

AN EMBEDDED SYSTEM FOR DRIVER DROWSINESS DETECTION

BY

**MD. Saiful Islam
ID- 151-15-4831**

AND

**MD. Rafael Hossen
ID- 151-15-5428**

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Nazmun Nessa Moon
Assistant Professor
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2018

APPROVAL

This Project titled “An Embedded System for Driver Drowsiness Detection”, submitted by MD. Saiful Islam and MD. Rafael Hossen to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering (BSc) and approved as to its style and contents.

BOARD OF EXAMINERS



Prof. Dr. Syed Akther Hossain
Head

Department of CSE
Faculty of Science & Information Technology
Daffodil International University

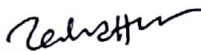
Chairman



Dr. Sheak Rashed Haider Noori
Associate Professor & Associate Head

Department of CSE
Faculty of Science & Information Technology
Daffodil International University

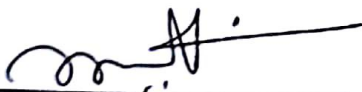
Internal Examiner



Md. Zahid Hasan
Assistant Professor
Department of CSE

Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Mohammad Shorif Uddin
Professor

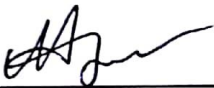
Department of Computer Science & Engineering
Jahangirnagar University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Nazmun Nessa Moon, Assistant Professor, Department of CSE Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Nazmun Nessa Moon
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:



MD. Saiful Islam
ID: 151-15-4831
Department of CSE
Daffodil International University



MD. Rafael Hossen
ID: 151-15-5428
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Nazmun Nessa Moon, Assistant Professor**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Embedded System and Digital Image Processing*” to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Prof. Dr. Syed Akhter Hossain, Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Our project titled “An Embedded System for Driver Drowsiness Detection” which aimed towards developing a real-time embedded system that detects driver drowsiness. A direct way of measuring driver fatigue is to measure the state of driver drowsiness. This system which captures image continuously and measures the state of the eye. For drowsiness detection, we considered the closure value of the eye. When the closure of eye crosses threshold value for a certain period of time then the driver is identified to be sleepy. To build the prototype we use Raspberry-pi, Camera, Audio Module. This system has several OpenCV libraries, dlib library and the method we are using is eye aspect ratio and facial landmark localization. To localize facial structure, we use facial landmark and using eye aspect ratio we can get the closure value of the eye. We use multiple software such as Raspbian Operating System, Python IDLE, CMake, Advance IP Scanner, MobaXterm to develop this system.

TABLE OF CONTENTS

CONTENTS	PAGE
Approval	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
CHAPTER	
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	2
CHAPTER 2: BACKGROUND	4-5
2.1 Introduction	4
2.2 Related Works	4
2.2.1 Volvo (Driver Alert Control)	4
2.2.2 Mercedes-Benz (Attention Assist)	4
2.2.3 BMW (Active Driving Assistant)	5
2.2.4 Anti Sleep Pilot	5
2.2.5 Vigo Smart Headset	5
2.3 Comparative Studies	5
CHAPTER 3: REQUIREMENT SPECIFICATION	6-9
3.1 Work Flow Diagram	6
3.2 Hardware Specification	6
3.2.1 Raspberry Pi 3 Model B	7
3.2.2 Camera	7

3.2.3 Audio Module	8
3.3 Software Requirements	8
3.3.1 Raspbian Operating System	9
3.3.2 Python 3 IDLE	9
3.3.3 Visual Studio	9
3.3.4 Advance IP Scanner	9
3.3.5 MobaXterm	9
3.3.6 CMake	9
CHAPTER 4: METHODS AND LIBRARY	10-14
4.1 Introduction	10
4.2 Facial Landmark	10
4.2.1 Image Localization	10
4.2.2 Detecting Key Structure	10
4.3 Eye Aspect Ratio	11
4.4 Library	13
4.4.1 SciPy Package	13
4.4.2 Dlib Library	13
4.4.3 Imutils Package	14
4.4.4 PyGame Module	14
CHAPTER 5: IMPEMETATION AND TESTING	15-19
5.1 Introduction	15
5.2 Hardware Setup	15
5.2.1 Raspberry Pi 3 Model B Setup	15
5.2.2 Camera Setup	15
5.3 Library Setup	16
5.3.1 SciPy Package Installation	16
5.3.2 Installing Dlib Library	17
5.3.3 Installing Imutils Package	17
5.3.4 PyGame Module for Sound	17
5.4 Algorithm Implementation	17

5.5 Result	18
5.6 System Outlook	19
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	20-22
6.1 Obstacles and Discussion	20
6.2 Limitation	20
6.3 Future Work	21
6.4 Conclusion	22
REFERENCES	23

LIST OF FIGURES

FIGURES	PAGE
Figure 3.1 Workflow Diagram	6
Figure 3.2 Raspberry Pi 3 Model B	7
Figure 3.3 Raspberry Pi Camera Module	8
Figure 4.1 Visualizing the 68 facial landmark coordinates	11
Figure 4.2 Open and closed eyes with facial landmarks	12
Figure 5.1 Raspberry Pi 3 Model B Ports	16
Figure 5.2 Eye is in Open State	18
Figure 5.3 Eye is in Close State	18
Figure 5.4 System Outlook	19

LIST OF TABLES

TABLES	PAGE
Table 3.1 Specification of Raspberry Pi 3 Model B	7
Table 3.2 Specification of Raspberry Pi Camera Module	8

CHAPTER 1

INTRODUCTION

1.1 Introduction

This project deals with a severe social problem that takes 1 life every 25 seconds in the world. It is no different in Bangladesh. Detection of driver's drowsiness can save life and property. Consciousness level of driver decreases because of a long continuous journey or lack of sleep. Drowsiness is a very complex situation where the body feels abnormally sleepy or tired. Though there is no direct way of measuring drowsiness but several indirect methods which can be used. These methods mostly image processing based methods. The method we are using is facial landmark localization [1]. This method helps to find a face where the face is also object.

In the latter section, we detect the eye region where we can compute eye aspect ratio. This aspect ratio will determine whether the eyes are closed or not. If eyes have been closed for a certain period of time we will sound alarmed to notice the driver.

In chapter 4, the entire system is explained which includes Facial Landmark Localization, Real-Time Eye Blink Detection [2]. This system fully performs on these methods. Later part in chapter 4, the implementation part is described. The whole hardware system of this project.

In chapter 6, the future scope and the drawback of the system were explained. Finally, the conclusive part where overall performance is summarized.

1.2 Motivation

Motivation means a moving cause. Living in Bangladesh where road safety is a big issue. According to the Bangladesh Passenger Welfare Committee, 339 people killed in 277 road accidents during Eid holidays. According to the National Committee to Protect Shipping, Roads, and Railways (NCPSRR), "More than 4,284 people were killed in 2017" [3]. All these accidents not because of only driver fatigue but it is one of the important factors. Because sometimes each driver drove 10-15 hours long

journey without sleep. Sleep deprived driving also makes them reckless and cause the accident which we can avoid. From this thinking, we wanted to make a cost-efficient embedded system that everyone can use.

1.3 Objectives

The objectives which have been assigned to the project seem to be-

- To build an embedded system where the camera can monitor face, detect eye region and compute eye aspect ratio.
- To create a system where drivers get notified when they turn fatigue using sound.
- To make the system automatic when it gets a power supply.
- To make the product cost efficient so that everyone has access to it.
- To reduce vehicle-based road accident and create an impact on this social problem.

1.4 Expected Outcome

This system will deliver the following outcomes-

- Cost-efficient drowsiness detection system.
- Reduce the vehicle-based road accident rate.
- Make the driver pay attention while driving.
- Improve the driver's ability to make a good decision.
- A system that less expensive but more effective.
- An automatic system works simultaneously.

1.5 Report Layout

The following description will demonstrate which section covers which themes furthermore, their relevant discussions-

Chapter 1: Introduction

The basic foundation, selection & outcome of this project is discussed in this chapter.

