpreted high-level programming language for general purpose programming. Python has a design philosophy that emphasizes code readability and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant white space. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming, paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python Interpreters are available for many OS. Mostly Python implementation includes a read-eval-print loop, permitting to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Some things that Python is often used for are:

- Web development.
- Scientific programming.
- Desktop GUIs.
- Network programming.
- Game programming.

Fig 3.2.2 Python

This is a small example of a Python program. It shows "Hello World!" on the screen.

```
print("Hello World!")
# This code does the same thing, only it is longer:
ready = True
if ready:
print ("Hello World!")
```

- **python dlib**

Developed by Davis King, the dlib C++ library is a cross-platform package for threading, networking, numerical operations, machine learning, computer vision, and compression, placing a strong emphasis on extremely high-quality and portable code.

From a computer vision perspective, dlib has several state-of-the-art implementations, including:

- Facial landmark detection
- Correlation tracking
- Deep metric learning

- **Installation and Usage**

  - **c:\python35\site-packages>pip install** dlib-18.17.100-cp27-none-win32.whl

wheel (whl)

A WHL file is a package saved in the Wheel format, which is the standard built-package format used for Python distributions. It contains all the files for a Python install and metadata, which includes the version of the wheel implementation and specification used to package it. WHL files are compressed using Zip compression.

- **Import the Packages for Dlib**

Open Python IDLE and type following codes in Python terminal.

- >>> import dlib
- >>> dlib._ version__

3.2.3 pygame

pygame (**the library**) is a Free and Open Source python programming language library for making multimedia applications like games built on top of the excellent SDL library. Like SDL, pygame is highly portable and runs on nearly every platform and operating system.

- **Installation and Usage**
  - Pip install pygame

- **Import the Packages for pygame**

Open Python IDLE and type following codes in Python terminal.

- >>> import pygame

pyInstaller

PyInstaller bundles a Python application and all its dependencies into a single package. The user can run the packaged app without installing a Python interpreter or anymodules. PyInstaller supports Python 2.7 and Python 3.3+, and correctly bundles the major Python packages such as NumPy, PyQt, Django, wxPython, and others.

- **Installation and Usage**
  - Pip install pyInstaller

- **Import the Packages for pyInstaller**

Open Python IDLE and type following codes in Python terminal.

- >>> import pyInstaller

3.2.4 Twilio API

The Twilio REST API allows you to query metadata about your account, phone numbers, calls, text messages, and recordings.

- **Import the Packages for twilio**

Open Python IDLE and type following codes in Python terminal.

- >>> import pyInstaller

- **Account for Twilio Sample Code**

```
account_sid = "AC9b355d488ffe644e4c0c2a21e43a42c7"
            auth_token = "5dcd0305122226138f44b9c0493afcc5"
            client = Client(account_sid, auth_token)
            client.api.account.messages.create(
   to="+91990020030",
   from_="+12028833955",
   body="Emergency Hurry up !")
```

CHAPTER 4
DESIGN

CHAPTER 4

DESIGN

4.1 WorkFlow Diagram

The design part includes the data flow diagram. The data flow diagram explains the workflow of the system proposed. The work flow mainly highlights the data flow direction that means how the data is being modified? How it is being used? and how the results vary with it?

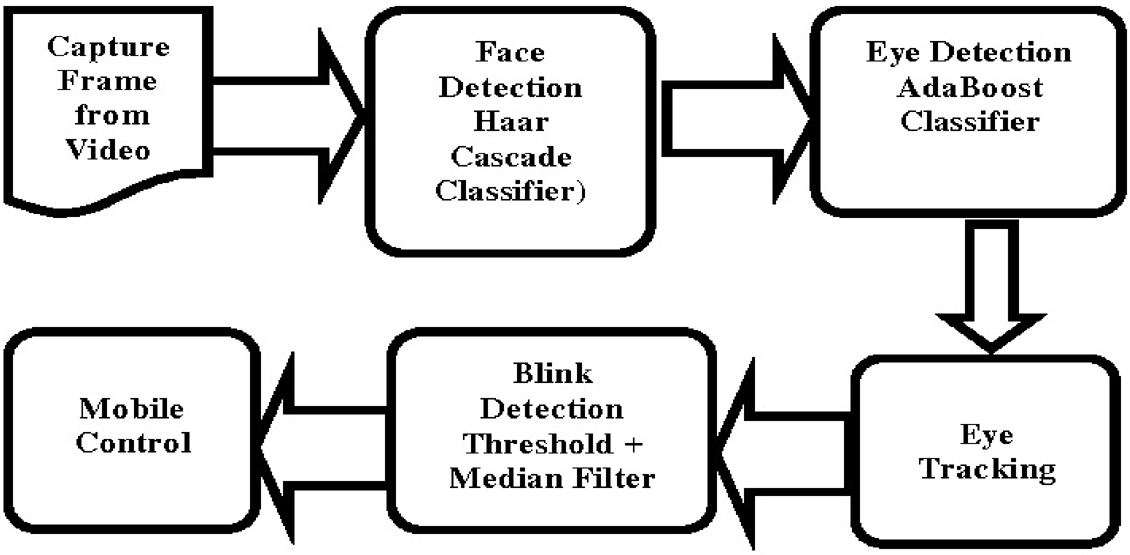


Fig 4.1 Workflow diagram

- Capturing the frame from the video using the system's camera initialises the execution of the proposed system.
- The Face Detection Algorithm then processes on the captured video frames to give out the rectangular boxed face. This output from Face Detection Algorithm then gets processed using AdaBoost Classifier to detect the eye region in the face.
- Eye detected will be sent to check if there is any movement of eyeball.
- If it's there, then this movement will be tracked to give out the combination the patient is using to express the dialogue.
- If not, then the blink pattern will be processed to give out the voice as well as the text input with respective dialogue.

4.2 Face Detection and Eye Detection Flow Chart

4.2.1 Frame Capturing

The first step of the proposed EBCM application is the initialization. After taking a short video of the participant's face using the front camera of Laptop/External Camera. A process Frame method will be used to create the frames from the captured video.

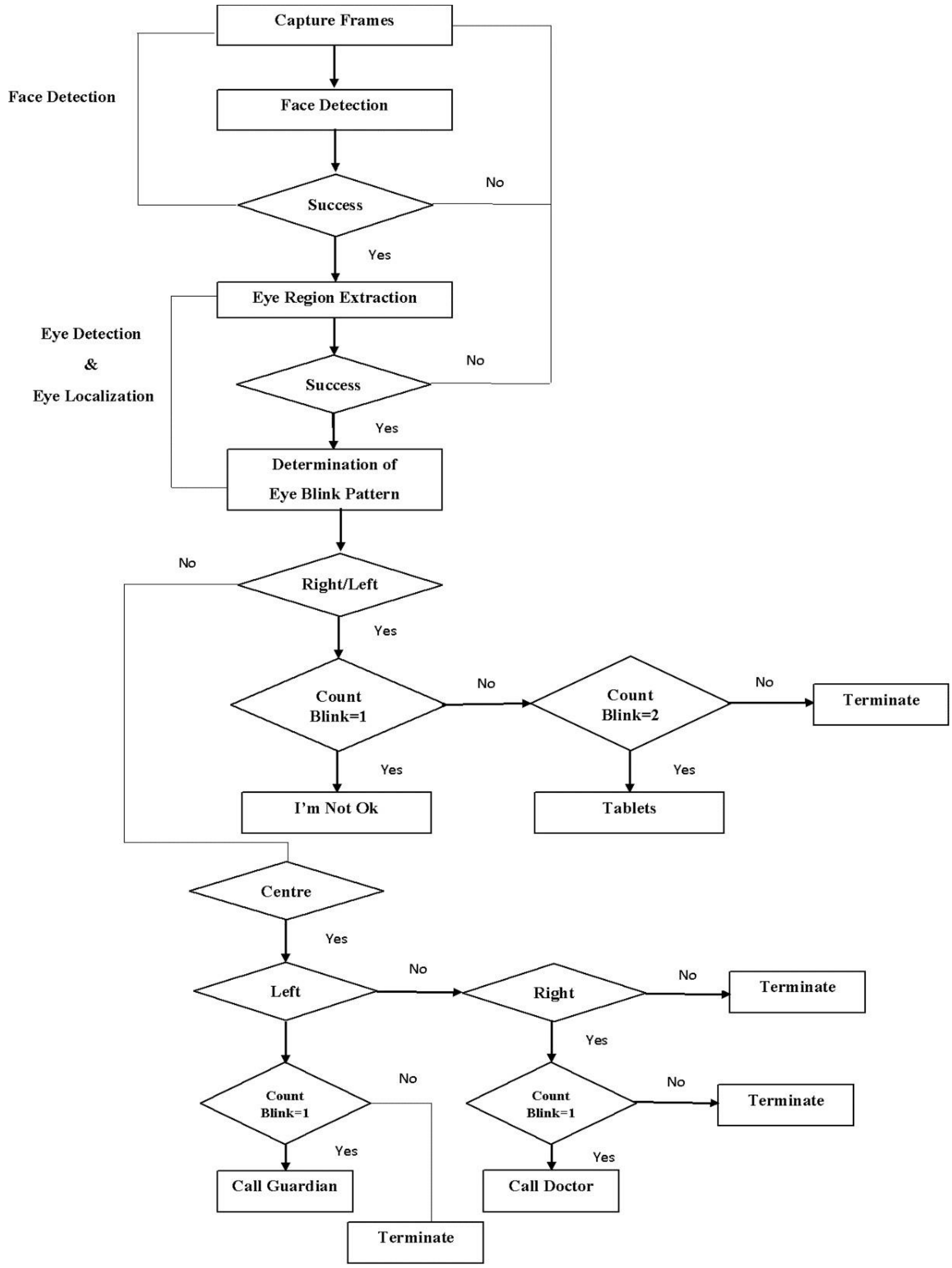


Fig 4.2 Flow Chart

Fig. 4.2 shows how the system operates. The following are the steps from start to end of the Eye Blinking software.

