ANJUMAN-I-ISLAM'S

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

(Affiliated to University of Mumbai)

BYCULLA, MUMBAI-400008

A MINI-PROJECT REPORT

ON

"Drowsiness Detection System using Python"

Submitted by

Mohamed Adib Shaikh	5118027
Rizwana Khatoon Siddiqui	5117030
Nomaan Maqsud Patvi	5219063

Department of Electronics & Telecommunication Engineering

M. H. Saboo Siddik College of Engineering

2020-2021

ANJUMAN-I-ISLAM'S

M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

BYCULLA, MUMBAI-400008

Department of Electronics & Telecommunication Engineering

CERTIFICATE

Certified that the mini-project work entitled **“Drowsiness Detection System using Python”** is a bonafide work carried out by

Mohamed Adib Shaikh	**5118027**
Rizwana Khatoon Siddiqui	**5117030**
Nomaan Maqsud Patvi	**5219063**

The report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed for the course **Image processing and machine vision**

………………………………. …………………………

Dr. Ravindra Patil **Er. Abdul Sayeed**

Mini-Project Coordinator Head of the Department

CONTENTS

ACKNOWLEDGMENT

We give my sincere thanks to our guide **Dr. Ravindra Patil** for his guidance and constant support, whenever we needed it. We also extend thanks to my teaching staff in the department whom we approached for academic help for our project. We also thank the mini project coordinators and the non-teaching staff as well for arranging the necessary facilities to carry out the mini project work. We are also highly grateful to Head of Department (EXTC), and Principal and Director for providing the facilities, conducive environment and encouragement.

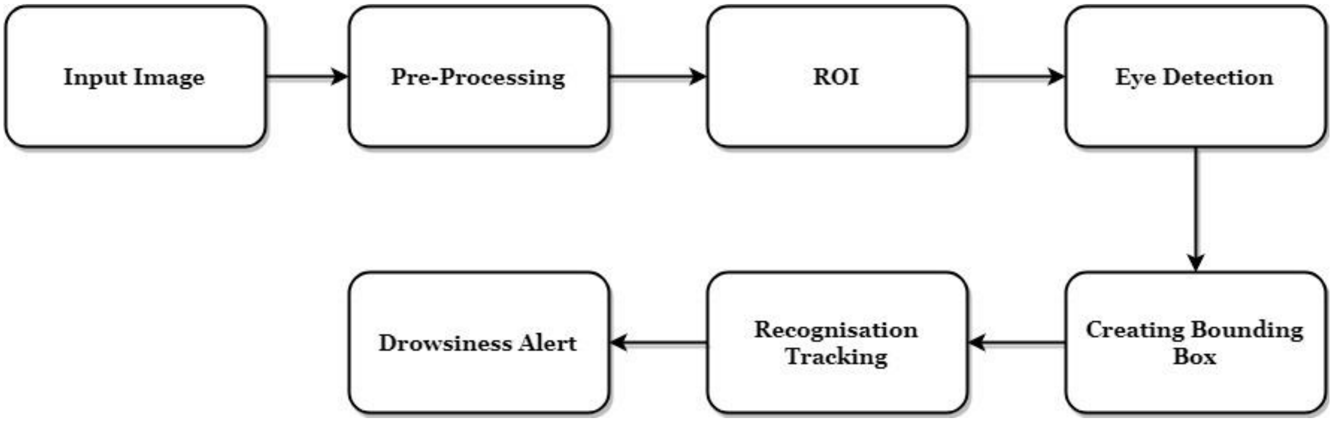
2. ABSTRACT:

Drivers must keep a close eye on the road, so they can react to sudden events immediately. Driver fatigue often becomes a direct cause of many traffic accidents. Therefore, there is a need to develop the systems that will detect and notify a driver of her/him bad psychophysical condition, which could significantly reduce the number of fatigue-related car accidents. However, the development of such systems encounters many difficulties related to fast and proper recognition of a driver’s fatigue symptoms. One of the technical possibilities to implement driver drowsiness detection systems is to use the vision-based approach. Here we are detecting the driver drowsiness by estimating vision system of him.

3. INTRODUCTION:

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. It overcomes the problem related to the accidents. Drivers experiencing fatigue leads to a need arises to design a system that keeps the driver focused on the road. Therefore, many designs and prototypes have been implemented in automobiles to avoid such accidents by keeping the whole focus and concentration on accurately monitoring the open and closed state of the driver’s eye in real time. Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.