Chapter 2: Background

Consists of relevant project & research work which have been studied while doing this project.

Chapter 3: Requirement Specification

This chapter contains a brief discussion on the hardware and software requirements. Also, the workflow of the system is discussed.

Chapter 4: Methods and Library

The algorithm and the python and OpenCV library that we used is discussed in this section.

Chapter 5: Implementation and Testing

A brief discussion on how the hardware, library and algorithm set up to build the system.

Chapter 6: Conclusions and Future Scope

It gives an overall idea, limitations of the project also the future aspect of this project.

CHAPTER 2

BACKGROUND

2.1 Introduction

There are millions of problem in our social life. Technologies are meant to solve the problem. Vehicle-based road accident problem growing rapidly. To reduce the casualties from road accident we initiate this project. This project deals with driver drowsiness. We will detect drowsiness using OpenCV, python & dlib library. The method we are using to implement this project is Facial landmark localization.

2.2 Related Works

Definitely, driver drowsiness detection system is not new. Many other top vehicle manufacturers have their own driver assistant system. Studying most of the system they use different sensors that track and analysis driving behavior. These systems also advise the driver to rest if they drove for a long time. Various technologies from various manufacturers are used to detect driver drowsiness. Some of the driver drowsiness detection system listed below-

2.2.1 Volvo Cars (Driver Alert Control): World's first Driver Drowsiness Detection system was introduced by Volvo cars name Diver Alert Control [4] back in 2007. This system monitors the vehicle is being driven in a controlled way or not instead of monitoring the driver's eye.

2.2.2 Mercedes-Benz (Attention Assist): Mercedes unveiled a complex structure where driver fatigue is being monitored using various high-end sensors named Attention Assist [5]. Although it doesn't have facial recognition. It examines driver driving pattern, time behind the wheel, driver activity, weather and many more factors.

2.2.3 BMW (Active Driving Assistant): Active Driving Assistant [6] is a camera-based system. It uses road markings, signals and makes orientation to another vehicle to stay in lane. It's attention assistant feature analyses driving pattern, driving time etc. and if necessary it advises the driver to take a break on the control display.

2.2.4 Anti Sleep Pilot: Anti Sleep Pilot [7] device suitable for any wheels. Basically, it calculates 26 different parameters including driver risk profile. However, it takes a while to set up.

2.2.5 Vigo Smart Headset: Vigo Smart Headset [8] is wearable device that monitor alertness through the eyes and head motion. If there is any drowsiness it will vibrate, beep and flash to alert the driver.

2.3 Comparative Studies

Studying about related drowsiness system on vehicle mostly they follow a driving pattern, driving time and many more. Very few systems use eye recognition. Besides all the above system comes in very high-end luxurious cars. The goal was to build it at the lowest possible cost which everyone can afford. Also, we didn't want a system that is wearable because people often tend to avoid this kind of system. So, our system is an embedded system that monitors the human eye to detect drowsiness.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Work-Flow Diagram

Technically our project deals with the detection of the human eye aspect ratio. To accomplish the goal, we break down the project shown in figure 3.1 with small steps.

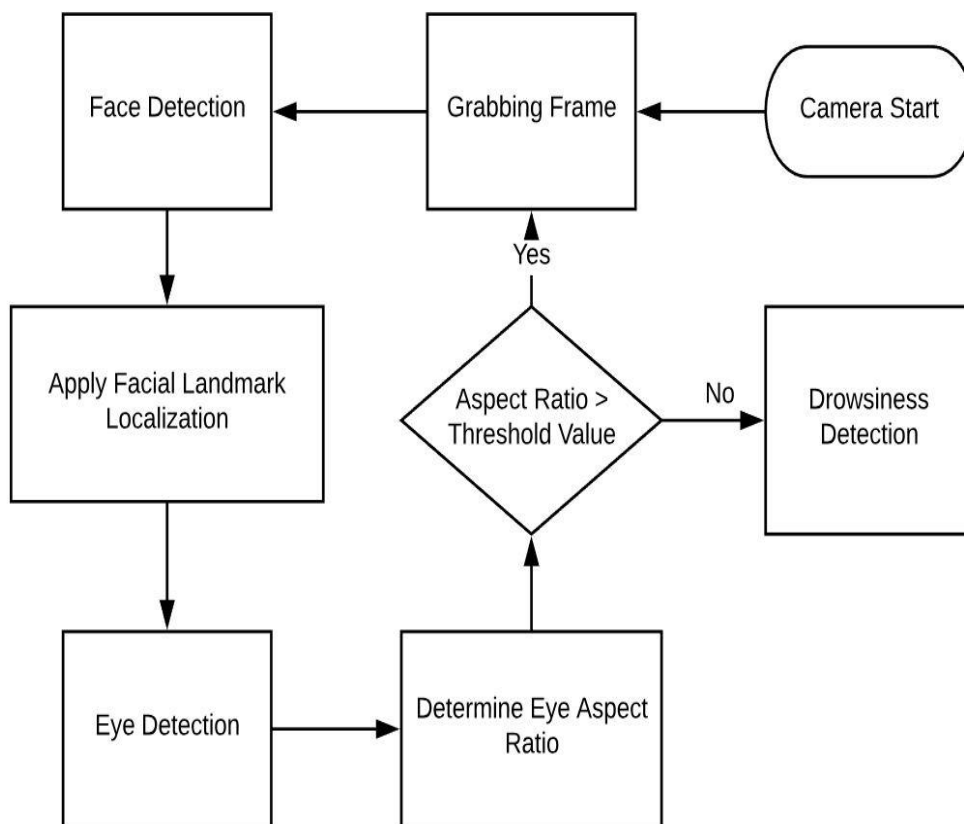


Figure 3.1: Work Flow Diagram

3.2 Hardware Specification

The crucial part of the project because we don't want a complex system. So the goal was to keep it simple. For the prototype, we are using low-cost hardware. The list of the hardware we are using listed below-

3.2.1 Raspberry Pi 3 Model B

One of the most use microcontroller at the lowest cost. Basically Raspberry Pi is a single board computer. Size of this microcontroller is one of the reasons for using it. Figure 3.2 shows the Raspberry Pi that we used for this project.



Figure 3.2: Raspberry pi 3 model B

Table 3.1 shows the specification of Raspberry Pi 3 model B.

Table 3.1 Specification of Raspberry Pi 3 Model B

Specification	Raspberry Pi 3 Model B
Processor	Quad Core 1.2GHz
System Type	64bit CPU
RAM	1 GB @ 400MHz
Power	2.5A@5v (max)

3.2.2 Camera

To manipulate the image of the driver camera is very essential hardware. The camera we are using is Raspberry Pi camera module. It is a 5-megapixel camera and captures 1080p video. It is very popular in home security. Figure 3.3 shows the camera and the specification of this camera shown in table 3.2.



Figure 3.3: Raspberry Pi Camera Module

Table 3.2 Specification of Raspberry Pi Camera Module

Specification	Raspberry Pi Camera Module
Resolution	5 Megapixel @1080p
Dimension	0.98inch x0.94inch x0.25 inch
Aperture	f 1.8
Focal Length	2.8 mm

3.2.3 Audio Module

The audio module we are using to alert the driver drowsiness is pam8403 audio amplifier. The new filter less architecture allows the device to drive the speaker directly, requiring no low-pass output filters, thus saving system cost and PCB area.

3.3 Software Requirements

Because of the project in an embedded system so that it has to be a combination of hardware and software. The software we are using to implement is listed below-

3.3.1 Raspbian Operating System

The Raspbian operating system is the primary operating system of Raspberry Pi single-board computers. It is Debian-based computer operating system.

3.3.2 Python 3 IDLE

Python 3 IDLE is an integrated development environment. It has the in-built implementation of python language. To develop the project in python we used python IDLE.

3.3.3 Visual Studio (For testing)

Visual Studio is an integrated development environment. The purpose of using visual studio was to test the program without hardware implementation.