Open the webcam/ External Camera and capture the video.

1. The system performs an action and detects the face.
2. The system performs eyes detection.
3. The system detects the eye pupil moment and eye Blink.
4. With only the image of face from the webcam/External Camera, the system will locate the eyes and perform geometry translations or output.

4.3 Use Case Diagram

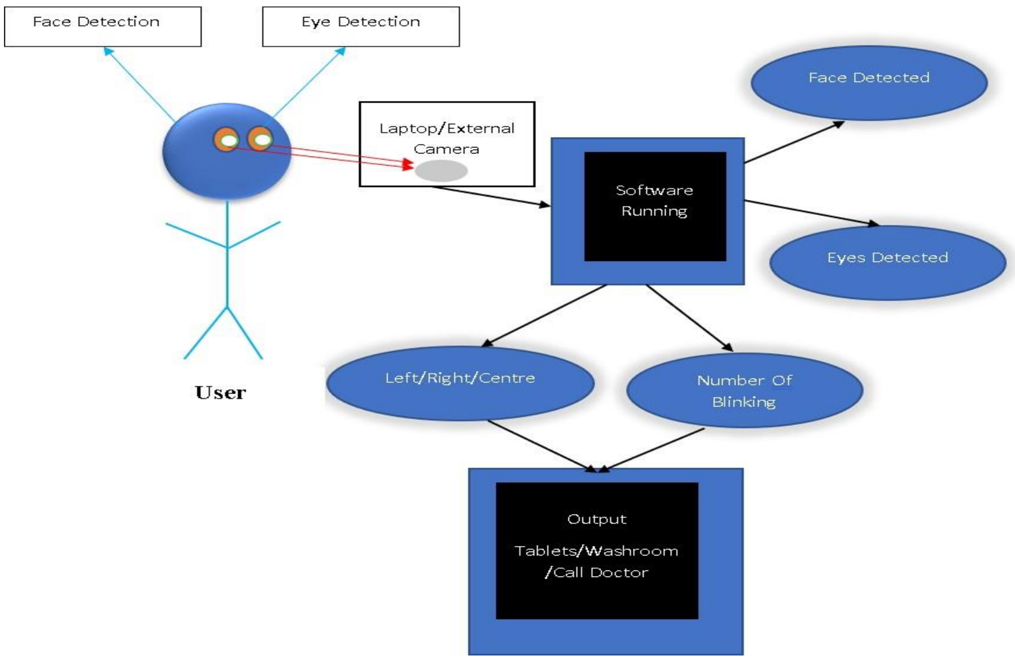


Fig 4.3 Use Case Diagram

In Fig 4.3 shows use case diagram; the system accomplishes to have the following steps.

1. Open the webcam on the laptop and show the image of a person.
2. Face detection action is performed.
3. The system detects the eyes of a person.
4. After the above action system move on to the next operation.
5. In the next step the system detects eyes and face through webcam of a laptop.
6. The blinking is detected with Right eye and left eye respectively, then display the respected output.

4.4 Sequence Diagram

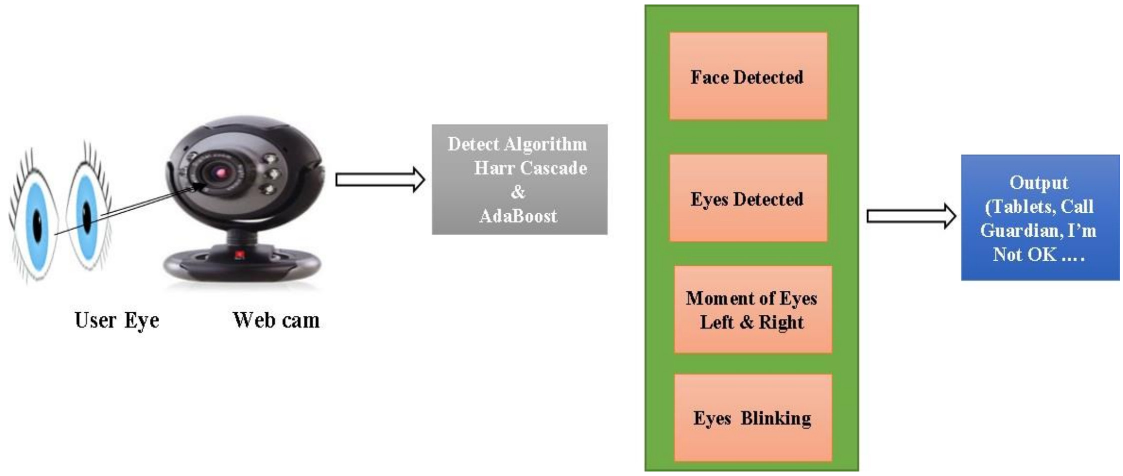


Fig 4.4 Sequence Diagram

In Fig. 4.4 the system sequence diagram, elaborates the six basic modules of our system. In the first module, the system detects the face of a person through the webcam using detection algorithms (Haar Cascade). Then the system detects the Eyes (AdaBoost). After that the system detects and captures the eyes moments (Centre, Left, Right). Then the system detects the Blinking (Count Blink). In the last module, the system starts displaying a message depend upon the moments of eye and number of blinking.

Example :

```
if right ==1 and left ==1 and blink==5:
        right =0
        left =0
        blink=0
        print('I am Not Ok ')
        service=1
    elif centre ==1 and right ==0 and left ==1 and blink==1:
        right =0
        left =0
        blink=0
        print('Call Guardian')
```

4.5 System Functionality Diagram

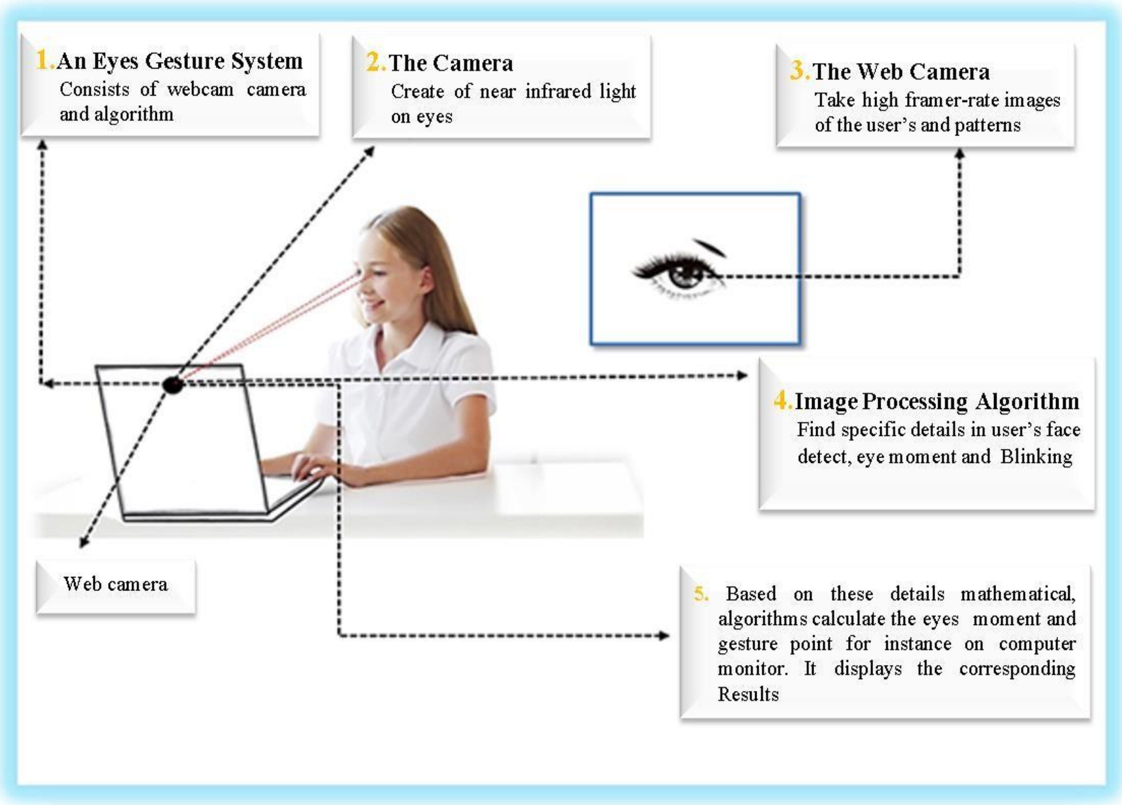


Fig 4.5 System Functionality Diagram

CHAPTER 5

IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

Methodology

1. Camera and frame capturing
2. Face Landmark
3. Eye detection
4. Eye tracking
5. Eye blinking

5.1 CAMERA AND FRAME CAPTURING

The first step of the proposed system is the initialization. After taking a short video of the participant’s face using the front camera of the device used which is more likely a laptop. A process Frame method will be used to create the frames from the captured video. Afterwards the coloured frames will be converted to gray scale frames by extracting only the luminance component.