4. IMPLIMENTATION:



Step 1 – Take Image as Input from a Camera

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.

Step 2 – Detect Face in the Image and Create a Region of Interest (ROI)

To detect the face in the image, we use the OpenCV algorithm for object detection. We don’t need color information to detect the objects. It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object.

Step 3 – Landmark will Categorize whether Eyes are Open or Closed

We are using landmark for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with.
We loaded our model using

```
python driver_drowness.py --shape-predictor
shape_predictor_68_face_landmarks.dat --alarm alarm.wav
```

Step 4 – Calculate Score to Check whether Person is Drowsy
The score is basically a value we will use to determine how long the person has closed his eyes. 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.
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()**

SOFTWARE USED:

- Python
- Open CV
- Command prompt

HARDWARE USED:

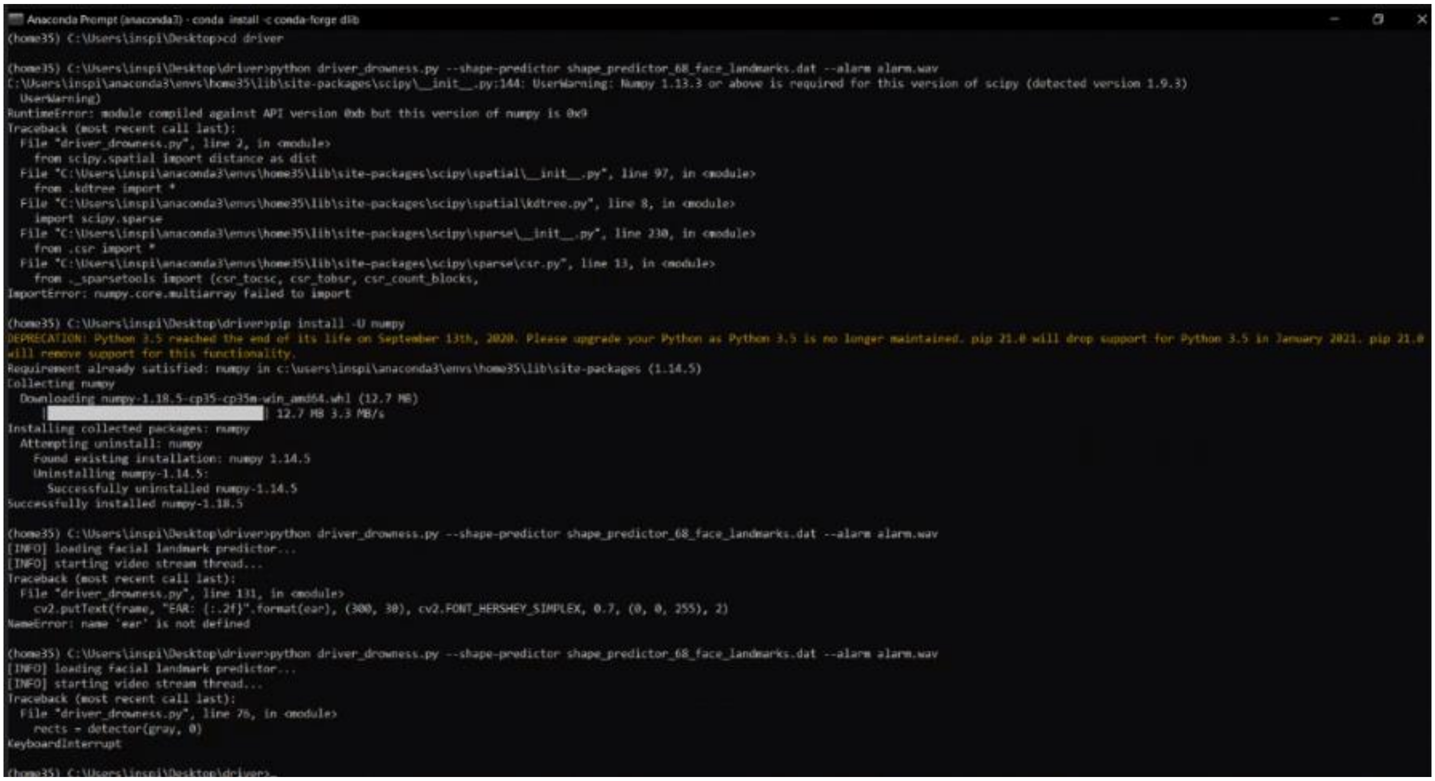
- Windows 7/8/10

SOURCE CODE:

