A Project Report

on

Drowsiness Detection System Using Computer Vision

Submitted in partial fulfillment of the
requirement for the award of the degree of

Bachelors of Technology

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr.Jayakumar V.
Assistant Professor

Submitted By

Shashwat Mishra
18SCSE1140052/18021140098
Suraj Kumar Karna
18SCSE1140049/18021140095

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA INDIA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"DROWSINESS DETECTION SYSTEM USING COMPUTER VISION"** in partial fulfillment of the requirements for the award of the Bachelors of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July 2021 to December 2021, under the supervision of Mr.Jayakumar V. Assistant Professor, Department of Computer Science and Engineering,School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Shashwat Mishra, 18SCSE1140052

Suraj Kumar Karna, 18SCSE1140049

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Jayakumar V.

Assistant Professor

CERTIFICATE

The Final Project Viva-Voce examination of Shashwat Mishra:18SCSE1140052 and Suraj Kumar Karna:18SCSE1140049 have been held on _____ and his/her work is recommended for the award of Computer Science and Engineering with Specialization on Computer Network and Cyber Security

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Abstract

Feeling sleepy in inappropriate places like while driving is very dangerous and in places like offices and schools, it becomes a waste of time for the employers and teachers for their hard work they're putting on their minors and the minors not looking after or following the peers.

Nothing officially has been made to come to be known widely about a system software that detects drowsiness or sleepy behavior in unlikely times. This has provided us with the urge to develop such a software that can detect such behavior and sound an alarm or alert whenever this is caught in the eyes of the computer cameras.

We are going to develop this system with the help of Python and it's large variety of modules that includes OpenCV, i.e.,Computer Vision, TensorFlow, Keras and Pygame.

This can help even employees to not waste too much time on unplanned naps or so. It can save so many lives from road accidents as we have come to know that most of the accidents take place because the driver became drowsy while they were driving. This can prevent all those misfortunes from happening and can be a real game changer for normal people as well as people connected to the corporate world.

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

Table of Contents

CHAPTER-1

Introduction

Drowsiness is the state where person have strong desire to sleep. It is the state where person has the high desire to sleep. It has two definite meanings, referring both to the state foregoing falling asleep and chronic condition referring to being in that state independent of the daily rhythm. While performing the tasks that needed the constant concentration such as driving can be dangerous if it is done in a drowsiness state. Person can experienced drowsiness if they had a sufficient fatigue and this can leads to road accidents.

The major challenge in the field of accident avoidance system faced by the developer and researcher is in developing the technologies for detecting or preventing drowsiness among the drivers. Solutions need to be done in order to counteract the presence hazards of drowsiness on a road.

The purpose of this project is to develop the simulation of drowsiness detection system. The focus of the project is to design a system that will detect the drowsiness by detecting the closed eyes of the driver. By monitoring the state of the eyes, it is believed can detect the

early symptom of the driver's drowsiness, to avoid car accidents. The process of detecting the drowsiness between drivers is to detect the open and closed of the eyes.

Sometimes, people needs an alertness and warn to prevent them from sleeping while driving. There are plenty driver drowsiness detection system were developed by many researcher. One of the example is the use a computer vision or OpenCV to detect image or video processing. This alternative will help developer to developed computer vision based system.

The purpose of the project is to design the simulation of drowsiness detection system. The system will accurately monitor the state of driver's eyes whether it is open or closed. The road accidents believed can be avoided by monitoring the early stage of drivers' drowsiness.

Face recognition is one of the application besides human identification and tracking for security systems becomes popular research area by analysing the face images. This project is focused on the localization of the eyes, which involves the both eyes and face by applying the existed image-processing algorithm. The detection of the drowsiness will be determined once the position of the eyes located.

Formulation of Problem

With this Python project, we will be making a drowsiness detection system. A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy.

Proposed Solution

Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.

The objective of this intermediate Python 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.

Tools and Technology Used

Basic structure will be made using Python. Framework like Open CV will used to support the backend and database of app. Keras will used to create the whole application with coding of each and every module.