3.3.4 Advance IP Scanner

Advance IP Scanner helps to show all network devices. It helps to find any connection between Raspberry Pi and Windows computer. It also gives access to shared folders.

3.3.5 MobaXterm

MobaXterm is a secure shell protocol that helps to connect to a remote computer. To establish a connection between Raspberry Pi to Windows computer we use MobaXterm.

3.3.6 CMake

CMake is an open source, cross-platform, extensible software application that helps to build C/C++ projects on any platform. To use C++ library dlib with python bindings we use CMake.

CHAPTER 4

METHODS AND LIBRARY

4.1 Introduction

This chapter deals with the methods and the libraries used to implement to achieve the target.

4.2 Facial Landmark

Facial landmark is basically a shape prediction problem of an image. It has become a key interest in computer vision. From facial expression to understanding human behavior this method is used. If there is an image input the shape predictor target to localize key points of interest with the shape.

Facial landmark implementation can be done using two steps.

- Image Localization
- Detecting Key Structure

4.2.1 Image Localization

Image localization can be achieved in several ways. There is built in Haar Cascade method in OpenCV. For fairly detailed information the Histogram of Oriented Gradients and Object Detection [9] method can also use to localize the image.

Most importantly there should be (x, y) coordinates of the face.

4.2.2 Detecting key Structure

Various facial landmark detectors we can localize and give a label to the following facial region-

- Eyes & Eyebrows
- Nose

- Jaw
- Mouth etc.

The process starts by using-

1. Manually labelled on an image specifying (x, y) coordinates of regions of each structure.
2. Also the probability of distance between the pixels.

Figure 4.1 is the image of pre-trained facial landmark detector that estimates the location of (x, y) coordinates on the face.

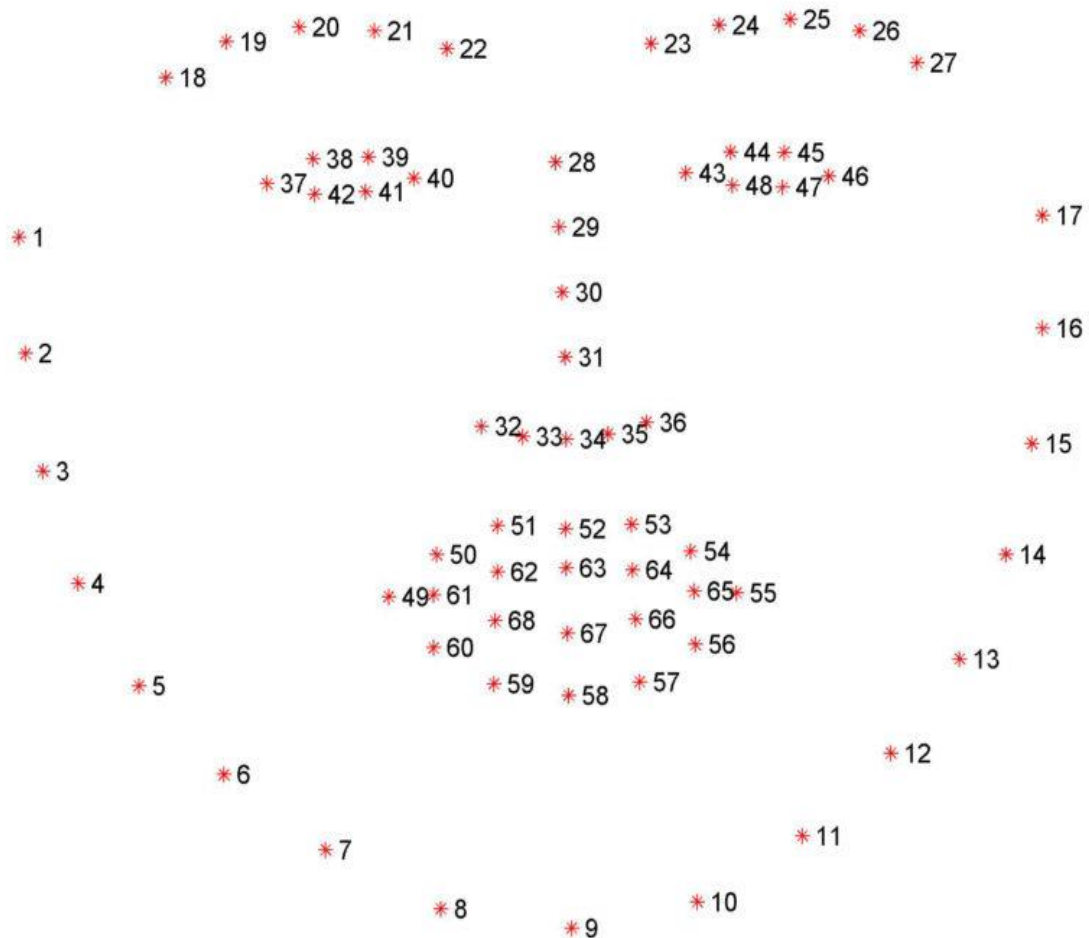


Figure 4.1 Visualizing the facial landmark coordinates

4.3 Eye Aspect Ratio

To compute eye blink there is a traditional image processing method which is slow and less effective. The method involves the following steps-

- Localization of eyes
- Find the white section of the eyes.
- Determine when the white section went for a time period resulting in an eye blink.

Unlike this method, the eye aspect ratio is much faster and powerful. The method by which we compute eye aspect ratio was introduced by Tereza Soukupová and Jan Čech in their paper, Real-Time Eye Blink Detection using Facial Landmarks [2].

The equation derived from the work done by Tereza Soukupová and Jan Čech in their paper is called the eye aspect ratio equation.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Where P_1, \dots, P_6 are 2D facial landmark position. The numerator and denominator respectively calculate the distance between the vertical and horizontal eye landmarks. This equation is so important because we can simply rely on the equation rather than other image processing techniques. Consider the figure 4.2 for better understanding.

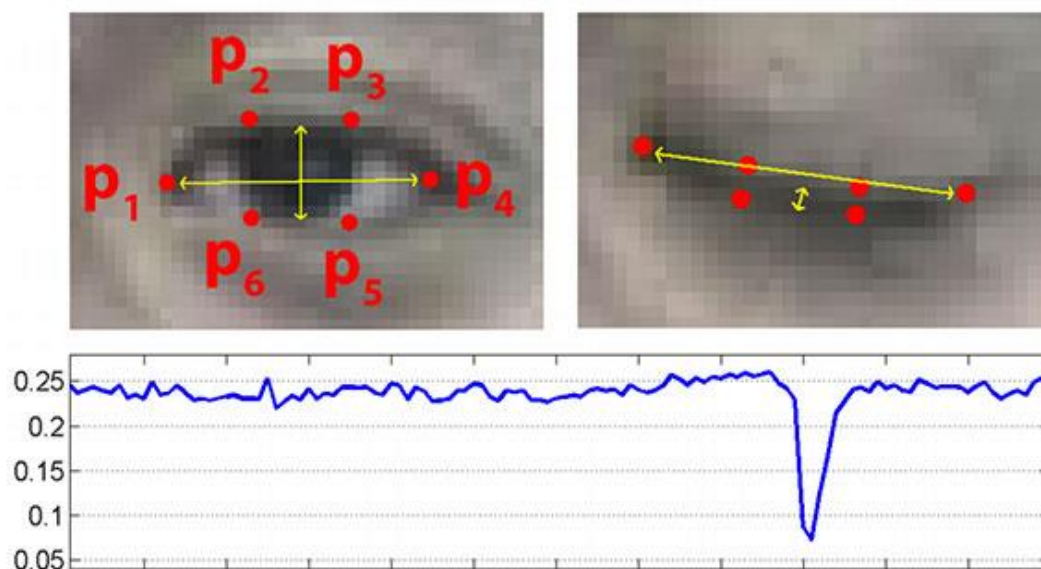


Figure 4.2 Open and closed eyes with facial landmarks

The eye aspect ratio value is approximately constant while the eyes are open and the value drops when a person starts blinking.