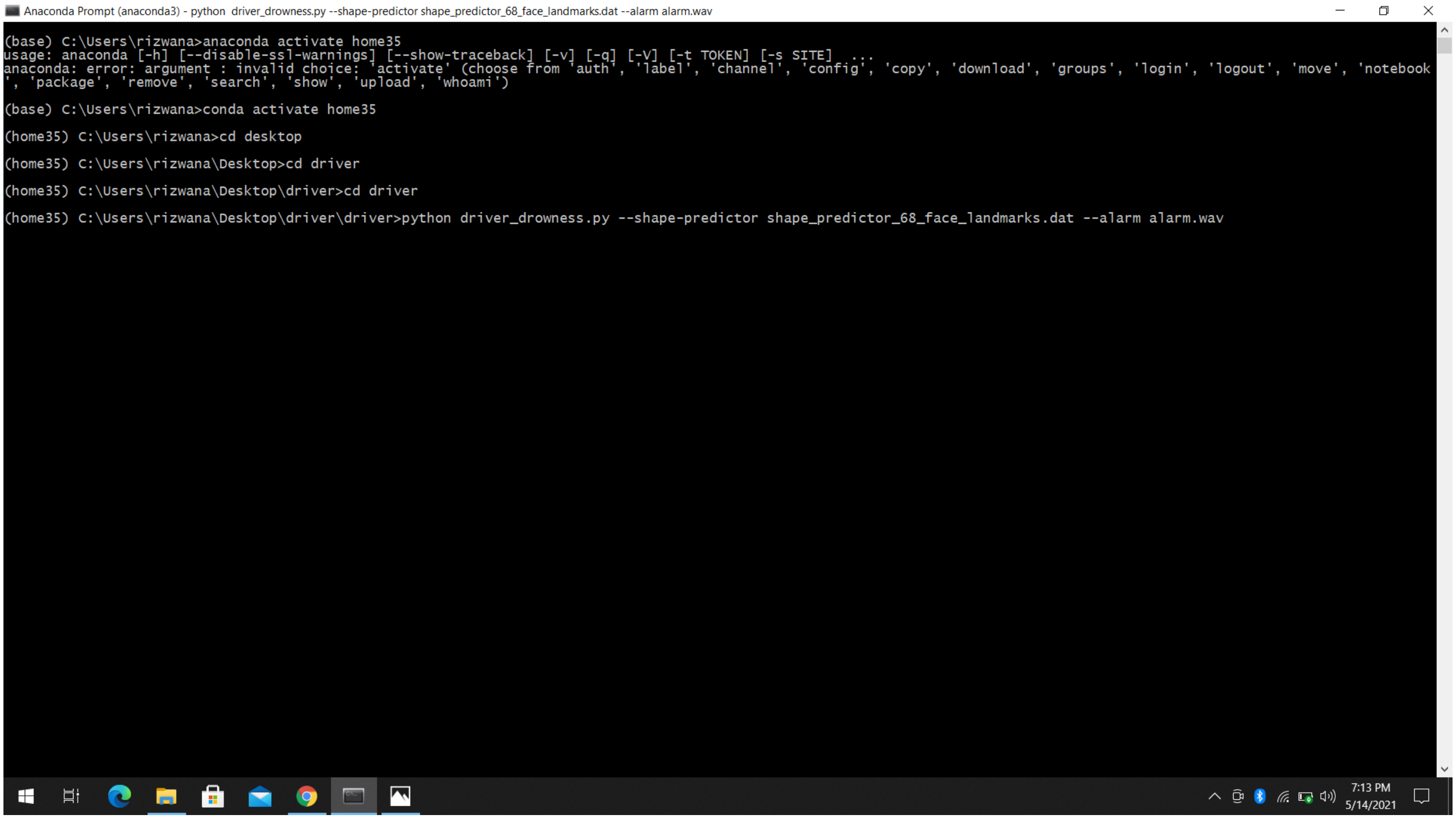
```
# import the necessary packages
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import playsound
import argparse
import imutils
import time
import dlib
import cv2
def sound_alarm(path):
    # play an alarm sound
    playsound.playsound(path)
    def eye_aspect_ratio(eye):
    # compute the euclidean distances between the two sets of
    # vertical eye landmarks (x, y)-coordinates
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    # compute the euclidean distance between the horizontal
    # eye landmark (x, y)-coordinates
    C = dist.euclidean(eye[0], eye[3])
    # compute the eye aspect ratio
    ear = (A + B) / (2.0 * C)
    # return the eye aspect ratio
     return ear
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--shape-predictor", required=True,
    help="path to facial landmark predictor")
ap.add_argument("-a", "--alarm", type=str, default="",
    help="path alarm .WAV file")
ap.add_argument("-w", "--webcam", type=int, default=0,
    help="index of webcam on system")
args = vars(ap.parse_args())
# define two constants, one for the eye aspect ratio to indicate
# blink and then a second constant for the number of consecutive
# frames the eye must be below the threshold for to set off the
# alarm
EYE_AR_THRESH = 0.3
```

```
EYE_AR_CONSEC_FRAMES = 48
# initialize the frame counter as well as a boolean used to
# indicate if the alarm is going off
COUNTER = 0
ALARM_ON = False
# initialize dlib's face detector (HOG-based) and then create
# the facial landmark predictor
print("[INFO] loading facial landmark predictor...")
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])
```