The luminosity method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. We’re more sensitive to green than other colours, so green is weighted most heavily. The formula for luminosity is 0.21 R + 0.72 G + 0.07 B. The luminosity method works best overall.

Fig5.1: Original Image **Grayscale Image**

5.2 FACE DETECTION

The Haar classifier is used in algorithm for face detection. . Haar classifier rapidly detects any object, based on detected feature not pixels, like facial feature. However, the area of the image being analysed for a facial feature needs to be regionalized to the location with the highest probability of containing the feature. By regionalizing the detection area, false positives are eliminated. As the result face is detected and marked with rectangle box and will be used later to approximate an axis of the eyes for eye detection step.

5.2.1 Facial Landmark Algorithm

Detecting facial landmarks is a subset of the shape prediction problem. Given an input image (and normally an ROI that specifies the object of interest), a shape predictor attempts to localize key points of interest along the shape. They are hence important for various facial analysis tasks. Regionalising the face is done by using Facial Landmark Algorithm.

In the context of facial landmarks, our goal is detecting important facial structures on the face using shape prediction methods. The steps involved in calling the Facemark

Detecting facial landmarks is therefore a twostep process:

- Step 1: Localize the face in the image.
- Step 2: Detect the key facial structures on the face ROI

Face detection (Step 1) can be achieved in several ways. We could use OpenCV's built-in **Haar cascades.**

5.2.2 Haar Cascade Classifier

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video.

The algorithm has four stages:

- Haar Feature Selection
- Creating Integral Image
- Adaboost Training
- Cascading Classifiers

It is well known for being able to detect faces and body parts in an image but can be trained to identify almost any object. The algorithm needs a lot of positive images of faces and negative images without faces to train the classifier. Then we need to extract features from it. First step is to collect the Haar Features. A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.

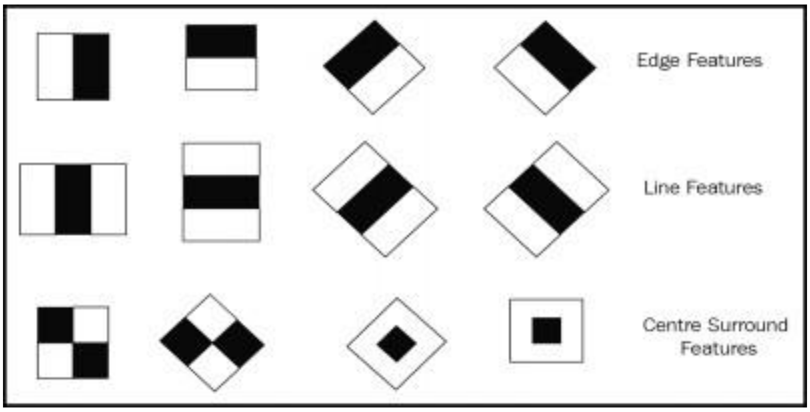


Fig 5.2.1 Rectangular masks used for edge feature (Haar feature) object detection.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant.

Selecting the best feature out of 160000+ features is accomplished by using AdaBoost which both selects the best features and trains the classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers. During the detection phase, a window of the target size is moved over the input image, and for each subsection of the image and Haar features are calculated. Because each Haar feature is only a "weak classifier" (its detection quality is slightly better than random guessing) many Haar features are necessary to describe an object with sufficient accuracy and are therefore organized into cascade **classifiers** to form a strong classifier.

The cascade classifier consists of a collection of stages, where each stage is an ensemble of weak learners. The weak learners are simple classifiers called decision stumps. Each stage is trained using a technique called boosting. Boosting provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

Conversely, true positives are rare and worth taking the time to verify.

- A true positive occurs when a positive sample is correctly classified.
- A false positive occurs when a negative sample is mistakenly classified as positive.
- A false negative occurs when a positive sample is mistakenly classified as negative.

Cascade classifier training requires a set of positive samples and a set of negative images. You must provide a set of positive images with regions of interest specified to be used as positive samples. You can use the Image Labeller to label objects of interest with bounding boxes.

5.2.3 Detecting key facial structures in the face region.

There are a variety of facial landmark detectors, but all methods essentially try to localize and label the following facial regions:

- Face part extraction (i.e., nose, eyes, mouth, jawline, etc.)
- Facial alignment
- Head pose estimation
- Face swapping
- Blink detection

The facial landmark detector included in the dlib library is an implementation of the One Millisecond Face Alignment with an Ensemble of Regression Trees

This method starts by using:

- A training set of labelled facial landmarks on an image. These images are manually labelled, specifying specific (x, y)-coordinates of regions surrounding each facial structure.
- Priors, of more specifically, the probability on distance between pairs of input pixels.

5.2.4 Facial landmark visualizations

Before we test our facial landmark detector, make sure you have upgraded to the latest version of imutils which includes the face_utils.py file:

Facial landmarks with dlib, OpenCV, and Python

- pip install --upgrade imutils

Sample Code for Shape Predictor

```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
def midpoint(p1 ,p2):
 return int((p1.x + p2.x)/2), int((p1.y + p2.y)/2)
font = cv2.FONT_HERSHEY_PLAIN
```

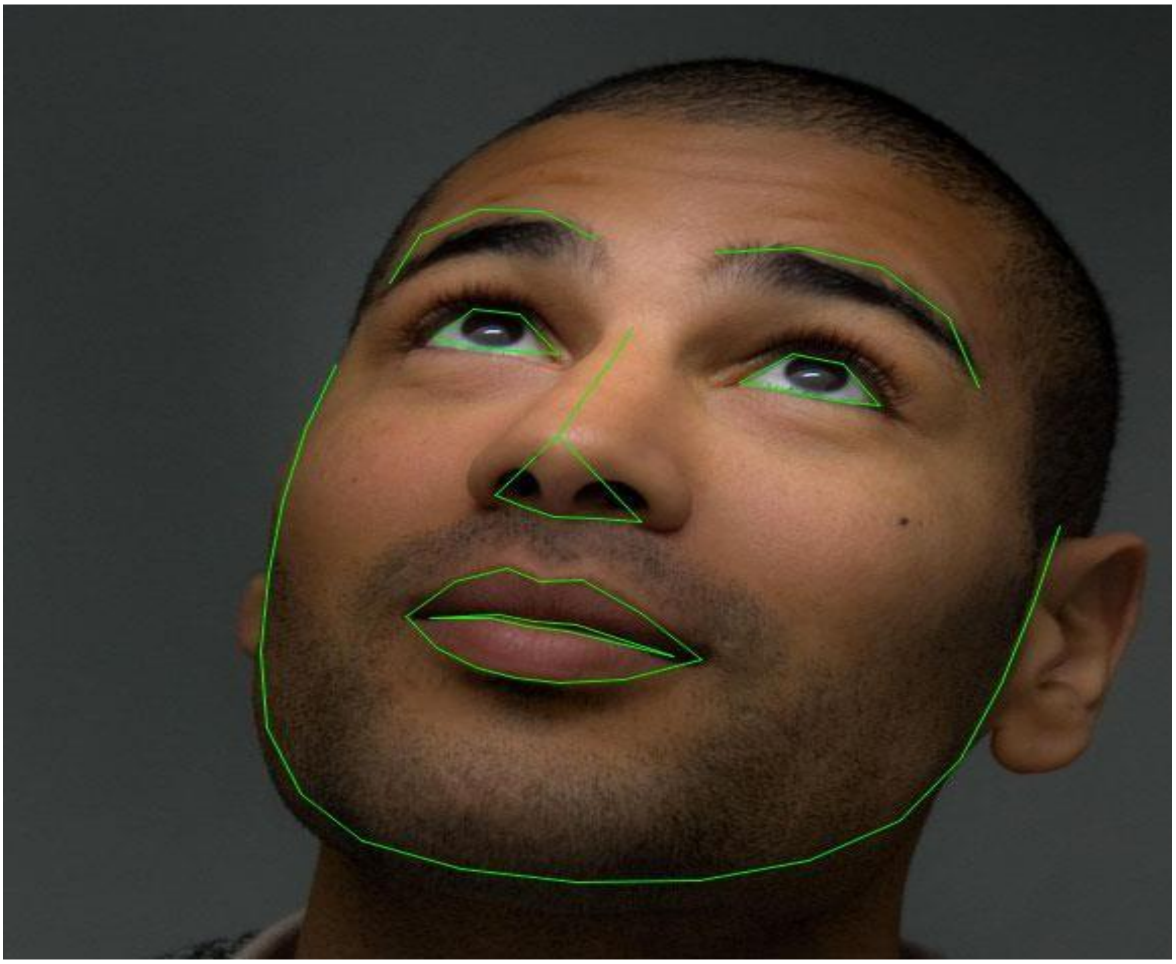


Fig 5.2 Face Detection

5.3 EYE DETECTION

To detect the eye, first, the Haar cascade classifier should be trained, in order to train the classifiers, the AdaBoost algorithm and Haar feature algorithms must be implemented, two set of images are needed. One set contains an image or scene that does not contain the object. The EBCM used all detected elements of Haar Cascade Classifier, and the result show the detected eye in rectangle box.