4.4 Library

A package which contains data, file and compiled code. Library is nothing but a collection of packages. For implementation purpose we used some of python and C++ library. List of packages are given below-

4.4.1 SciPy Package

SciPy is an open source python library mainly used for mathematical, scientific computation. To compute Euclidean distance between facial landmarks points the SciPy package were used.

4.4.2 Dlib Library

Dlib is one of the essential open source cross-platform tool for component-based software implementation, free of charge. It is written in C++ programming language. There are wide variety of features in this library. Some of them are given below-

- High-Quality Portable Code
- Documentation
- Machine Learning Algorithm
- Numerical Algorithm
- Image Processing
- Networking
- Graphical User Interface (GUI)
- Testing etc.

Because this library written in C++, we used CMake to convert the library with python bindings.

We use dlib library to detect and localize facial landmarks. Although dlib library is very complicated to import. It takes long time and it is the most important library.

4.4.3 Imutils Package

Basically it is a series of various image processing function. It becomes very easy to rotate, translate, display images with OpenCV and also with python.

4.4.4 PyGame Module

PyGame is a free open source cross-platform Python module that has include computer graphics, sound libraries etc. It is written in Python programming language. We used this mainly for playing sound as an alarm when camera detects drowsiness of the driver.

These are the methods and library that used while doing this project.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

Chapter 5 covers the implementation of drowsiness detection system with the hardware. Hardware part setup also the installation process of various library is described. Theoretical part of the algorithm that we are using described briefly.

5.2 Hardware Setup

Different hardware setup described individually-

5.2.1 Raspberry Pi 3 Model B

The operating system that we used in our Raspberry Pi is Raspbian. Installing operating system images on Raspberry Pi [10], some steps should be followed-

- Take the formatted memory card of the Raspberry pi and plug it to computer.
- Download the zipped Raspbian image file.
- Burn to write the Raspbian image to the SD card
- Eject the memory and plug it to Raspberry pi.
- Once it boots up then the installer will come up.
- Select Raspbian and click Install.

Thus Raspbian will setup on Raspberry pi 3 model B.

5.2.2 Camera Setup

Raspberry pi camera module is very easy to setup on Raspberry pi 3 model B. We just need to connect the cable into CSI camera port on Raspberry pi computer. For better understanding the ports of Raspberry pi 3 model B is given in figure 5.1

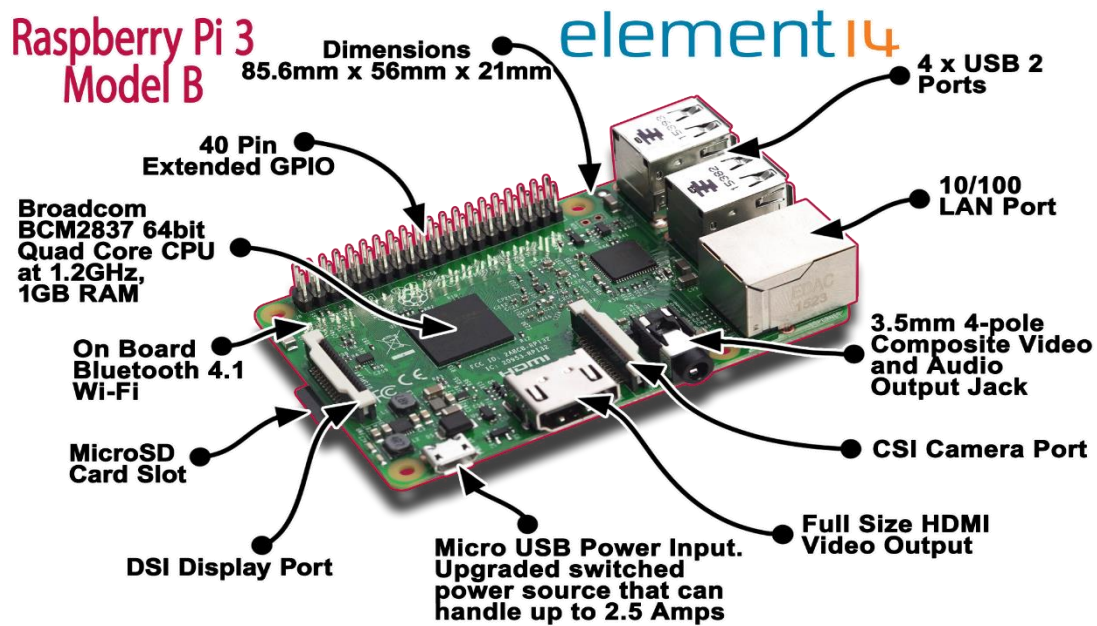


Figure 5.1 Raspberry pi 3 model B ports

5.3 Library Installation

Installation process of the library we used described below-

5.3.1 SciPy Package Installation

For installation we installed python and pip on our system. Entering the following command will import Scipy package.

\$ pip install numpy

\$ pip install scipy

\$ pip install scikit-image

5.3.2 Installing Dlib Library

Installing dlib library on Raspberry pi is very complicated. We followed an article by Adrian Rosebrock about “Install dlib on the Raspberry Pi” [11]. The step we followed is given below-

- i. Update swap file size, boot options, and memory split

- ii. Install dlib prerequisites
- iii. Access your Python virtual environment
- iv. Use pip to install dlib with Python bindings
- v. Test out your dlib install
- vi. Reset your swap file size, boot options, and memory split

The six steps were followed when we install dlib onto our Raspberry pi.

5.3.3 Installing Imutils Package

Imutils package is totally pip instable. Enter the following code to install imutils-

```
$ pip install imutils
```

5.3.4 PyGame Module for Sound

For alarm sound we use PyGame sound library. To import the module-

```
import pygame
```

5.4 Algorithm Implementation

After implementing all the library, we defined a function called `eye_aspect_ratio`. The function calculates the ratio distance between both horizontal and vertical eye landmarks. If the return value of the function is approximately constant, it is considered as open eyes. The value decreases we someone blink.

Anyone can blink while driving. System can't make alarm sound every time person blinks. For that we need a threshold value and a counter. Consider threshold value as limit of eye blink. That means if the return value from the function is lower than threshold value it will consider the frame as drowsy detected frame. If it happens consecutive time (more than 5 sec) system will consider the state as drowsy and will play the alarm sound.

Threshold value is different for different person. User have to adjust according to their eyes. Also, the increment of counter is changeable.

5.5 Result

The prototype of our system “Drowsiness detection” tested on different condition. The result is shown below-



Figure 5.2 Eye is in Open State

In figure 5.2 the eye aspect ratio value is .29. We initialized the threshold value to be .20. The return value is greater than the threshold value. So that, system considered the eye state as open state.



Figure 5.3 Eye is in close state

Another example, see figure 5.3, where system detects drowsiness. Here the eye aspect ratio is .20 which is same as the threshold value. System will continuously check next frame. If the value of next 5 frame is same or lower than the threshold value, then system will detect drowsiness.

