DRIVER'S DROWSINESS DETECTION

DESIGN PROJECT – 2 REPORT

Submitted by

A.Milan
Hussain(19113106)
Bhanu Sankar Ravi (19113109)
Josaiah W. Lyngdoh N. (19113063)

Under the guidance of

DR R. KRISHNAVENI
Professor

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE
CHENNAI - 603 103

APRIL 2022

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)
CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report **"DRIVER'S DROWSINESS DETECTION"** is the bonafide work of **"A.Milan Hussain (19113106) Bhanu Sankar Ravi (19113109) Josaiah W. Lyngdoh N. (19113063)"** who carried out the project work under my supervision during the academic year 2**021-2022**.

Dr.R.KRISHNAVENI,
SUPERVISOR
PROFESSOR
Department of CSE.

INTERNAL EXAMINER
Name:_____
Designation: _____

EXTERNAL EXAMINER
Name: _____
Designation:_____

Project Viva - voce conducted on _____

CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	
I	**INTRODUCTION**	I -IV
	1.1 Overview	I
	1.2 Motivation	I
	1.3 Goal and Objectives	II
	1.4 Scope and Motivation	II
	1.5 General	II -III
	1.6 Organization of Design Project Report	IV
II	**LITERATURE REVIEW**	IV-V
	2.1 Introduction	IV- V
	2.2 Summary	V
III	**PROJECT DESCRIPTION**	VI
	3.1 Existing System and Disadvantages	VI
	3.2 Proposed system and Advantages	VI
IV	**SYSTEM DESIGN**	VII- X
	4.1 System architecture	VII- VIII
	4.2 Block diagram	XI
	4.3 Use case diagram	X

V	**PROJECT REQUIREMENTS**	XI
	5.1 Software Requirements	XI
	5.2 Hardware Requirements	XI
VI	**MODULE DESCRIPTION**.	XII- XIV
	6.2.1 Module 1	XII
	6.2.2 Module 2	XII
	6.2.3 Module 3	XIV
VII	**IMPLEMENTATION**	XV
VIII	**RESULT AND ANALYSIS**	XVI- XVIII
IX	**CONCLUSION AND FUTURE WORK**	XIX
X	**INDIVIDUAL TEAM MEMBERS REPORT**	XX

REFERENCES
APPENDIX A: SAMPLE SCREEN SHOTS
APPENDIX B: SAMPLE CODE
APPENDIX C: TEAM DETAILS

ACKNOWLEDGEMENT

First and foremost we would like to thank **ALMIGHTY** who has provided us the strength to do justice to our work and contribute our best to it.

We wish to express our deep sense of gratitude from the bottom of our heart to our guide **Dr.R.Krishnaveni, Professor ,Computer Science and Engineering,** for her motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

We would like to extend our heartfelt gratitude to **Dr. J.Thangakumar, Ph.D., Professor & Head, Department of Computer Science and Engineering** for her valuable suggestions and support in successfully completing the project.

We thank the management of **HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE** for providing us the necessary facilities and support required for the successful completion of the project.

As a final word, we would like to thank each and every individual who have been a source of support and encouragement and helped us to achieve our goal and complete our project work successfully.

ABSTRACT

Driver fatigue is one of the major causes of accidents in the world. Detecting the drowsiness of the driver is one of the surest ways of measuring driver fatigue. In this project we aim to develop a prototype drowsiness detection system. This system works by monitoring the eyes of the driver and sounding an alarm when he/she is drowsy.
The system so designed is a non-intrusive real-time monitoring system. The priority is on improving the safety of the driver without being obtrusive. In this project the eye blink of the driver is detected. If the drivers' eyes remain closed for more than a certain period of time, the driver is said to be drowsy and an alarm is sounded.

In this project, we propose and implement a hardware system which is based on infrared light and can be used in resolving these problems. In the proposed method, following the face detection step, the facial components that are more important and considered as the most effective for drowsiness, are extracted and tracked in video sequence frames. The system has been tested and implemented in a real environment.