Required Tools Software and Hardware Requirements:

Software Requirements:

- Python3.5+
- Visual Studio Code
- Chrome
- Anaconda

- TensorFlow
- Keras
- NLP Libraries

Hardware Requirements:

- OS: Windows / Linux/ Mac
- Processor: intel i5
- RAM: 4GB
- ROM: 500GB
- Graphic Card: Good but Not Necessary

Chapter 2

Literature Survey

2.1 INTRODUCTION

There are many research paper that focusing on detecting the drowsiness among the drivers that developer can refer in order to develop the drowsiness detection system. Literature review focusing on what have been done by other authors.

2.2 RESEARCH OF STUDIES

2.2.1 Automatic Driver Drowsiness Detection and Accident Prevention System using Image Processing.

In this project, sleep is a primary human need. Basically, human needs to sleep at least eight hours a day by schedule for the day. If someone needs adequate relief the frame does not have a proper characteristic (function). The first to do so drivers all need enough sleep to boost their work. If we don't sleep enough, we might get drowsy and fall asleep. A few seconds of drowsiness will also cost lives for both drivers and travelers. This

gadget gives you a vibrant monitoring assistant who analyses blinking of the drivers' eyes action and, in addition, portion of the mouth examining whether driving is yawning

2.2.2 Understanding of a Convolutional Neural Network

In this paper[1], deep learning or deep neural network refers to Artificial Neural Networks (ANN) with multiple layers. It has been considered one of the most recent decades. Most efficient devices, and has been very popular literature as it is capable of processing a large volume of data. This is the interest in having deeper hidden layers has recently begun. Surpass the performance of classical methods in different fields; especially in recognition of patterns.

One of the deepest, most popular the neural networks are the Convolutional Neural Network (CNN). Taking its name from the mathematical linear operation between matrixes are called convolution. CNN has several layers, like a convolutionary layer, a non-linear layer, a pooling layer and a completely connected layer. Convolutionary and fullyconnected layers have parameters but there are no pooling and non-linear layers its' parameters. CNN has an outstanding computer efficiency things of learning. Applications

dealing with, in particular, image data, such as the largest image classification data set (Image Net), computer vision and natural language processing (NLP) and the results obtained have been very wonderful. This paper discusses and describes all the elements and issues of significance related to CNN, and how it works. In addition, we are this will also state the parameters that have an impact on CNN performance. This paper assumes that readers have sufficient knowledge of both machine learning and artificial neural networks.

2.2.3 A Survey Paper on Drowsiness Detection & Alarm System for Drivers

In this research paper[2], our safety is the first priority while traveling or driving. A driver's mistake can lead to a serious mistake. Physical injury, mortality and major economic losses. There are several applications available on the market today, such as navigation devices**,** multiple sensors, etc. Easy to work. There are numerous reasons, particularly human’s faults that give rise to road accidents. Studies say that there is a huge rise in the number of road accidents in our country has been in existence for the last few years. The key explanation that is happening on the highway injuries are somnolence and sleepiness driver as he's driving.

2.2.4 Real-time Drowsy Driver Detection using Haarcascade Samples

In this paper, with population increase, car injuries have also occurred growth. A thorough study reveals that there are over half of a million incidents in India per year. However, owing to driver exhaustion, about 60 percent of such injuries are induced. Tired rider influences driving skill in the following 3 regions, (a) hinder balance, (b) affect slower driving abilities reaction cycles, and, c) Perception loss. We have a real-time via this paper monitoring device of image recognition technologies, facial / head identification. Even, to ensure real-time computation, Haarcascade tests are used to discern between a blink of the eye and detection in drowsiness / exhaustion.

2.2.5 Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network

This research paper[3], the growth in car crash deaths has risen rapidly across the world in recent years. Nevertheless, road security has been a global problem and a complex topic that needs to be tackled. Deaths from traffic accidents are also on the rise and are actually deemed a major general medical issue. The most recent advances in promoting research and information Vehicle skills, allowing drivers to see and analyse conditions in the streets to prevent collisions and secure travelers. Analysing driver activities on the road has therefore been one of the leading research topics in recent years, particularly somnolence, as it provides the highest mishaps impact and is the main cause of death on the highways. This paper includes an overview and predict driver drowsiness by running

a Recurrent Neural Network over a sequence frame eye to the rider. We used a dataset to mold and authorize our concept and provide repeated implementations Neural Network Multi-Layer 3D Convolutionary Networks for Driver Detection Drowsiness. The accuracy of acceptance rate is 92% after the training session.