AdaBoost algorithm is used to train node classifiers on a Haar-like feature set to improve the generalization ability of the node classifier. Consequently, the face detection performance of the face detector is improved. Experimental results have proved that the proposed algorithm can significantly reduce the number of weak classifiers, increase the detection speed, and slightly raise the detection accuracy as well.

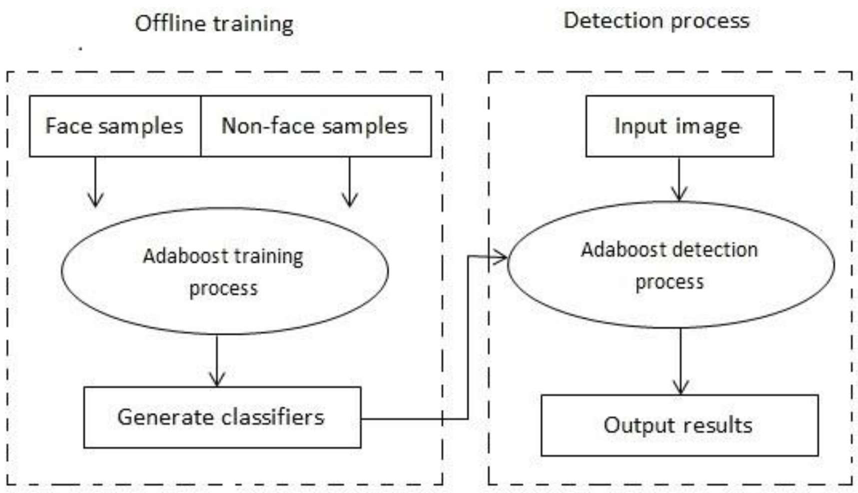


Fig 5.3 The flow chart of face detection based on Adaboost algorithm.

Fig 5.3 Eye Detector

5.4 EYE PUPIL TRACKING

The corneal-reflection and pupil-centre are the two eye's parts that are the most important parts to extract the features that will be used in EBCM method. These features help us in tracking the eyes movement. By identifying the center of the pupil and the location of the corneal reflection, the vector between them is measured. Besides, with further trigonometric calculations, point-of-regard can be found. The EBCM method succeeded in making the face and the eye's pupil moved together in the same direction synchronously and with the same direction. Let suppose that X is the human face which has been detected , P1 and P2 are two points related to the left eye, and they are moving synchronously with the movement of X.

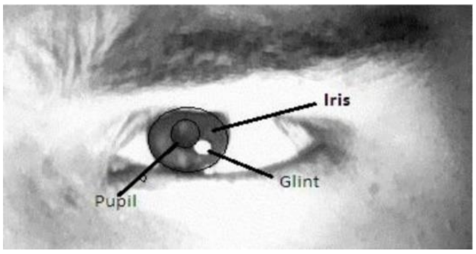


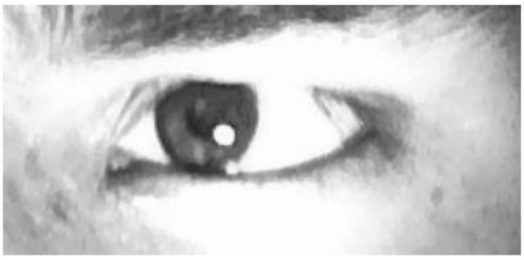
Fig 5.4 Parts of Eye

5.4.1 Initial Approach

Extraction of pupil and glint.

First, head is detected using haar cascades and location of eye is localised.

Glint extraction is fairly simple as it is a white speck in black background. Pupil extraction is very difficult is absence of IR light. I tried with various configuration of light and camera . Pupil is visible in cases when light source and camera both are kept very close to eye and illumination level of room is low (Fig.5.4.1).



Pupil in Dark room **pupil during Daytime**

Fig 5.4.1 Pupil in different configuration

5.4.2 Final Approach:

Extraction of glint and using Iris as an approximation for pupil. Step1. Image is converted to grayscale and Head is detected using Haar cascades. Image is cropped around approximate location of eye using formula :

Origin cropped image = Originhead + (0, Heighthead / 5.5) WidthcroppedImage = Widthhead , HeightcroppedImage = Heighthead / 3.0;

This approach was chosen instead of direct eye detection because, eye detection using Haar cascades is not very accurate, resulting in eyebrows being detected as eyes often.

5.5 EYE BLINKING

Eye blinking and movement can be detected with relatively high reliability by unobtrusive techniques. Though, there are few techniques discovered for the active scene where the face and the camera device move independently, and the eye moves freely in every direction independently of the face. Although care must be taken, that eye-gaze tracking data is used in a sensible way, since the nature of human eye movements is a combination of several voluntary and involuntary cognitive processes.

5.5.1 EYE ASPECT RATIO" (EAR)

we can apply facial landmark detection to localize important regions of the face, including eyes, eyebrows, nose, ears, and mouth. This also implies that we can extract specific facial structures by knowing the indexes of the particular face parts

In terms of blink detection, we are only interested in two sets of facial structures — the eyes. Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region. Based on this image, we should take away on key point. There is a relation between the width and the height of these coordinates Where p1, …, p6 are 2D facial landmark locations.

$$\text{EAR} = \frac{\| p_2 - p_6 \| + \| p_3 - p}{2 \| p_1 - p_4 \|}$$

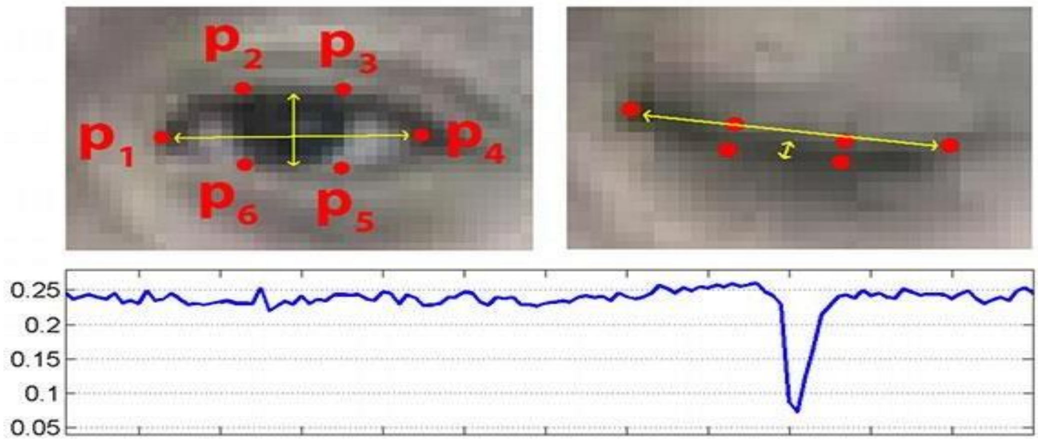


Fig 5.5.1 Eye Accept Ratio

CHAPTER 6
TESTING

CHAPTER 6

TESTING

6.1 Introduction to testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects) and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided its design and development.
- Responds correctly to all kinds of inputs.
- Performs its functions within an acceptable time.
- It is sufficiently usable.
- Can be installed and run in its intended environments, and
- Achieves the general result its stakeholders desire.

The aim of testing stage is to discover defects or errors by testing individual program components. These components may be functions, objects or modules. During system testing, these components are integrated to form the complete system. At this stage, testing should focus on establishing that the system meets its functional requirements and does not behave in an unexpected way. Test data are inputs which have been devised to test the system whereas test cases are inputs to test the system and the outputs are predicted from these inputs if the system operates according to its specification. This is to examine the behaviour in a cohesive system. The test cases are selected to ensure that the system behaviour can be examined in all possible combinations of conditions.

Accordingly, expected behaviour of the system under different combinations is given. Therefore, test cases are selected which have inputs and the outputs are on expected lines, inputs that are not valid and for which suitable messages must be given and inputs that do not occur frequently which can be regarded as special cases.
In this chapter, several test cases have been explained of project Fast clustering application.

6.1.1 Test Environment

The software was tested on the following platform.