TABLE NO	TITLE	PAGE NO
3.1	CNN (Convolutional Neural Network)	VII
3.1	EEG (Electroencephalogram)	VII
3.1	EOG (Electroculogram)	VII
3.1	PPG (Photoplethysmography)	VII

CHAPTER 1
INTRODUCTION

1.1 Overview

Driver drowsiness detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads.

The drowsiness detection system is capable of detecting drowsiness quickly.

The driver behaviors are noticed in many conditions such as wearing spectacles and also in the dark condition inside the vehicle.

The system is capable of detecting the drowsiness condition within the duration of more than two seconds. After the detection of abnormal behaviors, it is alerted to the driver through alarms and the parking lights will be on that will stop the vehicle which reduces the accidents due to drowsiness of the driver.

A deep learning Architecture detects the face and eyes, based on the status of the eyes. If the eyes are closed more than usual time, it generates an alarm, intimating the driver.

Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year.

In order to monitor and prevent a destructive outcome from such negligence, many researchers have written research papers on driver drowsiness detection systems. But at times, some of the points and observations made by the system are not accurate enough. Hence, to provide data and another perspective on the problem at hand,
in order to improve their implementations and to further optimize the solution, this project has been done.

1.2 Motivation

Witnessing accidents.

Bringing awareness.

Taking advantage of Technology.

Bringing up Suitable Software.

Include the Software into convenience.

Looking for updating the versions.

1.3 Goal and Objectives

There are many products out there that provide the measure of fatigue level in the drivers which are implemented in many vehicles. Also, it alerts the user on reaching a certain saturation point of the drowsiness measure.

1.4 SCOPE

This project can be implemented in the form of mobile application to reduce the cost of hardware. This project can be integrated with car, so that automatic speed control can be imparted if the driver is found sleeping.

1.5 GENERAL

1.5.1

In this Python project, we will be using OpenCV for gathering the images from webcam and feed them into a Deep Learning model which will classify whether the person's eyes are 'Open' or 'Closed'.

1.5.2

PREREQUISITES

The requirement for this Python project is a webcam through which we will capture images. You need to have Python (3.6 version recommended) installed on your system, then using pip, you can install the necessary packages.

1. **OpenCV –** pip install opencv-python (face and eye detection).
2. **TensorFlow –** pip install tensorflow (keras uses TensorFlow as backend).
3. **Keras –** pip install keras (to build our classification model).
4. **Pygame –** pip install pygame (to play alarm sound).

1.6 ORGANISATION OF DESIGN PROJECT REPORT

A. Milan Hussain	**Bhanu S. Ravi**
Abstract	Introduction
System Requirements	Motivation
Implementation	Objective
Literature Survey	Module Description

Fig. Table 1.6

Josaiah W. Lyngdoh N.	
Existing System with Disadvantages	Proposed System with advantages
Sample Screenshots	Flow Diagram
Result and Demo	Conclusion and Future Work

Fig. Table 1.6

CHAPTER 2
LITERATURE REVIEW

2.1 INTRODUCTION

This survey is done to comprehend the need and prerequisite of the general population, and to do as such, we went through different sites and applications and looked for the fundamental data. Based on these data, we made an audit that helped us get new thoughts and make different arrangements for our task. We reached the decision that there is a need of such application and felt that there is a decent extent of progress in this field too.

There are some research on Driver's Drowsiness Detection, for the proper outcome of the subject and usage of it. The researches use different approaches for the application and the requirement processes.

Drowsiness Detection Based On Driver Temporal Behaviour(31-March 2021, F. Faraji, F. Lotfi, J. Khorramdel, A. Najafi, A. Ghaffari). In this research YOLOv3 CNN is applied as a pretrained network, which is proved to be utilized as a powerful means for object detection.LSTM (Long-Short Term Memory) neural network is employed to learn driver temporal behaviors including yawning and blinking time period as well as sequence classification. One of the main factors of the temporal behavior is that the driver becomes gradually diverted from the road and road traffic. Hence detection is not always accurate.

A Survey on State of The Art Driver Drowsiness Detection Techniques(1st December 2020, FHikmat Ullah Khan).The detection system includes the processes of face image extraction, yawning tendency, blink of eyes detection, eye area extraction etc. The percentage of the eyelid closure of the algorithms over the pupil over time is relatively very low.