2.2.6 An Efficient K–NN Approach for Automatic Drowsiness Detection Using Single–Channel EEG Recording

The author said[4], drowsy driving is a major source of many traffic accidents. The purpose of this research is the automated creation of detection method for drowsiness use an effective k-nearest neighbors (K-NN) algorithm. Next, power distribution in time-frequency space was computed using short-term Fourier transform (STFT), and then the mean power value was determined for each EEG subband across time-segments of 0.5 second. In addition, the time-domain was calculated for standard deviation (SD) and Shanon entropy linked to increasing time-segment. Ultimately, they removed 52 elements. Random forest algorithm was added to the extracted data in order to pick the most helpful apps subpackage. To identify drowsiness and alertness a minimum of 11 apps have been chosen. Kd-trees are used as a search algorithm for nearest neighbors in order to provide a fast classifier. The experimental findings demonstrate that the techniques and materials introduced in this paper can be used to reliably diagnose drowsiness with 91% precision

2.2.7 Target Recognition in Infrared Circumferential Scanning System via Deep Convolutional Neural Networks

In this article[5], with an IRCSS (Infrared Circumferential Scanning System) we will realize long-term oversight over a wide field of vision. Automatically identifying goals inside the field of vision is a critical part of increasing environmental consciousness in the computerisation cycle, particularly with the framework of protection. Recognition of objectives consists of two subtasks: identification and identification, which refers to the goal location and type, respectively. In the analysis, we propose a Deeply Convolutionary Neural Network (DCNN)-based method for end-to - end realization recognition of goal in IRCSS. Current DCNN-based approaches need a broad data collection, annotated for instruction, while public infrared databases are often used to monitor objectives. Hence, we create an infrared object recognition dataset to both solve data limitations and boost the algorithm's adaptability in various scenes. We can using data raise and take advantage of optimum cross-domain software transition technique for Network Learning. Within this step we implement the smoother L1 as the loss function for improved localization efficiency within bounding box regression. The proposed approach obtained 82.7 mAP in the tests, achieving end-to - end identification of the infrared target with high efficiency on accuracy

2.2.8 An Investigation of Early Detection of Driver Drowsiness Using Ensemble Machine Learning Based on Hybrid Sensing

What can I conclude from this paper[6], drowsy driving is one of the most important sources of road collisions. Reducing these injuries, it is essential to detect drowsy driving early. It has been demonstrated, in previous research, that driver drowsiness impacted driving performance, conduct indices and physiological indices. This is for the purpose the research would investigate the feasibility of classifying driver warning systems, in particular the slightly somnolent, dependent on engine, behavioral, and physiological hybrid sensing indicators of concern for the analysis of such identification setup. Next, we assessed the degree of drowsiness, driving efficiency and physiological signal (From the analysis of electroencephalograms and electrocardiograms) and the conductor's behavioural measures driving simulator and control panel for passengers. Therefore, driver alarm and drowsy conditions have been established using machine learning algorithms and constructing a catalogue of the indices derived from a duration 10 s. Ultimately, description of the ensemble algorithms was used. Reports suggested the ensemble algorithm will achieve 82.4 per cent classification accuracy using hybrid warning recognition methods yet mildly somnolent, 95.4 percent precision classifying alarm and somewhat drowsy.

2.2.9 Non-intrusive driver drowsiness detection system.

The main contribution of this study is a novel algorithm for drowsiness detection and tracking, which is based on the incorporation of information from a road vision system and vehicle performance parameters. Refinement of the algorithm is more specifically identified the degree of drowsiness by the implementation of a service computer vector detection with a reliable and effective alarm method with drowsiness. By utilizing non-intrusive devices with regular equipment sensors, the Support Vector Machine (SVM) detection methodology decreased drowsiness level to eliminate such road injuries triggered by drowsiness users. It identification device includes a non-contact tool for assessing various forms of driver alertness rates and enables early identification of a decrease in driving alertness. The findings reported are focused on a collection of somnolence databases which cover approximately 60 hours driving hours to assess data collection. Everything system parameters derived data on parameters are obtained in a driving simulator. A true car with all the equipment, a model for detecting drowsiness in SVM is being developed. The grouping, after many changes for certain methods, the findings gave a very strong sign of drowsiness.