- **Hardware**
  - ➢ 1000 GB Hard Disk, 8 GB RAM with 1.84 GHz Dual Core processor.
  - ➢ Web Camera
  - ➢ External Camera: iball super-view C8.0

- **Software**
  - ➢ Operating System – Windows 10 (32 bit and 64 bit)
  - ➢ Python3.5.2
  - ➢ Open CV
  - ➢ PyInstaller
  - ➢ Html Server

6.2 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

- **Testing Strategy**

The strategy that is used to perform unit testing is described below:

- **Features to be tested** – The features to be tested, most importantly includes the operation of individual component for the proper execution of the entire program.
- **Items to be tested** – The items to be tested include all the individual units or functions, which collectively form the whole system. This session includes eye blinking, Web camera\external camera, moment of eyes and displaying messages.
- Admin user for approval tasks, Admin user for approval tasks provision for retrieval of expired files.
- **Purpose of testing** – The purpose of the testing is to check the unit functionality of the main project source.
- **Pass/Fail Criteria** – The pass or fail criteria are designed with the basis of appropriate compilation of the main source file.

6.2.1 Unit Testing of Main Project URL Webpage

Sl # Test Case:	**UTC-1**
Name of Test:	Unit Testing of “URL Webpage”
Item being tested:	URL Web Page
Sample Input:	➢ https://yashu41197.wixsite.com/eyeblinktospeak
Expected output:	Successfully Webpage is opened
Actual output:	Same as expected output
Remarks:	Successful

Table 6.2.1 Unit Test Case URL Webpage

6.2.2 Unit Testing of Execution (test1.exe) record URL Webpage

S2 # Test Case:	**UTC-2**
Name of Test:	Unit Testing of "test1.Exe Webpage **Unit Testing of Execution (test1.exe) record URL Webpage**"
Item being tested:	(test1.exe) record URL Webpage
Sample Input:	➢ https://drive.google.com/file/d/16C633ak-clVEAexYxnT6J5oeAD8xeSuX/view
Expected output:	Successfully Webpage is opened and downloading
Actual output:	Same as expected output
Remarks:	Successful

Table 6.2.2 Unit Test Case test1.exe record URL Webpage

6.2.3 Unit Testing of Web camera\External camera enable

S3 # Test Case:	**UTC-3**
Name of Test:	Unit Testing of" Web camera\External camera
Item being tested:	Web camera \External camera working
Sample Input:	➢ **Enable**
Expected output:	Successfully Opening
Actual output:	Same as expected output
Remarks:	Successful

Table 6.2.3 Unit Test Case Web camera\External camera

6.2.4 Unit Testing of Service 1

S4 # Test case	**UTC-4**
Name of Test:	Unit Testing of “I’m Not Ok” Eye Blinking
Item being tested:	To detect the blink combination with right=1, left=1, blink=1
Sample Input:	➢ 5 times Beep Sound
Expected output:	5 times Beep Sound and displaying a message on user screen
Actual output:	Same as expected output
Remarks:	Successful

Table 6.2.4 Unit Test Case I’m Not Ok Service 1

6.2.5 Unit Testing of sending a message caretaker “I’m Not Ok” Service 1

S5 # Test Case:	**UTC-5**
Name of Test:	Unit Testing of “Service 1 “Sending Emergency Message
Item being tested:	to="+917892106671", from_="+12028833955", body="Emergency Hurry up!")
Sample Input: -	➢ Emergency Hurry Up
Expected output:	Message Sent
Actual output: -	Same as expected output
Remarks:	Successful

Table 6.2.5 Unit Test Case Sending a message to Caretaker Service 1

6.2.6 Unit Testing of Service 2

S6 # Test Case:	**UTC-6**
Name of Test:	Unit Testing of "Call Guardian" Eye Blinking
Item being tested: -	To detect the blink combination with entre=1, left=1, blink=1
Sample Input: -	➢ 5 times Beep Sound
Expected output:	5 times Beep Sound and Displaying a message on user Screen
Actual output:	Same as expected output
Remarks:	Successful

Table 6.2.6 Unit Test Case Call Guardian Service 2

6.2.7 Unit Testing of Service 3

S7 # Test Case:	**UTC-7**
Name of Test:	Unit Testing of "Tablets" Voice Message
Item being tested:	To detect the blink combination with "blink=1"
Sample Input:	➢ **Voice message Tablets.mp3**
Expected output:	Voice message Tablets.mp3 and Displaying a message on user Screen
Actual output:	Same as expected output
Remarks:	Successful

Table 6.2.7 Unit Test Case Tablets Service 3

6.2.8 Unit Testing of Service 4

S8 # Test Case:	**UTC-8**
Name of Test:	Unit Testing of “Washroom” Voice Message
Item being tested:	To detect the blink combination with “centre=1, blink=2”
Sample Input:	➢ **Voice message Washroom.mp3**
Expected output:	Voice message Washroom.mp3 and Displaying a message on user screen
Actual output: -	Same as expected output
Remarks:	Successful

Table 6.2.8 Unit Test Case Washroom Service 4

6.2.9 Unit Testing of Service 2

S9 # Test Case:	**UTC-9**
Name of Test:	Unit Testing of “Service 2 “Sending Emergency Message
Item being tested:	to="+917892106671", from_="+12028833955", body="Emergency Hurry up!")
Sample Input:	➢ Emergency Hurry Up
Expected output:	Message Sent
Actual output:	Message sending is failed “Server lost”
Remarks:	Unsuccessful

Table 6.2.9 Unit Test Case Call Guardian Service 2

6.3 Integration Testing

Data can be lost across an interface: one module can have an adverse effect on another's sub functions, when combined may not produce the desired major function; global data structures can present problems. Integration testing was a symmetric technique for constructing the program structure while at the same time conducting tests to uncover errors associated with the interface. All modules are combined in this testing step. Then the entire program was tested.

- **Approach**

Some different types of integration testing are big-bang, mixed (sandwich), risky-hardest, top-down, and bottom-up. Other Integration Patterns are collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher-level components.

- **Testing Strategy**

The strategy that is used to perform Integration testing is described below:

➢ **Features to be tested** – The features to be tested are majorly the integration of two or more than two components.

➢ **Items to be tested** – The items to be tested includes web pages, web camera, blinking and moment of eyes.

➢ **Purpose of testing** – The purpose of this integration testing is to scrutiny the functional modularity between all the modules considered in the respective test-cases mentioned below.

➢ **Pass/Fail Criteria** – The pass or fail criteria of this type of testing is based on the proper debugging of only the respective modules files related to the mentioned test cases.

6.3.1 Integration Testing of Main Project URL Webpage

Sl # Test Case:	**ITC-1**
Name of Test:	Integration Testing of “URL Webpage”
Item being tested:	URL Web Page
Sample Input:	➢ https://yashu41197.wixsite.com/eyeblinktospeak
Expected output:	Successfully Webpage is opened
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.1 Integration Testing URL Webpage

6. 3.2 Integration Testing of Execution (test1.exe) record URL Webpage

S2 # Test Case:	**ITC-2**
Name of Test:	Integration Testing of “test1.Exe Webpage **Unit Testing of Execution (test1.exe) record URL Webpage**”
Item being tested:	(test1.exe) record URL Webpage
Sample Input:	➢ https://drive.google.com/file/d/16C633ak-clVEAexYxnT6J5oeAD8xeSuX/view
Expected output:	Successfully Webpage is opened and downloading
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.2 Integration Testing test1.exe record URL Webpage

6. 3.3 Integration Testing of Web camera\External camera enable

S3 # Test Case:	**ITC-3**
Name of Test:	Integration Testing of" Web camera\External camera
Item being tested:	Web camera \External camera working
Sample Input:	➢ **Enable**
Expected output:	Successfully Opening
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.3 Integration Testing Case Web camera\External camera

6. 3.4 Integration Testing of Service 1

S4 # Test Case:	**ITC-4**
Name of Test:	Integration Testing of "I'm Not Ok" Eye Blinking
Item being tested:	To detect the blink combination with right=1, left=1, blink=1
Sample Input:	➢ 5 times Beep Sound
Expected output:	5 times Beep Sound and displaying a message on user screen
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.4 Integration Testing Case I'm Not Ok Service 1

6. 3.5 Integration Testing of sending a message caretaker "I'm Not Ok" Service 1

S5 # Test Case:	**ITC-5**
Name of Test:	Integration Testing of "Service 1 "Sending Emergency Message
Item being tested:	to="+917892106671", from_="+12028833955", body="Emergency Hurry up!")
Sample Input:	➢ Emergency Hurry Up
Expected output:	Message Sent
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.5 Integration Testing Case Sending a message to Caretaker Service 1

6. 3.6 Integration Testing of Service 2

S6 # Test Case:	**ITC-6**
Name of Test:	Integration Testing of "Call Guardian" Eye Blinking
Item being tested:	To detect the blink combination with centre=1, left=1, blink=1
Sample Input:	➢ 5 times Beep Sound
Expected output:	5 times Beep Sound and Displaying a message on user Screen
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.6 Integration Testing Case Call Guardian Service 2

6. 3.7 Integration Testing of Service 3

S7 # Test Case:	**ITC-7**
Name of Test:	Integration Testing of “Tablets” Voice Message
Item being tested:	To detect the blink combination with “blink=1”
Sample Input:	➢ **Voice message Tablets.mp3**
Expected output: -	Voice message Tablets.mp3 and Displaying a message on user Screen
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.7 Integration Testing Case Tablets Service 3

6. 3.8 Integration Testing Service 4

S8 # Test Case:	**ITC-8**
Name of Test:	Integration Testing of “Washroom” Voice Message
Item being tested:	To detect the blink combination with “centre=1, blink=2”
Sample Input:	➢ **Voice message Washroom.mp3**
Expected output:	Voice message Washroom.mp3 and Displaying a message on user screen
Actual output:	Same as expected output
Remarks:	Successful

Table 6. 3.8 Integration Testing Case Washroom Service 4

6. 3.9 Integration Testing of Service 2

S9 # Test Case:	**ITC-9**
Name of Test:	Integration Testing of “Service 2“Sending Emergency Message
Item being tested:	to="+917892106671", from_="+12028833955", body="Emergency Hurry up!")
Sample Input:	➢ Emergency Hurry Up
Expected output:	Message Sent
Actual output:	Message sending is failed “Server lost”
Remarks:	Unsuccessful

Table 6.3.9 Integration Testing Call Guardian Service 2

CHAPTER 7
RESULTS

CHAPTER 7

RESULTS

A screen dump, screen capture, screenshot, or print screen is an image taken by a computer to record the visible items displayed on the monitor, television, or another visual output device. Usually this is a digital image taken by the host operating system or software running on the computer, but it can also be a capture made by a camera or a device intercepting the video output of the display.