5.6 System Outlook

Some of the final output of the system is shown in figure 5.4

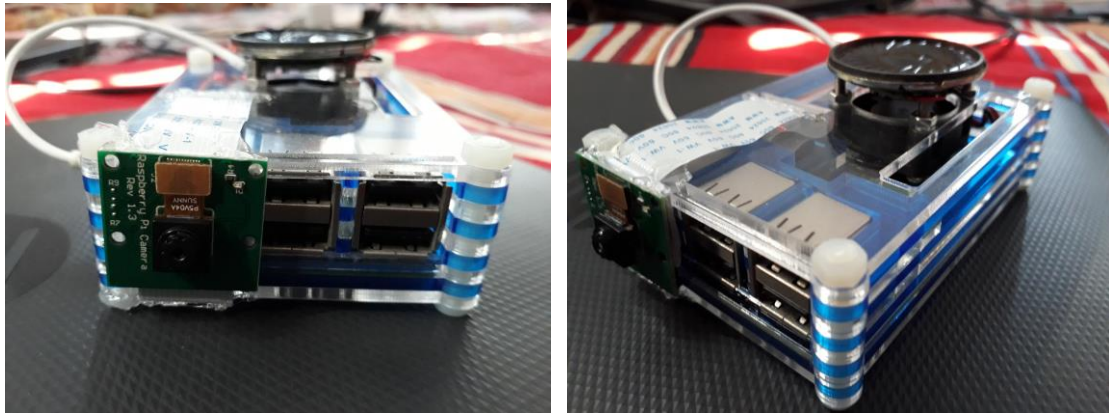


Figure 5.4 System Outlook

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Obstacles and Discussion

Image processing is a completely new field for us to work on. While doing this project we faced many difficulties. Finding appropriate resources was a difficult part. Choosing the right platform to build the system was confusing. The installation process of various libraries was quite hectic. Choosing the right hardware equipment took some time as the project objective was to build at the lowest cost.

Taking aside the obstacle part we have learned many new things while doing this project. OpenCV and working with Raspberry Pi was completely new and exciting. Getting started with various Python, OpenCV libraries made easier to work on. We came to know about various face detection algorithms and methods. While doing this project we also gathered information about road safety and the impact of the road accident on our society.

6.2 Limitation

Processor Speed: For the prototype of the system we have used Raspberry Pi 3 model B. The processor speed is not sufficient for video processing. It takes a longer time to process the video. So that, a high processor is needed that will eventually increase the cost of the product.

Dependency on Bright Light: The developed model depends on bright light condition. Because the algorithm needs to calculate the distance between both horizontal and vertical eye landmarks, it is essential to adjust facial landmark locations. Additionally, this product is not developed for night condition. The camera we used does not support night vision. Most of the time ambient light is not enough to detect. It depends on the

level of light. We can use an infrared camera with night vision feature or any bright LED light to avoid these circumstances.

Use of Eyewear: If the user uses eyewear then it becomes difficult to determine the state of the eye. Light reflection of the eyewear may produce different output.

The Distance of Camera: For the best result, the user face should be nearly 100 cm. It may take time to process when the distance exceeds.

Threshold Value Adjustment: Every person's eye shape is different. The threshold value for one user is not same to other users. The automatic adjustment of the threshold value is not present in the system.

6.3 Future Work

The designed system can detect the drowsy state of the user's eye and give the alarm to alert the user. For the future aspect of the system we have some plans. Some of the features are given below-

- Processor speed will be increased for faster action.
- The dependency of light will be reduced by using night vision infrared camera.
- Modify the algorithm for night purpose.
- Develop this system for a mobile application.
- A sound alarm is not enough to avoid accident so that this system should be embedded in a motor-driven system that can control the speed of the vehicle.
- Enabling drowsy light for pedestrians and other drivers so that they can take safety measures.
- Automatic adjustment of threshold value according to the user's eye.

We are hopeful to add these feature in future.

6.4 Conclusion

Technology solves problems and road accident is a big issue in any country. Our aim was to do any project that has an impact on our social life. Driver's drowsiness is one of the negative factors for road safety. The goal is to reduce the death rate of vehicle-based accident.

We build the system by capturing frame by frame and implementing the facial landmark (blink detection) algorithm by Tereza Soukupová and Jan Cech. Calculating the ratio distance between both horizontal and vertical eye landmarks can lead us when a person blinks. We integrated this with Raspberry Pi 3 Model B computer.

We took a great deal while doing this project hoping it will help to reduce the casualties from the road accident.

REFERENCES

- [1] Rosebrock, Adrian. "Facial landmarks with dlib, OpenCV, and Python" <<<https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>>> last accessed on 15 November 2018 at 11:05 pm.
- [2] Tereza Soukupová and Jan Cech, "Real-Time Eye Blink Detection using Facial Landmarks", 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, February 3–5, 2016
- [3] Islam, M Zahidul. "The challenges for safe roads in Bangladesh" <<<http://m.theindependentbd.com/home/printnews/157331>>> last accessed on 7 Nov. 2018 at 8.15 pm.
- [4] Volvo Launches Driver Alert Control, available at <<<https://www.motor1.com/news/3629/volvo-launches-driver-alert-control/>>> last accessed on 10 November 2018 at 7.50 pm.
- [5] What is Mercedes-Benz ATTENTION ASSIST, available at <<<https://mercedesbenz.starmotorcars.com/blog/what-is-mercedes-benz-attention-assist/>>> last accessed on 10 November at 8.00 pm.
- [6] What is Active Driving Assistant, available at <<<http://www.pacificbmw.com/blog/what-is-active-driving-assistant-in-2018-bmw-6-series-gran-turismo/>>> last accessed on 10 November at 8.15 pm.
- [7] Coxworth, Ben. "Anti-Sleep Pilot detects drowsy drivers", <<<https://newatlas.com/anti-sleep-pilot-monitors-driver-fatigue/17439/>>> last accessed on 10 November at 9.00 pm.
- [8] Rusco, Lainie. "Device could detect driver drowsiness, make roads safer", <<<https://phys.org/news/2015-07-device-driver-drowsiness-roads-safer.html>>> last accessed on 10 November at 11.00 pm.
- [9] Rosebrock, Adrian. "Histogram of Oriented Gradients and Object Detection." <<<https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/>>> last accessed on 20 November 2018 at 12:45 am.
- [10] Installing Operating System Images, available at <<<https://www.raspberrypi.org/documentation/installation/installing-images/windows.md>>> last accessed on 17 October 2018 at 9:30 pm.
- [11] Rosebrock, Adrian. "Install dlib on the Raspberry Pi" <<<https://www.pyimagesearch.com/2017/05/01/install-dlib-raspberry-pi/>>> last accessed on 22 November 2018 at 9:00 am.

Document Viewer

Turnitin Originality Report

Processed on: 24-Nov-2018 10:18 +06

ID: 1043958891

Word Count: 4928

Submitted: 1

151-15-4831 By Md. Saiful
Islam

Similarity Index <h2 style="margin: 0;">16%</h2>	Similarity by Source Internet Sources: 13% Publications: 3% Student Papers: 14%
---	---

[include quoted](#)
[include bibliography](#)
[excluding matches < 1%](#)
[download](#)
[refresh](#)
[print](#)
 mode:

4% match (student papers from 04-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [940891790](#)

3% match (student papers from 07-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [942463120](#)

2% match (student papers from 04-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [940850308](#)

1% match (Internet from 21-Sep-2018) ✕

<http://dspace.library.daffodilvarsity.edu.bd:8080>

1% match (student papers from 07-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [942529897](#)

1% match (Internet from 18-Nov-2018) ✕

<https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/>

1% match (Internet from 23-Nov-2018) ✕

<https://www.pyimagesearch.com/2017/05/01/install-dlib-raspberry-pi/>

1% match (student papers from 16-May-2016) ✕

[Submitted to National Institute of Technology, Rourkela on 2016-05-16](#)

1% match (publications) ✕

[Guy Hart-Davis. "Deploying Raspberry Pi in the Classroom", Springer Nature, 2017](#)

1% match (student papers from 18-Sep-2017) ✕

[Submitted to Universiti Tenaga Nasional on 2017-09-18](#)

1% match (student papers from 07-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [942529588](#)