Driver Drowsiness Detection(21-09-2020, V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P).The detection system includes the processes of face image extraction, yawning tendency, blink of eyes detection, eye area extraction etc. This paper provides a comparative study on papers related to driver drowsiness detection and alert system. It is designed in such a way where system does not continuously record or retain any data.

Driver Drowsiness Detection System(12 December 2019, Pratyush Agarwal)
This paper analyses the method used to detect driver's drowsiness and proposes the results & solutions on the limited implementation of the various techniques that are used in such embedded systems.

Driver Drowsiness Detection System(May 2019,Muhammad Faique Shakeel and Nabita Bajwa). In this article, they propose a novel deep learning methodology based on Convolutional Neural Networks (CNN) to tackle the Project. In the trained model, we only use 250 low-light images.

A Survey on Driver Drowsiness Detection Techniques(01 December 2020, Reshma , Ishwarya and Sai Vennala). Drowsiness Detection System, the detection system includes the processes of face image extraction, yawning tendency, blink of eyes etc.

2.2 SUMMARY

In this Python project, we have built a drowsy driver alert system that you can implement in numerous ways. We used OpenCV to detect faces and eyes using a haar cascade classifier and then we used a CNN model to predict the status.

CHAPTER 3

PROJECT DESCRIPTION

3.1 EXISTING SYSTEM AND DISADVANTAGES

Vision based techniques:

NO EYE DETECTION – most critical sign of drowsiness Yawning and nodding are not always practical. Varies from person to person – some may not yawn when they are sleepy sometimes.

Physiological sensors:

More accurate solutions

Needs to be attached to the human body

- If driver forgets to wear it?
- May hesitate to wear

Easy wear biosensors are developed.

- Head bands, headphones (Signal-Channel EEG)
- Portal glasses for EOG
- Hand bands for PPG

3.2 PROPOSED SYSTEM AND ADVANTAGES

The proposed system will be continuously monitoring the movement of the driver's eye by a live camera and all the monitored signals are pre - processed.

In order to overcome drawbacks, Python is used in which the trained system is already installed and avoids the time to process that occurs from the scratch.

A Black Box with the software installed is used to detect the driver drowsiness and alerts the driver with buzzer, if driver is affected by drowsiness.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The model we used is built with Keras using **Convolutional Neural Networks (CNN)**. A convolutional neural network is a special type of deep neural network which performs extremely well for image classification purposes. A CNN basically consists of an input layer, an output layer and a hidden layer which can have multiple numbers of layers. A convolution operation is performed on these layers using a filter that performs 2D matrix multiplication on the layer and filter.

The CNN model architecture consists of the following layers:

- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 64 nodes, kernel size 3
- Fully connected layer; 128 nodes

The final layer is also a fully connected layer with 2 nodes. In all the layers, a Relu activation function is used except the output layer in which we used Softmax.

Python Project on Steps for Performing Driver Drowsiness Detection

Download the Python project source code from the zip and extract the files in your system: **Python Project Zip File**

The contents of the zip are:

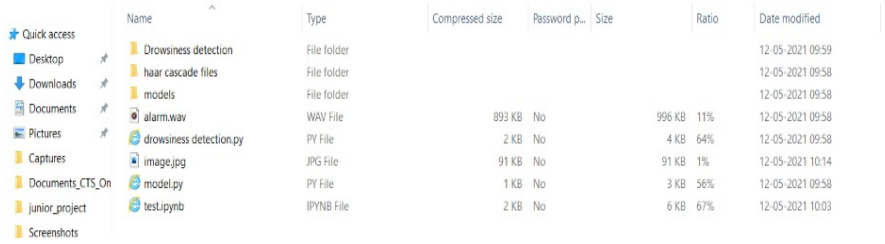


Fig. Table 4.1

1. The “haar cascade files” folder consists of the xml files that are needed to detect objects from the image. In our case, we are detecting the face and eyes of the person.
2. The models folder contains our model file “cnnCat2.h5” which was trained on convolutional neural networks.
3. We have an audio clip “alarm.wav” which is played when the person is feeling drowsy.