7.1 Web Page

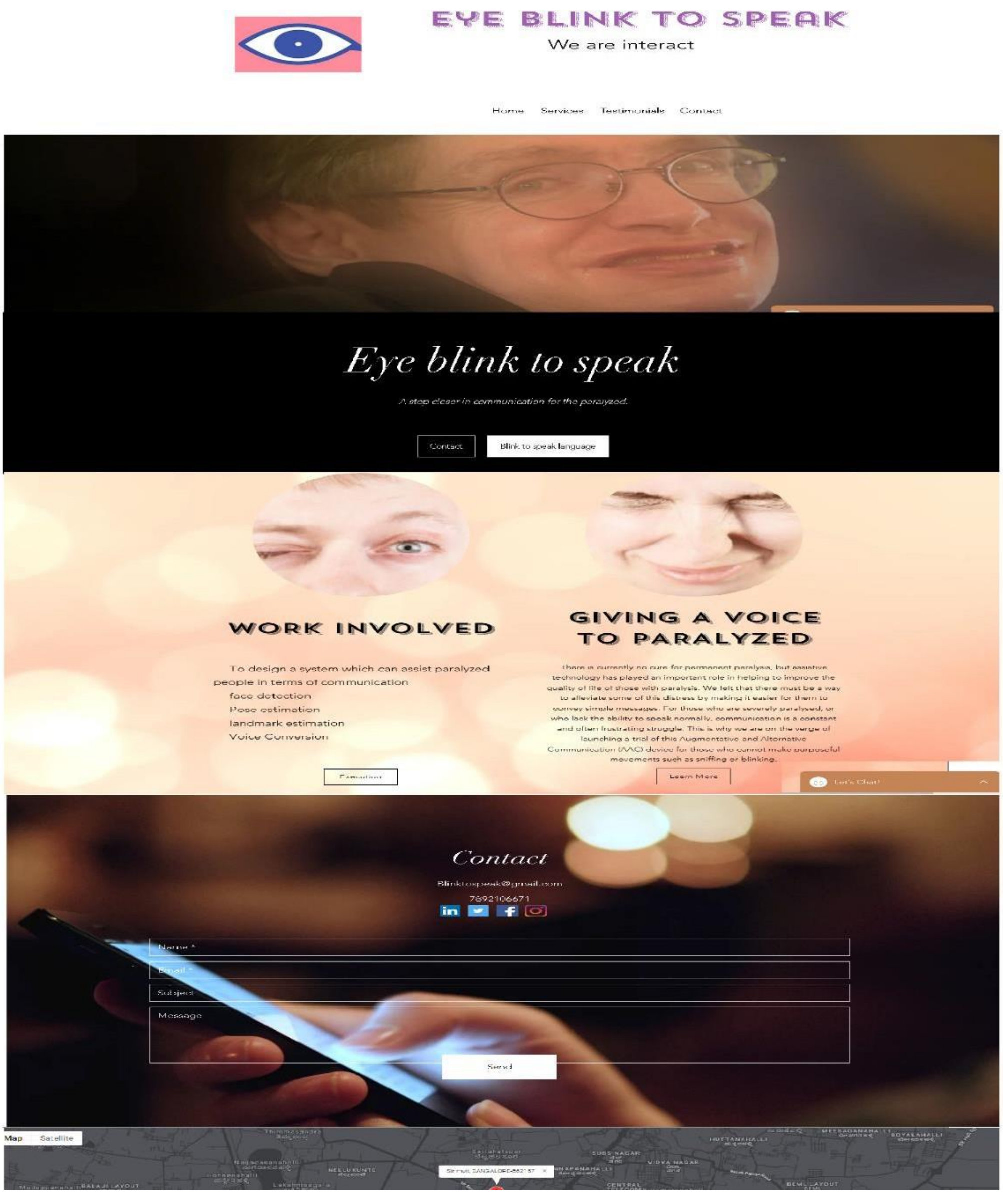


Fig 7.1 Main Webpage

7.2 Sample Training Language image 1

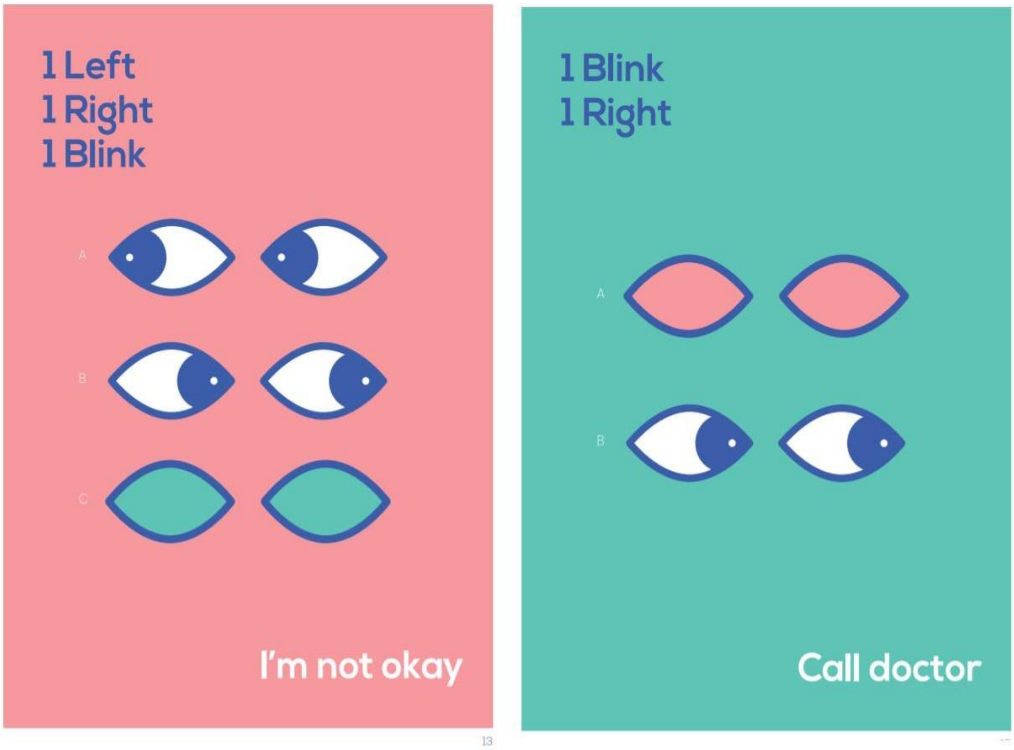


Fig 7.2 Sample Training Language 2

7.3 Sample Training Language image 2

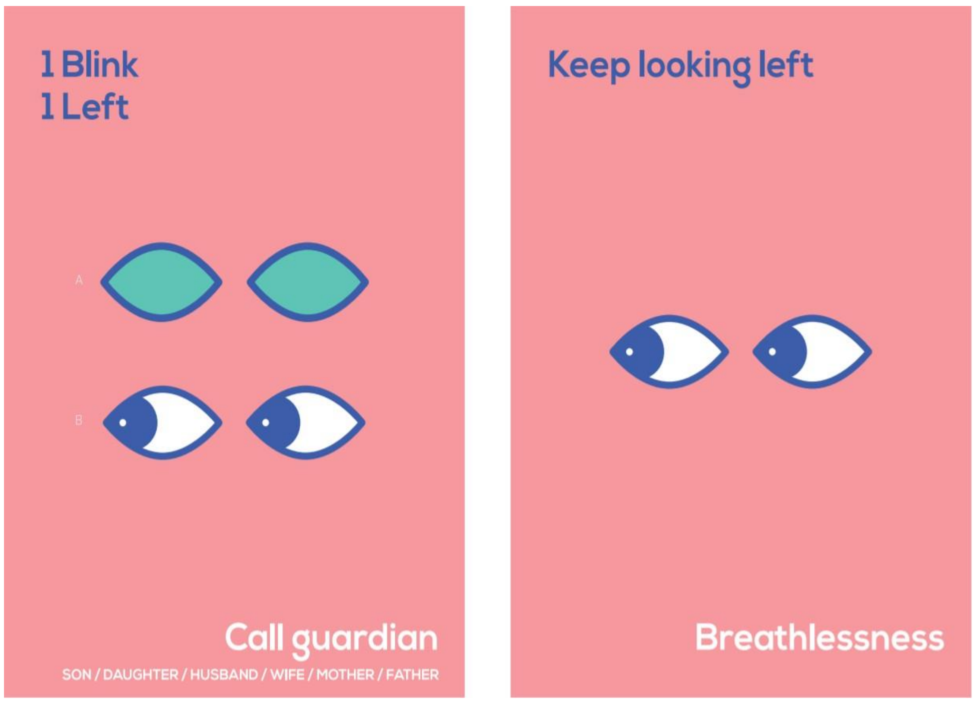
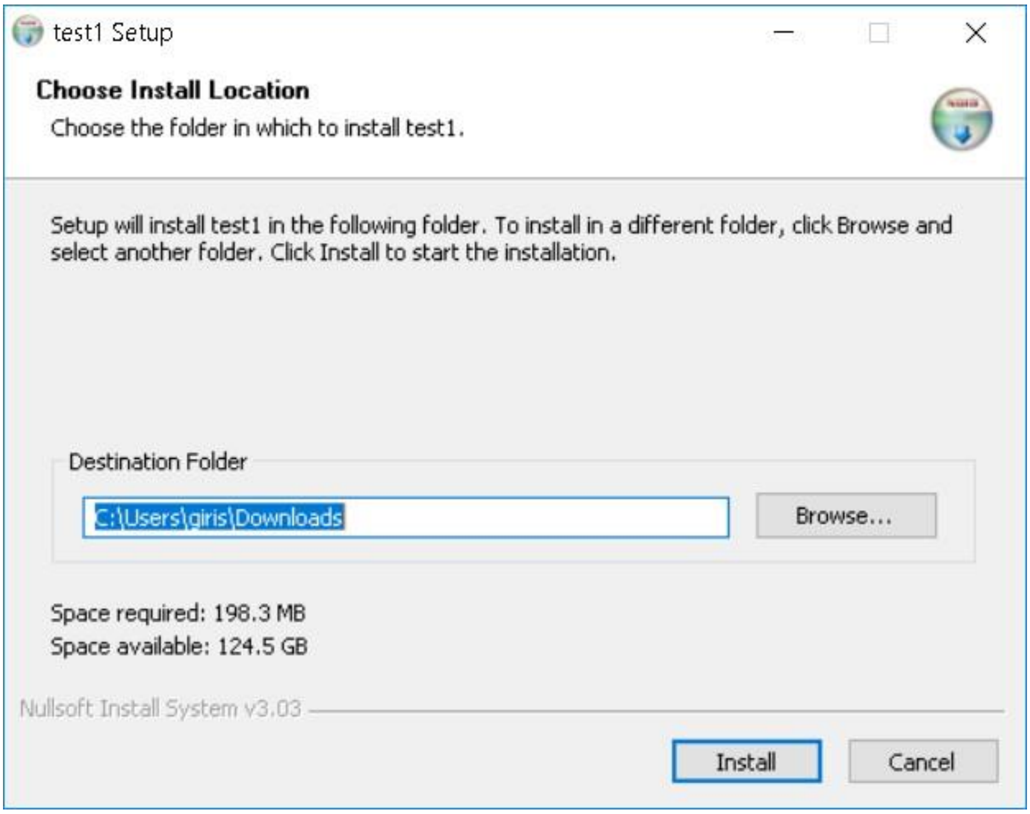


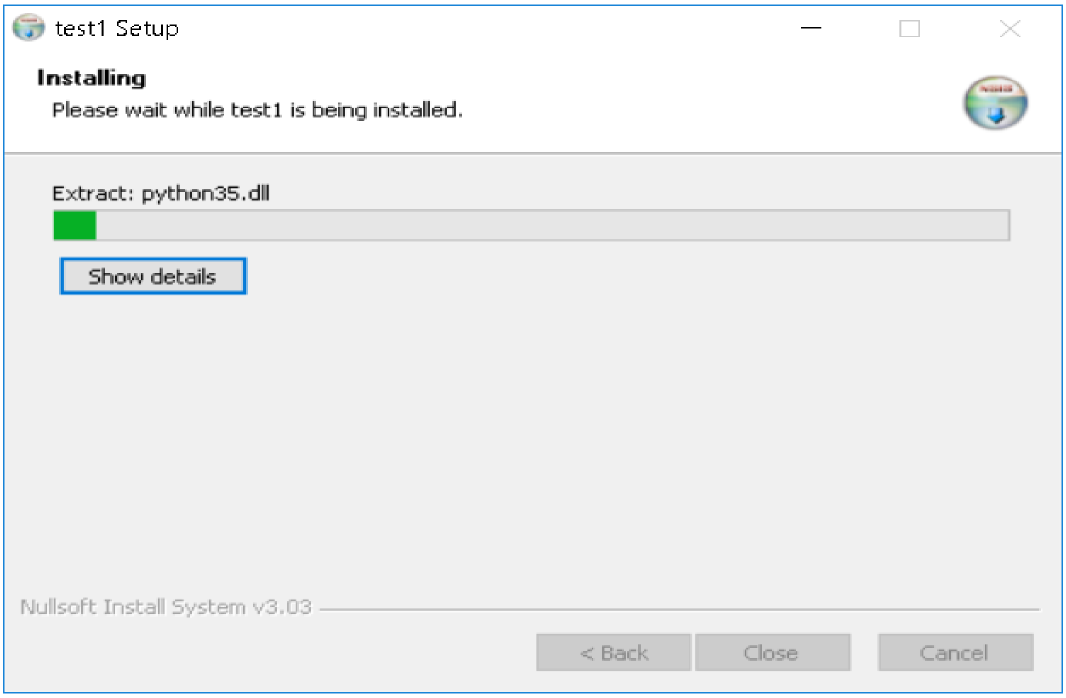
Fig 7.3 Sample Training Language 2

7.4 Eye Blink Setup.exe file



7.4 Setup.exe

7.5 Installation of Eye Blink.exe File



7.5 Installation

7.6 Distance between Web camera and User

```
==================== RESTART: G:\Project\lrcb\test1.py ==================