AN EMBEDDED SYSTEM FOR DRIVER DROWSINESS DETECTION BY MD. Saiful Islam ID- 151-15-4831 AND MD. Rafael Hossen ID- 151-15-5428 [This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering Supervised By Nazmun Nessa Moon Assistant Professor Department of CSE Daffodil International University Co-Supervised By Dr. Fernaz Narin Nur Assistant Professor Department of CSE Daffodil International University DAFFODIL INTERNATIONAL UNIVERSITY DHAKA, BANGLADESH NOVEMBER 2018 APPROVAL](#) This [Project titled "An Embedded System for Driver Drowsiness Detection"](#), submitted by MD. Saiful Islam and MD. Rafael Hossen [to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering \(BSc\) and approved as to its style and contents. BOARD OF EXAMINERS Prof. Dr. Syed Akther Hossain Head Department of CSE Faculty of Science & Information Technology Daffodil International University Chairman Dr. Sheak Rashed Haider Noori Associate Professor & Associate Head Department of CSE Faculty of Science & Information Technology Daffodil International University Internal Examiner Md. Zahid Hasan Assistant Professor Department of CSE Faculty of Science & Information Technology Daffodil International University Internal Examiner Dr. Mohammad Shorif Uddin Professor Department of Computer Science & Engineering Jahangirnagar University External Examiner ii \[DECLARATION We hereby declare that, this project has been done by us under the supervision of Nazmun Nessa Moon, Assistant Professor, Department of CSE Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma. Supervised by: Nazmun Nessa Moon Assistant Professor Department of CSE Daffodil International University Co-Supervised by: Dr. Fernaz Narin Nur Assistant Professor Department of CSE Daffodil International University Submitted by: MD. Saiful Islam ID: 151-15-4831 Department of CSE Daffodil International University MD. Rafael Hossen ID: 151-15-5428 Department of CSE Daffodil International University iii ACKNOWLEDGEMENT First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully. We really grateful and wish our profound our indebtedness to Nazmun Nessa Moon, Assistant Professor, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "Embedded System" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project. We would like to express our\]\(#\)](#)

[heartiest gratitude to Prof. Dr. Syed Akhter Hossain, Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University. We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work. Finally, we must acknowledge with due respect the constant support and patients of our parents.](#)

[iv ABSTRACT](#) Road accident [is](#) supposed to be an unfortunate event that happens unintentionally. In recent decades, the understanding of the factors that cost over 1.2 million lives each year due to the road accident. Driver fatigue becoming one of the major factors of the road accident which can be avoided. An unswerving approach to measure driver fatigue is to measure the state of driver drowsiness. This project aimed towards developing a real-time embedded system that detects driver drowsiness. This

[system which captures image continuously and measures the state of the eye.](#) For drowsiness detection we considered [closure value of eye. When the closure of eye](#) crosses threshold value for certain period of time then the driver is identified to be sleepy. To build the prototype we use Raspberry-pi, Camera, Alert sound. This system has several OpenCV libraries, dlib library and the method we are using to detect face and eye is facial landmark localization.

[v TABLE OF CONTENTS CONTENTS](#) PAGE Approval ii [Declaration](#)

iii [Acknowledgements](#) iv [Abstract](#) v [CHAPTER CHAPTER 1: INTRODUCTION](#)

[1.1 Introduction 1.2 Motivation 1.3 Objectives 1.4 Expected Outcome 1.5](#)

Report Layout 1-3 1 1 2 2 2 CHAPTER 2: BACKGROUND 2.1 Introduction 2.2

Related Works 2.2.1 Volvo (Driver Alert Control) 2.2.2 Mercedes-Benz

(Attention Alert) 2.2.3 BMW (Active Driving Assistant) 2.2.4 Anti Sleep Pilot

2.2.5 Vigo Smart Headset 2.3 Comparative Studies 4-5 4 4 4 4 5 5 5

CHAPTER 3: REQUIREMENT SPECIFICATION 3.1 Work Flow Diagram 3.2

Hardware Specification 3 [.2.1 Raspberry Pi 3 Model B](#) 3.2.2 Camera 6-9 6 6 7

7 vi 3.3 Software Requirements 3.3.1 Raspbian Operating System 3.3.2

Python 3 IDLE 3.3.3 Visual Studio 3.3.4 Advance IP Scanner 3.3.5

MobaXterm 3.3.6 CMake 8 8 9 9 9 9 9 CHAPTER 4: METHODS AND LIBRARY

4.1 Introduction 4.2 Facial Landmark 4.2.1 Image Localization 4.2.2

Detecting Key Structure 4.3 Eye Aspect Ratio 4.4 Library 4.4.1 SciPy Package

4.4.2 Dlib Library 4.4.3 Imutils Package 4.4.4 PyGame Module 10-14 10 10

10 10 11 13 13 13 14 14 CHAPTER 5: IMPLEMETATION AND TESTING 5.1

Introduction 5.2 Hardware Setup 5.2.1 Raspberry Pi 3 Model B Setup 5.2.2

Camera Setup 5.3 Library Setup 5.3.1 SciPy Package Installation 5.3.2

Installing Dlib Library 5.3.3 Installing Imutils Package 5.3.4 PyGame Module

for Sound 5.4 Algorithm Implementation 5.5 Result 15-19 15 15 15 16 16 16

17 17 17 17 18 vii 5.6 System Outlook 19 [CHAPTER 6: CONCLUSION AND](#)

[FUTURE SCOPE 6.1](#) Obstacles [and](#) Discussion [6.2 Limitation 6.3](#) Future Work

6.4 Conclusion 20-22 20 20 21 22 REFERENCES 23 LIST OF FIGURES

FIGURES ©Daffodil International University PAGE viii Figure 3.1 Workflow

Diagram Figure 3.2 Raspberry Pi 3 Model B Figure 3.3 Raspberry Pi Camera

Module Figure 4.1 Visualizing the 68 facial landmark coordinates Figure 4.2

Eye aspect ratio equation Figure 4.3 Open and closed eyes with facial

landmarks Figure 5.1 [Raspberry Pi 3 Model B Ports](#) Figure 5.2 Eye is in Open

State Figure 5.3 Eye is in Close State Figure 5.4 System Outlook 6 7 8 11 12

12 16 18 19 19 LIST OF TABLES TABLES PAGE Table 3.1 [Specification of](#)

[Raspberry Pi 3 Model B](#) Table 3.2 [Specification of Raspberry Pi Camera](#)

[Module](#) 7 8 ix CHAPTER 1 Introduction 1.1 Introduction This project deals

with a severe social problem that takes 1 life every 25 seconds in the world.

It is no different in Bangladesh. Detection of driver's drowsiness can save life

and property. Consciousness level of driver decreases because of a long

continuous journey or lack of sleep. Drowsiness is a very complex situation

where the body feels abnormally sleepy or tired. Though there is no direct

way of measuring drowsiness but several indirect methods which can be

used. These methods mostly image processing based methods. The method we are using is facial landmark localization [1]. This method helps to find a face where the face is also object. In the latter section, we detect the eye region where we can compute eye aspect ratio. This aspect ratio will determine whether the eyes are closed or not. If eyes have been closed for a certain period of time we will sound alarmed to notice the driver. In chapter 4, the entire system is explained which includes Facial Landmark Localization, Real-Time Eye Blink Detection [2]. This system fully performs on these methods. Later part in chapter 4, the implementation part is described. The whole hardware system of this project. In chapter 6, the future scope and the drawback of the system were explained. Finally, the conclusive part where overall performance is summarized.

1.2 Motivation

Motivation means a moving cause. Living in Bangladesh where road safety is a big issue. According to the Bangladesh Passenger Welfare Committee, 339 people killed in 277 road accidents during Eid holidays. According to the National Committee to Protect Shipping, Roads, and Railways (NCPSRR), "More than 4,284 people were killed in 2017" [3]. All these accidents not because of only driver fatigue but it is one of the important factors. Because sometimes each driver drove 10-15 hours long journey without sleep. Sleep deprived driving also makes them reckless and cause the accident which we can avoid. From this thinking, we wanted to make a cost-efficient embedded system that everyone can use.

1.3 Objectives

The objectives which have been assigned to the project seem to be-

- To build an embedded system where the camera can monitor face, detect eye region and compute eye aspect ratio.
- To create a system where drivers get notified when they turn fatigue using sound.
- To make the system automatic when it gets a power supply.
- To make the product cost efficient so that everyone has access to it.
- To reduce vehicle-based road accident and create an impact on this social problem.