4. “Model.py” file contains the program through which we built our classification model by training on our dataset. You could see the implementation of convolutional neural network in this file.
5. “Drowsiness detection.py” is the main file of our project. To start the detection procedure, we have to run this file.

4.2 BLOCK DIAGRAM

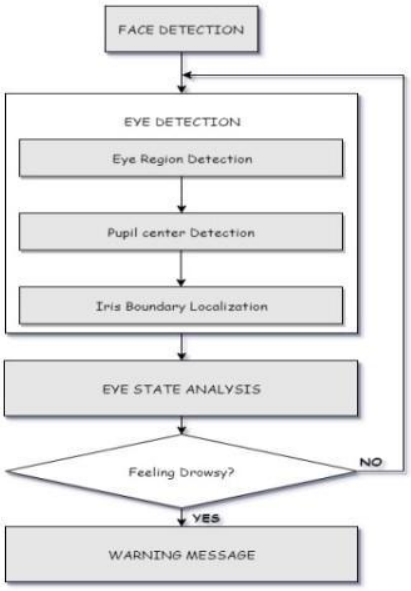


Fig 4.2

The system will be continuously monitoring the movement of the driver’s eye by a live camera and all the monitored signals are pre - processed. A Black Box with the software installed is used to detect the driver drowsiness and alerts the driver with buzzer, if driver is affected by drowsiness.

4.2 USE CASE DIAGRAM

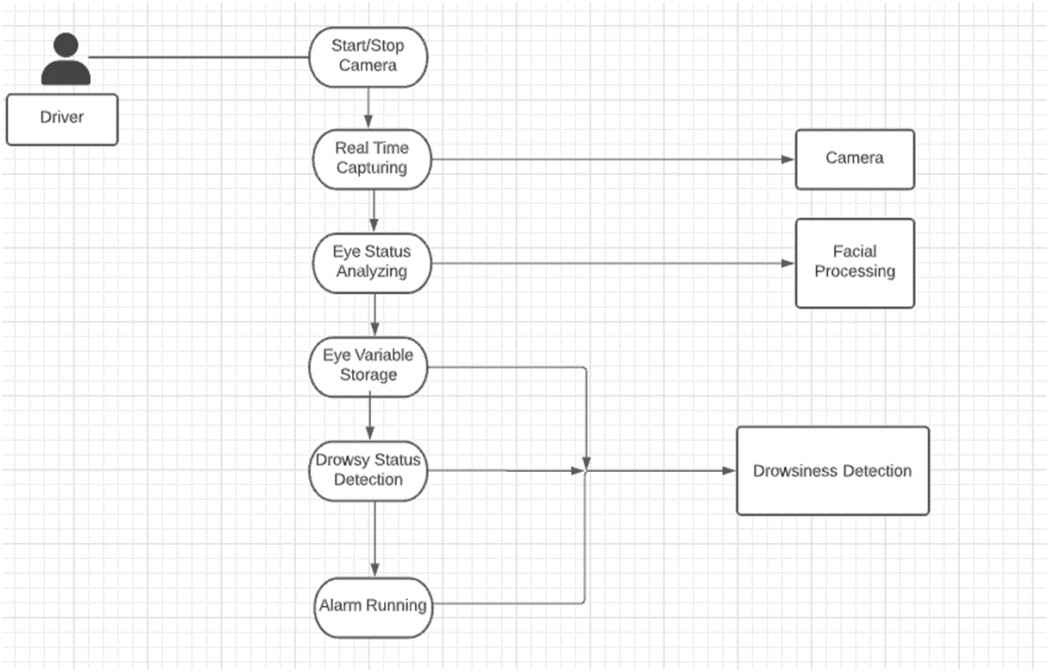


Fig. 4.3

Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving. The project is to build a drowsiness detection system that will detect that a person's eyes are closed for a few seconds. This system will alert the driver when drowsiness is detected. In this Python project, we will be using OpenCV for gathering the images from webcam and feed them into Deep Learning a model which will classify whether the person's eyes are 'Open' or 'Closed'.