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
1.7433005590900328
1.6274867374005306
1.7619825708061003
1.4105263157894736
1.4857064321055524
1.360764042459089
1.392082493777409
1.4667571234735415
1.4470264867566216
1.4763888888888888
1.241432178932179
2.095779220779221
4.0
5.0
8.75
14.25
5.0
3.0
3.0
5.0
3.0
3.0
2.5625
40.0
2.5
1.0
Water
0.537037037037037
1.4444444444444444
2.8214285714285716
0.5666666666666667
3.0
```

Fig 7.6 Distance between camera

7.7 Web Camera New Frame

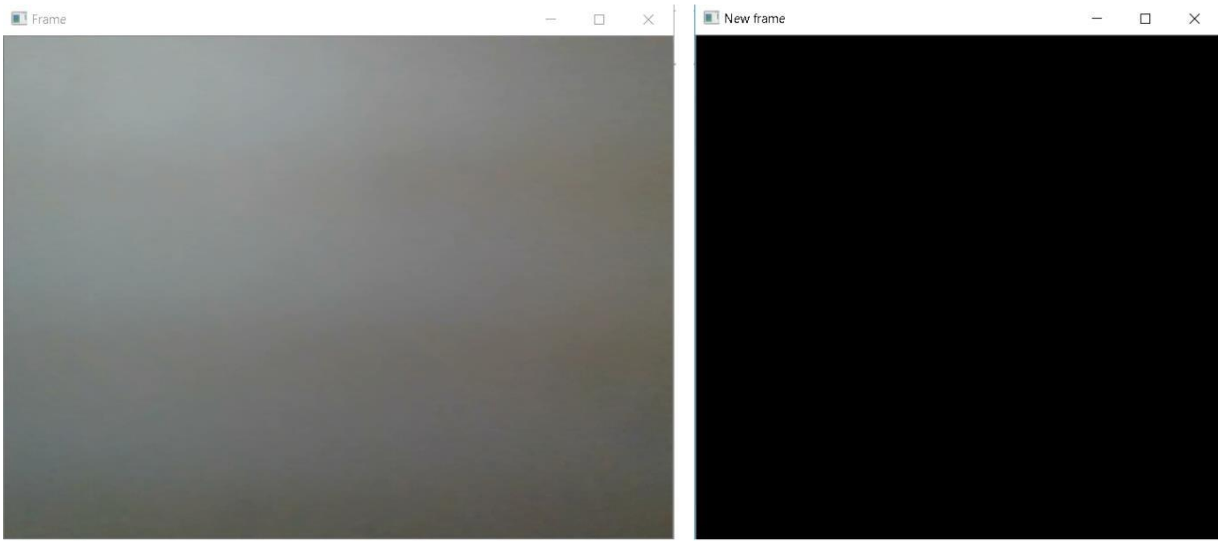
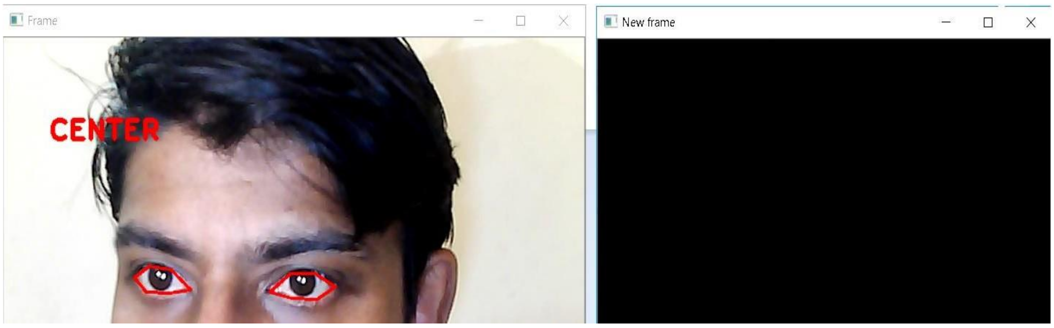