1.4 Expected Outcome

This system will deliver the following outcomes-

- Reduce the vehicle-based road accident rate.
- Make the driver pay attention while driving.
- Improve the driver's ability to make a good decision.
- A system that less expensive but more effective.
- An automatic system works simultaneously.

1.5 Report Layout

The following description will demonstrate which section covers which themes furthermore, their relevant discussions-

Chapter 1: Introduction The basic foundation, selection & outcome of this project is discussed in this chapter. Chapter 2: Background Consists of relevant project & research work which have been studied while doing this project. Chapter 3: Requirement Specification This chapter contains a brief discussion on the hardware and software requirements. Also, the workflow of the system is discussed. Chapter 4: Methods and Library The algorithm and the python and OpenCV library that we used is discussed in this section. Chapter 5: Implementation and Testing A brief discussion on how the hardware, library and algorithm set up to build the system. Chapter 6: Conclusions and Future Scope It gives an overall idea, limitations of the project also the future aspect of this project.

CHAPTER 2 Background

2.1 Introduction

There are millions of problem in our social life. Technologies are meant to solve the problem. Vehicle-based road accident problem growing rapidly. To reduce the casualties from road accident we initiate this project. This project deals with driver drowsiness. We will detect drowsiness using OpenCV, python & dlib library. The method we are using to implement this project is Facial landmark localization.

2.2 Related Works

Definitely, driver drowsiness detection system [4] is not new. Many other top vehicle manufacturers have their own driver assistant system. Studying most of the system they use different sensors that track and analysis driving behavior. These systems also advise the driver to rest if they drove for a long time. Various technologies from various manufacturers are used to detect driver drowsiness. Some of the driver drowsiness detection system listed below-

2.2.1 Volvo Cars (Driver Alert Control): World's first Driver Drowsiness Detection system was introduced by Volvo cars back in 2007. This system monitors the vehicle is being driven in a controlled way or not instead of monitoring the driver's eye.

2.2.2 Mercedes-Benz (Attention Alert): Mercedes unveiled a complex structure where driver fatigue is being monitored using various high-end sensors. Although it doesn't have facial recognition. It examines driver driving pattern, time behind the wheel, driver activity, weather and many more factors.

2.2.3 BMW (Active Driving Assistant): This is a camera-based system. It uses road markings, signals and makes orientation to another vehicle to stay in lane. It's ©Daffodil International University attention assistant feature analyses driving pattern, driving time etc. and if necessary it advises the driver to take a break on the control display.

2.2.4 Anti Sleep Pilot: This device suitable for any wheels. Basically, it calculates 26 different parameters including driver risk profile. However, it takes a while to set up.

2.2.5 Vigo Smart Headset: This device is wearable that monitor alertness through the eyes and head motion.

2.3 Comparative Studies Studying about related drowsiness system on vehicle mostly they follow a driving pattern, driving time and many more. Very few systems use eye recognition. Besides all the above system comes in very high-end luxurious cars. The goal was to build it at the lowest possible cost which everyone can afford. Also, we didn't want a system that is wearable because people often tend to avoid this kind of system. So, our system is an embedded system that monitors the human eye to detect drowsiness.

Requirement Specification

3.1 Work-Flow Diagram Technically our project deals with the detection of the human eye aspect ratio. To accomplish the goal, we break down the project with small steps. Figure 3.1: Work Flow Diagram

3.2 Hardware Specification The crucial part of the project because we don't want a complex system. So the goal was to keep it simple. For the prototype, we are using low-cost hardware. The list of the hardware we are using listed below-

3.2.1 [Raspberry Pi 3 Model B](#) One of [the](#) most use microcontroller at the lowest cost. Basically Raspberry Pi is a single board computer. Size of this microcontroller is one of the reasons for using it. Figure 3.2: Raspberry pi 3 model B The specification of Raspberry Pi 3 model B is given below-

Table 3.1 Specification of Raspberry Pi 3 Model B Specification

Raspberry Pi 3 Model B Processor Quad Core 1.2GHz System Type 64bit CPU RAM 1 GB @ 400MHz Power 2.5A@5v (max) Ports 4x USB, Ethernet

3.2.2 Camera To manipulate the image of the driver camera is very essential hardware. The camera we are using is Raspberry Pi camera module. It is a 5-megapixel camera and captures 1080p video. It is very popular in home security. [Figure 3.3: Raspberry Pi Camera Module](#)

Table 3.2 Specification of [Raspberry Pi Camera Module](#) Specification Raspberry Pi Camera Module Resolution 5 Megapixel @1080p Dimension 0.98inch x0.94inch x0.25 inch Aperture f 1.8 Focal Length 2.8 mm

3.3 Software Requirements Because of the project in an embedded system so that it has to be a combination of hardware and software. The software we are using to implement is listed below-

3.3.1 Raspbian Operating System The Raspbian operating system is the primary operating system of Raspberry Pi single- board computers. It is Debian-based computer operating system.

3.3.2 Python 3 IDLE Python 3 IDLE is an integrated development environment. It has the in-built implementation of python language. To develop the project in python we used python IDLE.

3.3.3 Visual Studio (For testing) Visual Studio is an integrated development environment. The purpose of using visual studio was to test the program without hardware implementation.

3.3.4 Advance IP Scanner Advance IP Scanner helps to show all network devices. It helps to find any connection between Raspberry Pi and Windows computer. It also gives access to shared folders.

3.3.5 MobaXterm MobaXterm is a secure shell protocol that helps to connect to a remote computer. To establish a connection between

Raspberry Pi to Windows computer we use MobaXterm. 3.3.6 CMake CMake is an open source, cross-platform, extensible software application that helps to build C/C++ projects on any platform. To use C++ library dlib with python bindings we use CMake. Methods and Library 4.1 Introduction This chapter deals with the methods and the libraries used to implement to achieve the target. 4.2 Facial Landmark Facial landmark is basically a shape prediction problem of an image. It has become a key interest in computer vision. From facial expression to understanding human behavior this method is used. If there is an image input the shape predictor target to localize key points of interest with the shape. Facial landmark implementation can be done using two steps. ? Image Localization ? Detecting Key Structure 4.2.1 Image Localization Image localization can be achieved in several ways. There is built in Haar Cascade method in OpenCV. For fairly detailed information the Histogram of Oriented Gradients and Object Detection [5] method can also use to localize the image. Most importantly there should be (x, y) coordinates of the face. 4.2.2 Detecting key Structure Various facial landmark detectors we can localize and give a label to the following facial region- • Eyes & Eyebrows • Nose • Jaw • Mouth etc. The process starts by using- 1. Manually labelled on an image specifying (x, y) coordinates of regions of each structure. 2. Also the probability of distance between the pixels. Here is the image of pre-trained facial landmark detector that estimates the location of (x, y) coordinates on the face. Figure 4.1 Visualizing the facial landmark coordinates 4.3 Eye Aspect Ratio To compute eye blink there is a traditional image processing method which is slow and less effective. The method involves, Localization of eyes Find the white section of the eyes. Determine when the white section went for a time period resulting in an eye blink. Unlike this method, the eye aspect ratio is much faster and powerful. The method by which we compute eye aspect ratio was introduced by Tereza Soukupová and Jan Čech in their paper, Real-Time Eye Blink Detection using Facial Landmarks [2]. The equation derived from the work done by Tereza Soukupová and Jan Čech in their paper is called the eye aspect ratio equation. Figure 4. 2 Eye aspect ratio equation Where P1, ..., P6 are 2D facial landmark position. The numerator and denominator respectively calculate the distance between the vertical and horizontal eye landmarks. This equation is so important because we can simply rely on the equation rather than other image processing techniques. Consider the following figure to understand better- Figure 4.3 Open and closed eyes with facial landmarks The eye aspect ratio value is approximately constant while the eyes are open and the value drops when a person starts blinking. 4.4 Library A package which contains data, file and compiled code. Library is nothing but a collection of packages. For implementation purpose we used some of python and C++ library. List of packages are given below- 4.4.1 SciPy Package SciPy is an open source python library mainly used for mathematical, scientific computation. To compute Euclidean distance between facial landmarks points the SciPy package were used. 4.4.2 Dlib Library Dlib is one of the essential open source cross-platform tool for component-based software implementation, free of charge. It is written in C++ programming language. There are wide variety of features in this library. Some of them are given below- ? High-Quality Portable Code ? Documentation ? Machine Learning Algorithm ? Numerical Algorithm ? Image Processing ? Networking ? Graphical User Interface (GUI) ? Testing etc. Because this library written in C++, we used CMake to convert the library with python bindings. We use dlib library to detect and localize facial landmarks. Although dlib library is very complicated to import. It takes long time and it is the most important library. 4.4.3 Imutils Package Basically it is a series of various image processing function. It becomes very easy to rotate, translate, display images with OpenCV and also with python. 4.4.4 PyGame Module PyGame is a free open source cross-platform Python module