CHAPTER 5

PROJECT REQUIREMENTS

5.1 Software Requirements

Python: Python is the basis of the program that we wrote. It utilizes many of the python libraries.

Python 3

Libraries:

Numpy

Tensorflow

opencv,

keras

Pygame.

Operating System:

Windows

5.2 Hardware Requirements

Laptop with basic hardware

Webcam

CHAPTER 6

MODULES DESCRIPTION

Webcam Capturing

Face and Eye Detection

Drowsiness Detection (with Alert Message and Alarm Sound)

6.1 MODULE: 1

With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the image in a frame variable

Fig. 6.1

Camera appears and starts to capture the image.

6.2 MODULE: 2

Face and eye detection:

- The "haar cascade files" folder consists of the xml files that are needed to detect objects from the image (in machine learning). In our case, we are detecting the face and eyes of the person.
- To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier .

- The cascades themselves are just a bunch of XML files that contain OpenCV data used to detect objects. You initialize your code with the cascade you want, and then it does the work for you.
- It loads face cascade into memory and read the mage and convert into grayscale .

(many operations are done in grayscale as we need not define Green , Red and Blue colours here)

Haar cascade approach:

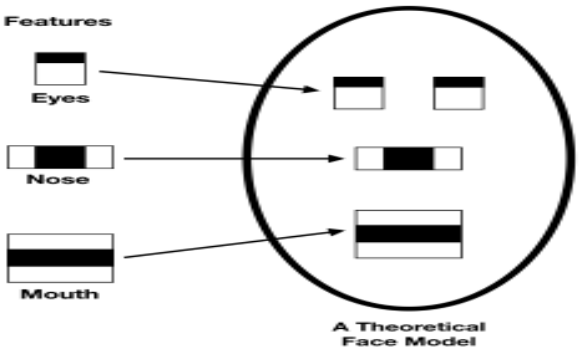


Fig 6.2

Theoretical Face Model for Classification

6.3 MODULE 3 :

Drowsiness Detection

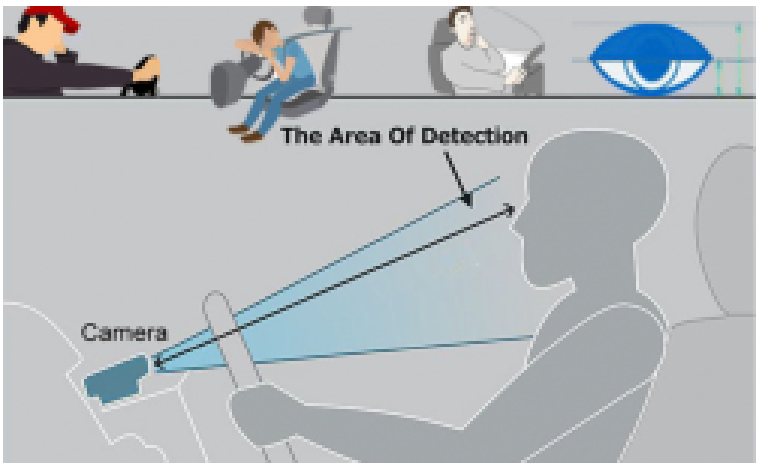


Fig. 6.3

- The value we will use to determine how long the person

has closed his eyes is threshold value. So if both eyes are

closed, we will keep on increasing score and when eyes are

open, we decrease the score. We are drawing the result on the screen

using cv2.putText() function which will display real time status of the person.

- A threshold is defined for example if score becomes greater than 15 that means the person's

eyes are closed for a long period of time. This is when we beep the alarm using sound.play().

- All three of us worked on code

CHAPTER 7

IMPLEMENTATION

1. Take image as input from a camera.
2. Detect the face in the image and create a Region of Interest(ROI).
3. Detect the eyes from ROI and feed it to the classifier.
4. Classifier will categorize whether eyes are open or closed.
5. Calculate score to check whether the person is drowsy.

First we have used a camera which is setup at desirable position in a car that looks for faces stream.