2.2.10 Prediction of Drowsy Driver Detection by Using Soft Computing Technique

This research[7] defines a system for automated drowsy drivers and crash avoidance focused on adjustments in the facial expressions. The central character the explanation for the road injuries may be the amount of years of driving. Face expression check will include the

driver's drowsiness assessment to insure the driver is vigilant. Hence, the research discusses the solution to recognizing vehicle drowsiness. We realize our methodology by taking the driver's face picture, searching for facial features by handling images and using hybrid strategy for assessing the degree of driver drowsiness.

Chapter 3

Functionality

3.1 INTRODUCTION

The research methodology is essential to ensure the research objectives can be achieved. This chapter will explain in detail regarding the methods used during conducting this project. In order to make sure the project is in the right path, methodology plays an important role as a guide for the project to complete and working well as planned. This project is involves many software such as Anaconda Navigator, Sublime text editor and etc.

3.2 ALGORITHM

In earlier stages of this final year project, it has been intended to use OpenCV to develop the algorithm for the system. Using OpenCV, the software have been used to ease the process of writing the codes. In the experimentation result, eyes has been done.

3.3 SYSTEM REQUIREMENT AND SPECIFICATION

System requirement is needed to accomplish this project and assist the development of the project that involves system requirement in hardware and software. Each of the requirement is related to each other to make sure that the system can be done smoothly.

3.3.1 HARDWARE

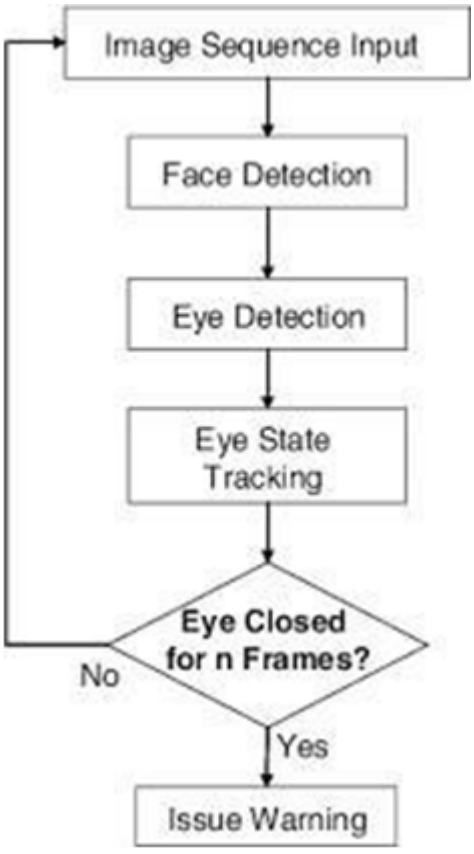
No	Hardware	Description
1	Laptop (Acer Aspire E 14)	Processor: Intel Core i5 RAM: 4 GB OS Version: Windows 32/64 bit
2	Webcam	Built-in webcam

3.3.2 SOFTWARE

No.	Software	Description
1	Microsoft Word 2019	Microsoft Word used for word processing, such as crating and editing report and thesis documentation
2	Microsoft PowerPoint 2019	To present the result and the findings of this project
3	Snipping Tool	Capture and screenshot images
4	Anaconda Navigator	Launch applications and easily manage conda packages, environments, and channels without using command-line commands.
5	Python 3.8	High-end language interpreter to program the project
6	WinZip	To extract the zipped data

3.4 FRAMEWORK

The core of this research is to develop an algorithm that can be used in a software system based on driver drowsiness detection system.



3.5 DATA SOURCE

Data source is the location where we can get data that has been accessible to others and that has been provided by a trustworthy platform based on the computer science. The primary database source is the database that can be located on a disk or remote server in a database management system. I use the collected data that has been uploaded into the trustworthy website.

3.6 Relevant code:

Home-

```
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='grayscale',
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)),

#64 convolution filters used each of size 3x3
#choose the best features via pooling

#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
#flatten since too many dimensions, we only want a classification output
    Flatten(),
#fully connected to get all relevant data
    Dense(128, activation='relu'),
#one more dropout for convergence' sake :)
    Dropout(0.5),
#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)
```

Model-

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


mixer.init()
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)

    faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
    left_eye = leye.detectMultiScale(gray)
    right_eye =  reye.detectMultiScale(gray)

    cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0,0) , thickness=cv2.FILLED )
```