Fig 7.7 New Frame

7.8 Centre Position of Eye



7.8 Centre position of Eye

7.9 Right moment of Eye

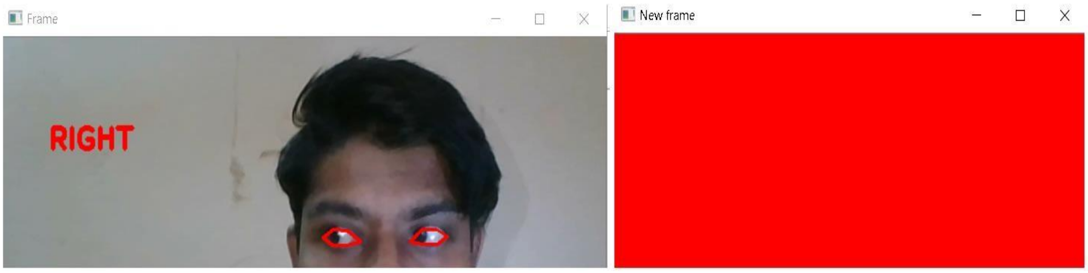
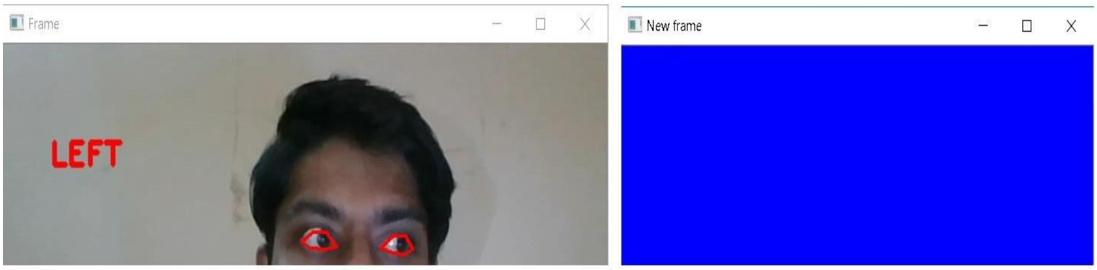


Fig 7.9 Right moment of Eye

7.10 Left moment of Eye



7.10 Left moment of Eye

7.11 Eye Blinking

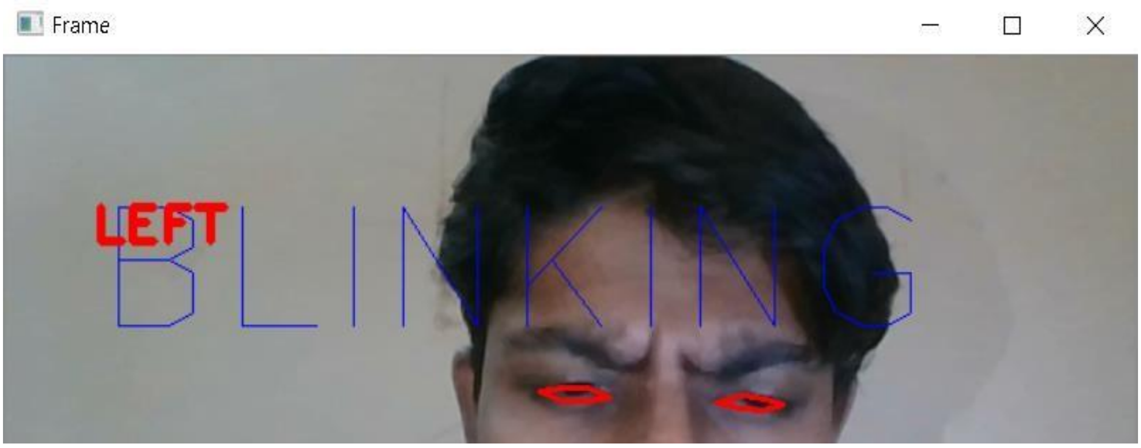


Fig 7.11 Eye Blinking

CHAPTER 8
CONCLUSION

CHAPTER 8

CONCLUSION

The proposed project aims to bring out a solution for the paralyzed people without any harm to their body externally or internally. It overweighs the previously developed prototypes in this field because none of the components are in direct contact with the patient’s body hence it definitely will prove to be safer.
Obtained results show that the proposed algorithm allows for accurate detection of voluntary eye-blinks with the rate of approximately 99%. Performed tests demonstrate that the designed eye-blink controlled user interface is a useful tool for the communication with the machine. The opinions of the users with reduced functioning were enthusiastic.

- **To make cost effective:** The main objective of developing algorithm of a real time video Oculography system is that to provide cost effective for those people who cannot afford. The existing technique for such patients to communicate is too costly.
- Thus, it is necessary to design a system which is affordable to common people which includes cost effective components for designing.
- **Fast:** There are few algorithms which are developed for video Oculography system for communication. The main objective of this project is to develop an algorithm which is extremely fast compared to the existing ones.
- **Accuracy:** The main objective of this project is to develop an algorithm which is more accurate compared to the existing ones.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Joshua D. Fischer, Dawie J. van den Heever, "Portable Video-Oculography Device for Implementation in Sideline Concussion Assessments: A Prototype", 2016 IEEE conference.
2. Sudhir Rao Rupanagudi, Vikas N S, Vivek C Bharadwaj, Manju; Dhruva N; Sowmya K. S, "Novel methodology for blink recognition using video oculography for communicating", IEEE Conference Publications, 2014, ICAEE.
3. Masaru Kiyama, Hitoshi Iyatomi and Koichi Ogawa, "Development of robust video-oculography system for non-invasive automatic nerve quantification", IEEE Conference Publications, 2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences.
4. S. M. H. Jansen, H. Kingma, R. L. M. Peeters and R. L. Westra, "A torsional eye movement calculation algorithm for low contrast images in video-oculography ", IEEE Conference Publications, 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology.
5. Akshat Gambhir, K.S Boppaiah, M Shruthi Subbaiah, Pooja M and KiranP, RNSIT, "Video Oculographic System using Real-Time Video Processing", International Journal of Computer Applications (0975 –8887) Volume 119 –No.22, June 2015.
6. Karthik K.P and Basavaraju S, Department of Electronics & Communication Engineering, Sapthagiri College of Engineering, Bangalore, "Design and Implementation of Unique Video Oculographic Algorithm Using Real Time Video Processing on FPGA", IJSRD -International Journal for Scientific Research & Development| Vol. 3, Issue 04, 2015 | ISSN (online): 2321-0613.
7. Sowmya K S, Roopa B S and Rishika Prakash, Don Bosco Institute of Technology, Electronics & Communication, Bangalore, "A Novel Eye Movement Recognition Algorithm using Video Oculography", Proc. of Int. Conf. on Recent Trends in Signal Processing, Image Processing and VLSI, ICrtSIV.
8. Jacek Naruniec, Michael Wiezczorek, Marek Kowalsak "Webcam-based system for video- oculography", 2017
9. Rachitha S, Prashanth Kumar K.N, "Eyelid movement communication media for paralyzed patient using Fn-M16p module", 2016.

10. S.Saravanakumar, and N.Selvaraju, “Eye Tracking and Blink Detection for Human Computer Interface”, International Journal of Computer Applications, Vol 2 – No.2, May 2010.
11. Hewitt, R., “Seeing with OpenCV A Computer-Vision Library,” SERVO, January 2007
12. Hewitt, R., “Seeing with OpenCV Finding Faces in Images,” SERVO, February 2007, pp. 48-52.
13. Jones, M. and Viola, P. 2001. Rapid Object Detection using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition. IEEE Computer Society Conference.
14. Himani, S. P., Darshak, G. T. and Udesang, K. Jal. 2014. A Survey on Object Detection and Tracking Methods. International Journal of Innovative. Research in Computer and Communication Engineering.
15. Oualla, M., Sadiq, A. and Mbarki, S. 2015. Comarative study of the methods using Haar-like feature. International Journal of Engineering Sciences & Research Technology.