that has include computer graphics, sound libraries etc. It is written in Python programming language. We used this mainly for playing sound as an alarm when camera detects drowsiness of the driver. These are the methods and library that used while doing this project. CHAPTER 5 Implementation and Testing [5.1 Introduction Chapter 5](#) covers [the implementation of drowsiness detection system with the hardware](#). Hardware part setup also the installation process of various library is described. Theoretical part of the algorithm that we are using described briefly. 5.2 Hardware Setup Different hardware setup described individually- 5.2.1 [Raspberry Pi 3 Model B The](#) operating system that we used in our Raspberry Pi is Raspbian. Installing operating system images on Raspberry Pi [6], some steps should be followed- ? Take the formatted memory card of the Raspberry pi and plug it to computer. ? Download the zipped Raspbian image file. ? Burn to write the Raspbian image to the SD card ? Eject the memory and plug it to Raspberry pi. ? Once it boots up then the installer will come up. ? Select Raspbian and click Install. Thus Raspbian will setup on Raspberry pi 3 model B. 5.2.2 Camera Setup Raspberry pi camera module is very easy to setup on Raspberry pi 3 model B. We just need to connect the cable into CSI camera port on Raspberry pi computer. For better understanding the ports of Raspberry pi 3 model B is given below- Figure 5.1 Raspberry pi 3 model B ports 5.3 Library Installation Installation process of the library we used described below- 5.3.1 SciPy Package Installation For installation we installed python and pip on our system. Entering the following command will import Scipy package. \$ [pip install numpy](#) \$ [pip install scipy](#) \$ [pip install scikit-image](#) 5.3.2 Installing Dlib Library Installing dlib library on Raspberry pi is very complicated. We followed an article by Adrian Rosebrock about "How to install dlib" [7]. The step we followed is given below- i. [Update swap file size, boot options, and memory split](#) ii. Install dlib prerequisites iii. Access your Python virtual environment iv. [Use pip to install dlib with Python bindings](#) v. Test out your dlib install vi. [Reset your swap file size, boot options, and memory split](#) The six steps were followed when we install dlib onto our Raspberry pi. 5.3.3 Installing Imutils Package Imutils package is totally pip instable. Enter the following code to install imutils- \$ pip install imutils 5.3.4 PyGame Module for Sound For alarm sound we use PyGame sound library. To import the module- import pygame 5.4 Algorithm Implementation After implementing all the library, we defined a function called eye_aspect_ratio. The function calculates the ratio distance between both horizontal and vertical eye landmarks. If the return value of the function is approximately constant, it is considered as open eyes. The value decreases we someone blink. Anyone can blink while driving. System can't make alarm sound every time person blinks. For that we need a threshold value and a counter. Consider threshold value as limit of eye blink. That means if the return value from the function is lower than threshold value it will consider the frame as drowsy detected frame. If it happens consecutive time (more than 5 sec) system will consider the state as drowsy and will play the alarm sound. Threshold value is different for different person. User have to adjust according to their eyes. Also, the increment of counter is changeable. 5.5 Result The prototype of our system "Drowsiness detection" tested on different condition. The result is shown below- Figure 5.2 Eye is in Open State Here the eye aspect ratio value is .29. We initialized the threshold value to be .20. The return value is greater than the threshold value. So that, system considered the eye state as open state. Another example, see figure 5.3, where system detects drowsiness. Here the eye aspect ratio is .20 which is same as the threshold value. System will continuously check next frame. If the value of next 5 frame is same or lower than the threshold value, then system will detect drowsiness. Figure 5.3 Eye is in close state 5.6 System Outlook Final output of the system is shown in figure 5.4. Figure 5.4 System Outlook [CHAPTER 6 Conclusion and Future Scope 6.1](#) Obstacles [and](#)

Discussion Image processing is a completely new field for us to work on. While doing this project we faced many difficulties. Finding appropriate resources was a difficult part. Choosing the right platform to build the system was confusing. The installation process of various libraries was quite hectic. Choosing the right hardware equipment took some time as the project objective was to build at the lowest cost. Taking aside the obstacle part we have learned many new things while doing this project. OpenCV and working with Raspberry Pi was completely new and exciting. Getting started with various Python, OpenCV libraries made easier to work on. We came to know about various face detection algorithms and methods. While doing this project we also gathered information about road safety and the impact of the road accident on our society.

6.2 Limitation

Processor Speed: For the prototype of the system we have used Raspberry Pi 3 model B. The processor speed is not sufficient for video processing. It takes a longer time to process the video. So that, a high processor is needed that will eventually increase the cost of the product.

Dependency on Bright Light: The developed model depends on bright light condition. Because the algorithm needs to calculate the distance between both horizontal and vertical eye landmarks, it is essential to adjust facial landmark locations. Additionally, this product is not developed for night condition. The camera we used does not support night vision. Most of the time ambient light is not enough to detect. It depends on the level of light. We can use an infrared camera with night vision feature or any bright LED light to avoid these circumstances.

Use of Eyewear: If the user uses eyewear then it becomes difficult to determine the state of the eye. Light reflection of the eyewear may produce different output.

The Distance of Camera: For the best result, the user face should be nearly 100 cm. It may take time to process when the distance exceeds.

Threshold Value Adjustment: Every person's eye shape is different. The threshold value for one user is not same to other users. The automatic adjustment of the threshold value is not present in the system.

6.3 Future Work

The designed system can detect the drowsy state of the user's eye and give the alarm to alert the user. For the future aspect of the system we have some plans. Some of the features are given below-

- Processor speed will be increased for faster action.
- The dependency of light will be reduced by using night vision infrared camera.
- Modify the algorithm for night purpose.
- Develop this system for a mobile application.
- A sound alarm is not enough to avoid accident so that this system should be embedded in a motor-driven system that can control the speed of the vehicle.
- Enabling drowsy light for pedestrians and other drivers so that they can take safety measures.
- Automatic adjustment of threshold value according to the user's eye. We are hopeful to add these feature in future.

6.4 Conclusion

Technology solves problems and road accident is a big issue in any country. Our aim was to do any project that has an impact on our social life. Driver's drowsiness is one of the negative factors for road safety. The goal is to reduce the death rate of vehicle-based accident. We build the system by capturing frame by frame and implementing the facial landmark (blink detection) algorithm by Tereza Soukupová and Jan Cech. Calculating the ratio distance between both horizontal and vertical eye landmarks can lead us when a person blinks. We integrated this with Raspberry Pi 3 Model B computer. We took a great deal while doing this project hoping it will help to reduce the casualties from the road accident.

REFERENCES

[1] Rosebrock, Adrian. "Facial landmarks with dlib, OpenCV, and Python" <> last accessed on 15 November 2018 at 11:05 pm.

[2] Tereza Soukupová and Jan Cech, "Real-Time Eye Blink Detection using Facial Landmarks", 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, February 3-5, 2016

[3] Islam, M Zahidul. "The challenges for safe roads in Bangladesh" <> last accessed on 7 Nov. 2018 at 8.15 pm.

[4] Driver drowsiness detection system, available at <>last accessed on 22