If face gets detected, the facial landmark detection task is applied and region of eyes is extracted.

Once we get the eye region, we calculate the Eye Aspect Ratio to find out if the eye-lids are down for a substantial amount of time.

On the off chance that the Eye Aspect Ratio demonstrates that the eyes are shut for a considerably long measure of time, the alert will sound noisy to wake the driver up.For the functionalities of the system and to make it work efficiently we have used OpenCv, dlib and Python. The implementation of the drowsiness detector system includes machine learning algorithms which are in turn included in OpenCv ML algorithms. There are numerous ML algorithms but for our purpose we required only the face detector algorithm. It works efficiently well overall. It can also be used to detect various different types of objects with the required software

CHAPTER 8

RESULT AND ANALYSIS

RESULT:

In this Python project, we have built a drowsy driver alert system that you can implement in numerous ways. We used OpenCV to detect faces and eyes using a haar cascade classifier and then we used a CNN model to predict the status.

ANALYSIS:

Opened Eyes Detection:

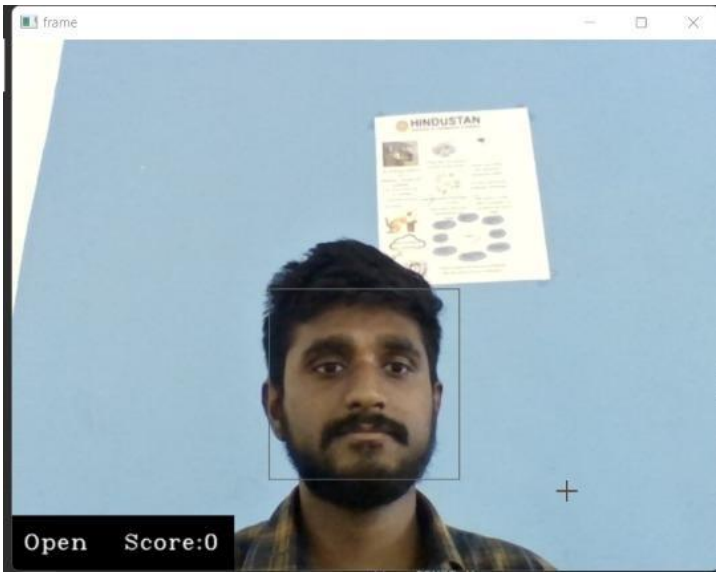


Fig 8.1 (a)

Fig 8.1 (b)

Closed Eyes Detection :

Sleep Alert :

Fig 8.1 (c)

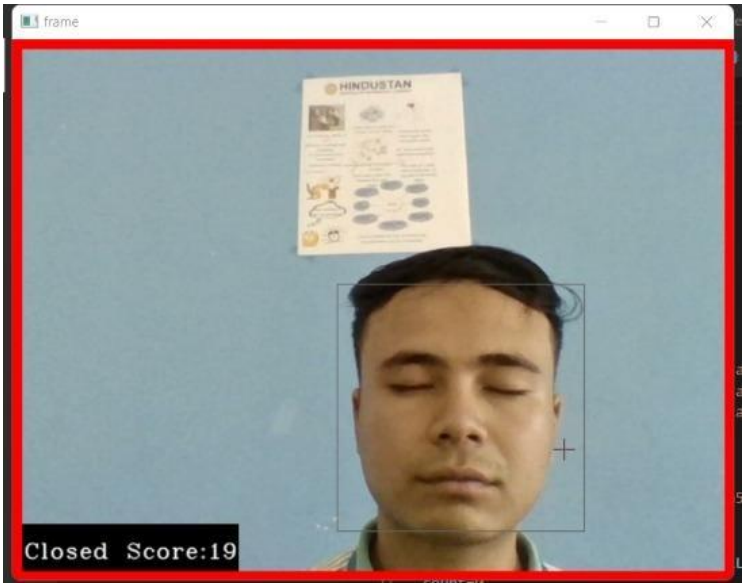


Fig 8.1 (d)

CHAPTER 9

CONCLUSION AND FUTURE WORK

CONCLUSION:

It completely meets the objectives and requirements of the system. The framework has achieved an unfaltering state where all the bugs have been disposed of. The framework cognizant clients who are familiar with the framework and comprehend it's focal points and the fact that it takes care of the issue of stressing out for individuals having fatigue-related issues to inform them about the drowsiness level while driving.