```
for (x,y,w,h) in faces:
    cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )

for (x,y,w,h) in right_eye:
    r_eye=frame[y:y+h,x:x+w]
    count=count+1
    r_eye = cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)
    r_eye = cv2.resize(r_eye,(24,24))
    r_eye= r_eye/255
    r_eye=  r_eye.reshape(24,24,-1)
    r_eye = np.expand_dims(r_eye,axis=0)
    rpred = model.predict_classes(r_eye)
    if(rpred[0]==1):
        lbl='Open'
    if(rpred[0]==0):
        lbl='Closed'
    break

for (x,y,w,h) in left_eye:
    l_eye=frame[y:y+h,x:x+w]
    count=count+1
    l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
    l_eye = cv2.resize(l_eye,(24,24))
    l_eye= l_eye/255
    l_eye=l_eye.reshape(24,24,-1)
    l_eye = np.expand_dims(l_eye,axis=0)
    lpred = model.predict_classes(l_eye)
    if(lpred[0]==1):
        lbl='Open'
    if(lpred[0]==0):
        lbl='Closed'
    break

if(rpred[0]==0 and lpred[0]==0):
    score=score+1
    cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
# if(rpred[0]==1 or lpred[0]==1):
else:
    score=score-1
    cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)


if(score<0):
```

```
        score=0
    cv2.putText(frame,'Score:'+str(score),(100,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
    if(score>15):
        #person is feeling sleepy so we beep the alarm
        cv2.imwrite(os.path.join(path,'image.jpg'),frame)
        try:
            sound.play()

        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()
```

Chapter 4
Result and Discussion

4.1 INTRODUCTION

This chapter will discuss about the project implementation and testing. Both implementation and testing result are the last stage of the project development. Implementation is necessary to verify that the project development of the trained model meet the requirement. Testing result is the process of showing the final result of the testing that have been done to ensure its functionality. At this phase, it will show the model is well functioned and identify any weaknesses to be improved later on. So, this chapter will generally discuss the implementation, deployment and testing of the entire project after being developed.

4.2 IMPLEMENTATION

All the implementation process of the drowsiness detection by using computer vision project will be presented.

4.2.1 Create and activate the virtual environment for compatible version that support the all packages needed.

4.2.2 Setting up the environment of the computer vision algorithm

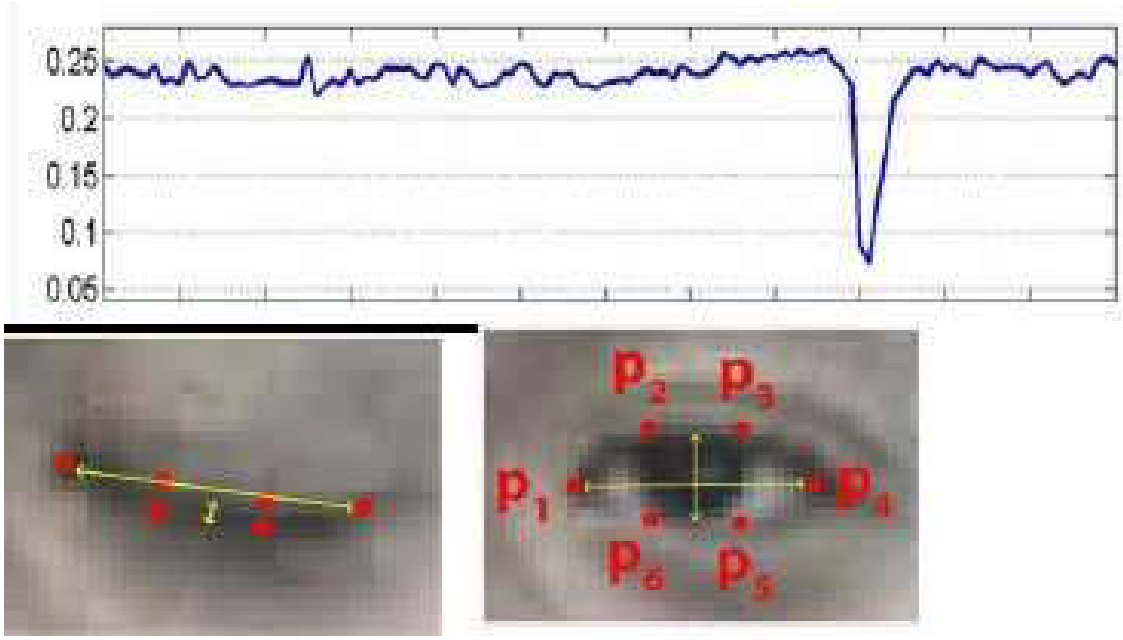
The installation of the python and pip must be made in order to run this system.

The packages need to be installed in this project is as below

- Imutils
- Dlib
- Cv2
- Scipy

4.3 Result

After the modelling is completed, the simulation was performed by using the real-time simulation using webcam



Chapter-5
Conclusion and Future Scope

In this chapter, the authors can conclude that, this project can be improved in the future to make it better and outstanding. Several techniques to develop the simulation system have been discovered. There are also other objectives that this project needs to achieve.

5.1 Research phase

This project is divided into four phases in order to fulfil the objective. The first are the preliminary study and literature review that study the previous research or works.

Next, the design and methodology phase which are system requirement, model and algorithm. The third phase is implementing, testing and expected result. Lastly, the conclusion of this project will be summarized that conclude the whole project.

5.2 Future Work and Recommendation

The main goal of this project is to detect the states of the drivers' eyes to determine they are in the state of drowsy or not. The system must meet certain requirement which is detecting drowsiness and alert the driver as well accurately.

Improvement on the algorithms to detect eyes and mouth need to be done for future implementation. Luminance changes have to be encounter to ensure the detection of the gradient of eyes is sufficient to improve the detection results. The quality of the video or images used in detecting drowsiness affects the result of the detection. Therefore, a good quality and high frame rate of images (number of pixel) is one of the factors to get better detection. Better techniques can be used to compare which technique is more reliable in detecting drowsiness.

Thus, by making this project successful, the numbers of road accident can be reduce when this project is implemented in the vehicle to detect the drowsiness of the driver.

REFERENCES

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, no. April 2018, pp. 1–6, 2018.

[2] P. Choudhary, R. Sharma, G. Singh, S. Das, and S. G. Dhengre, "A Survey Paper On Drowsiness Detection & Alarm System for Drivers," *Int. Res. J. Eng. Technol.*, pp. 1433–1437, 2016.

[3] Y. Ed-Doughmi, N. Idrissi, and Y. Hbali, "Real-time system for driver fatigue detection based on a recurrent neuronal network," *J. Imaging*, vol. 6, no. 3, 2020.

[4] A. Jalilifard and E. B. Pizzolato, "An efficient K-NN approach for automatic drowsiness detection using single-channel EEG recording," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2016-Octob, pp. 820–824, 2016.

[5] G. Chen and W. Wang, "Target recognition in infrared circumferential scanning system via deep convolutional neural networks," *Sensors (Switzerland)*, vol. 20, no. 7, pp. 1–18, 2020.

[6] J. Gwak, A. Hirao, and M. Shino, "An investigation of early detection of driver drowsiness using ensemble machine learning based on hybrid sensing," *Appl. Sci.*, vol. 10, no. 8, 2020.

[7] M. P. Singh, G. Srivastava, and P. Kumar, “Internet traffic classification using machine learning,” *Int. J. Database Theory Appl.*, vol. 9, no. 12, pp. 45–54, 2016.

[8] Abro,S., Shaikh,S., Khand,H,Z.,Ali,Z., Khan,S., Mujtaba,G., Automatic Hate Speech Detection using Machine Learning: A Comparative Study. International Journal of Advanced Computer Science and Applications (IJACSA),2020, Volume 11, Issue 8.

[9] Burnap, P. and M.L. Williams, Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Science, 2016. 5(1): p. 11.

[10] Cavnar, W.B. and J.M. Trenkle. N-grambased text categorization. in Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retreval. 1994. Citeseer.

[11] Davidson, T., Warmsley, D., Macy,M., and Weber, I.. Automated hate speech detection and the problem of offensive language. arXiv preprint arXiv:1703.04009, 2017.

[12] Fortuna, P. and S. Nunes, A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 2018. 51(4): p. 85.

[13] Kwok,I.,andWang,Y. 2013. Locate the hate: Detecting tweets against blacks. In AAAI.

[14] Mujtaba, G., et al., Prediction of cause of death from forensic autopsy reports using text classification techniques: A comparative study. Journal of forensic and legal medicine, 2018. 57: p. 41-50.