The ultimate goal of the system is to check the drowsiness condition of the driver. Based on the eye movements of the driver, the drowsiness is detected and according to eye blink, the alarm will be generated to alert the driver and to reduce the speed of the vehicle along with the indication of parking light. By doing this, many accidents will be reduced and provides safety to the driver and vehicle. A system that is driver safety and car security is presented only in luxurious costly cars. Using eye detection, driver security and safety can be implemented in normal car also.

FUTURE WORK:

The model can be improved incrementally by using other parameters like blink rate, yawning, state of the car, etc. If all these parameters are used it can improve the accuracy by a lot.

We plan to further work on the project by adding a sensor to track the heart rate in order to prevent accidents caused due to sudden heart attacks to drivers.

Same model and techniques can be used for various other uses like Netflix and other streaming services can detect when the user is asleep and stop the video accordingly. It can also be used in application that prevents user from sleeping.

CHAPTER 10

INDIVIDUAL TEAM

MEMBERS REPORT

Bhanu S. Ravi – As a member and leader of the team, I have contributed especially to the objective of this project, also I have implemented my skills and contributed to the 2nd module of the project, a short summary of the module is to detect the face in the image, we first converted the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input since we don't need color information to detect the objects. We used haar cascade classifier to detect faces and which is used to set our classifier.

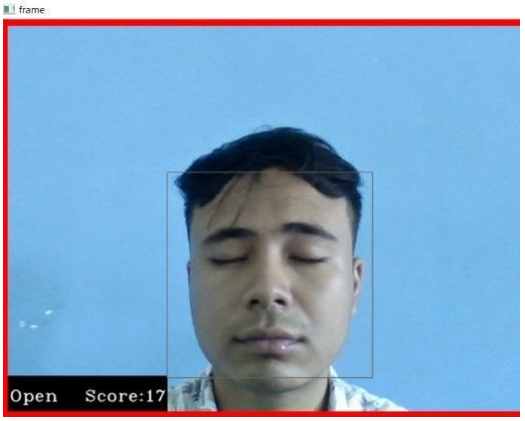
A.Milan Hussain – Being a member of the team, I have contributed to implementing the project as well looking into the literature review of the project. I have contributed also in working with the 1st module of this project where accessing the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the image in a frame variable

Josaiah W. Lyngdoh N. – With the help of my fellow team members, I have been able to proposed a new system with advantages in comparison to that of the old and existing system. I also have contributed in working with the 3rd module of the project wherein a value is use to determine how long the person has closed his eyes (threshold value).So, if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using cv2.putText() function which will display real time status of the person. The offset threshold score is greater than 15.

REFERENCES

- COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHLKOPF-A. BLAKE, S. ROMDHANI, AND P. TORR.
- USE OF THE HOUGH TRANSFORMATION TO DETECT LINES AND CURVES IN PICTURE; R. DUDA AND P. E. HART.
- JAIN, “FACE DETECTION IN COLOR IMAGES; R. L. HSU, M. ABDEL-MOTTALEB, AND A. K. JAIN.
- OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI.
- http://ncrb.gov.in/StatPublications/ADSI/ADSI2015/chapter1A%20traffic%20accidents.pdf
- http://www.jotr.in/text.asp?2013/6/1/1/118718
- http://dlib.net/face_landmark_detection_ex.cpp.html

APPENDIX A: SAMPLE SCREEN SHOTS



APPENDIX B: SAMPLE CODE

DETECTION : (detection.py)

```
import cv2
import os
from keras.models import load_model
import numpy as np
from pygame import mixer
import time
import tensorflow as tf

mixer.init()
print(os.getcwd())
sound = mixer.Sound('alarm.wav')

face = cv2.CascadeClassifier('haar cascade files\haarcascade_frontalface_alt.xml')
leye = cv2.CascadeClassifier('haar cascade files\haarcascade_lefteye_2splits.xml')
reye = cv2.CascadeClassifier('haar cascade files\haarcascade_righteye_2splits.xml')

lbl=['Close','Open']

model = load_model('models/cnnCat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
```

```
score=0
thicc=2
rpred=[99]
lpred=[99]

while(True):
    ret, frame = cap.read()
    height,width = frame.shape[:2]

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) ,1
        except: # isplaying = False
            pass
        if(thicc<16):
            thicc= thicc+2
        else:
            thicc=thicc-2
            if(thicc<2):
                thicc=2
        cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

TEST : (test.ipynb)


```
{
 "metadata": {
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.7.10"
  },
  "orig_nbformat": 2,
  "kernelspec": {
   "name":
"python3710jvsc74a57bd0fd034ed54560f0698c2946b7ca675e493afbd7ee3c0ecf162a
e3deac3cf4477b",
   "display_name": "Python 3.7.10 64-bit ('pytorch': conda)"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2,
 "cells": [
```

```
{
 "cell_type": "code",
  "    for (x,y,w,h) in faces:\n",
  "        cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) ,1)\n",
  "\n",
  "        if(rpred[0]==0):\n",
  "            lbl='Closed'\n",
  "        break\n",
  "\n",
  "    for (x,y,w,h) in left_eye:\n",
  "        l_eye=frame[y:y+h,x:x+w]\n",
  "        count=count+1\n",
  "        l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)\n",
  "        l_eye = cv2.resize(l_eye,(24,24))\n",
  "        l_eye= l_eye/255\n",
  "        l_eye=l_eye.reshape(24,24,-1)\n",
  "        l_eye = np.expand_dims(l_eye,axis=0)\n",
  "        lpred = model.predict_classes(l_eye)\n",
  "        if(lpred[0]==1):\n",
  "            lbl='Open'\n",
  "        if(lpred[0]==0):\n",
  "            lbl='Closed'\n",
  "        break\n",
  "\n",
{
 "cell_type": "code",
"execution_count": null,
"metadata": {},
```

```
  "outputs": [],
  "source": []
```

MODEL : (Model.py)

```
import os
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
from keras.utils.np_utils import to_categorical
import random,shutil
from keras.models import Sequential
from keras.layers import Dropout,Conv2D,Flatten,Dense, MaxPooling2D,
BatchNormalization
from keras.models import load_model


def generator(dir, gen=image.ImageDataGenerator(rescale=1./255),
shuffle=True,batch_size=1,target_size=(24,24),class_mode='categorical' ):

   return
gen.flow_from_directory(dir,batch_size=batch_size,shuffle=shuffle,color_mode='gra
yscale',class_mode=class_mode,target_size=target_size)

BS= 32
TS=(24,24)
train_batch= generator('data/train',shuffle=True, batch_size=BS,target_size=TS)
valid_batch= generator('data/valid',shuffle=True, batch_size=BS,target_size=TS)
SPE= len(train_batch.classes)//BS
VS = len(valid_batch.classes)//BS
```

```
print(SPE,VS)


# img,labels= next(train_batch)
# print(img.shape)

model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(24,24,1)),
    MaxPooling2D(pool_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
#32 convolution filters used each of size 3x3
#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),

#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy']
)
model.fit_generator(train_batch,
validation_data=valid_batch,epochs=15,steps_per_epoch=SPE ,validation_steps=VS)
 model.save('models/cnnCat2.h5', overwrite=True)
```

APPENDIX C: TEAM DETAILS

Bhanu S. Ravi. – The team's leader and an undergraduate B.Tech student in the field of Computer Science belonging to Hindustan Institute of Technology & Science, currently in his 3rd year bearing the roll no 19113109.

A.Milan Hussain – An undergraduate B.Tech student in the field of Computer Science belonging to Hindustan Institute of Technology & Science, currently in his 3rd year bearing the roll no 19113106.

Josaiah W. Lyngdoh N. – An undergraduate B.Tech student in the field of Computer Science belonging to Hindustan Institute of Technology & Science, currently in his 3rd year bearing the roll no 19113063.