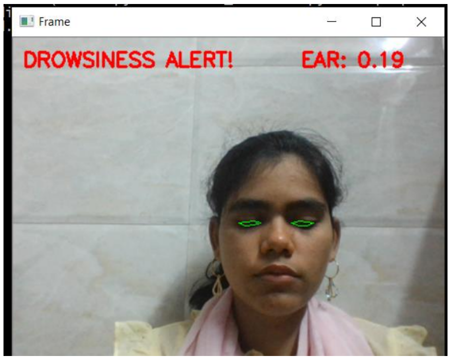
5. OUTPUT :


Anaconda Prompt (anaconda3) - conda install -c conda-forge dlib

```
(home35) C:\Users\inspi\Desktop>cd driver

(home35) C:\Users\inspi\Desktop\driver>python driver_drowness.py --shape-predictor shape_predictor_68_face_landmarks.dat --alarm alarm.wav
C:\Users\inspi\anaconda3\envs\home35\lib\site-packages\scipy\__init__.py:144: UserWarning: Numpy 1.13.3 or above is required for this version of scipy (detected version 1.9.3)
  UserWarning)
RuntimeError: module compiled against API version 0xb but this version of numpy is 0x9
Traceback (most recent call last):
  File "driver_drowness.py", line 2, in <module>
    from scipy.spatial import distance as dist
  File "C:\Users\inspi\anaconda3\envs\home35\lib\site-packages\scipy\spatial\__init__.py", line 97, in <module>
    from .kdtree import *
  File "C:\Users\inspi\anaconda3\envs\home35\lib\site-packages\scipy\spatial\kdtree.py", line 8, in <module>
    import scipy.sparse
  File "C:\Users\inspi\anaconda3\envs\home35\lib\site-packages\scipy\sparse\__init__.py", line 230, in <module>
    from .csr import *
  File "C:\Users\inspi\anaconda3\envs\home35\lib\site-packages\scipy\sparse\csr.py", line 13, in <module>
    from ._sparsetools import (csr_tocsc, csr_tobsr, csr_count_blocks,
ImportError: numpy.core.multiarray failed to import

(home35) C:\Users\inspi\Desktop\driver>pip install -U numpy
DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. Please upgrade your Python as Python 3.5 is no longer maintained. pip 21.0 will drop support for Python 3.5 in January 2021. pip 21.0
will remove support for this functionality.
Requirement already satisfied: numpy in c:\users\inspi\anaconda3\envs\home35\lib\site-packages (1.14.5)
Collecting numpy
  Downloading numpy-1.18.5-cp35-cp35m-win_amd64.whl (12.7 MB)
     |████████████████████████████████| 12.7 MB 3.3 MB/s
Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 1.14.5
    Uninstalling numpy-1.14.5:
      Successfully uninstalled numpy-1.14.5
Successfully installed numpy-1.18.5

(home35) C:\Users\inspi\Desktop\driver>python driver_drowness.py --shape-predictor shape_predictor_68_face_landmarks.dat --alarm alarm.wav
[INFO] loading facial landmark predictor...
[INFO] starting video stream thread...
Traceback (most recent call last):
  File "driver_drowness.py", line 131, in <module>
    cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
NameError: name 'ear' is not defined

(home35) C:\Users\inspi\Desktop\driver>python driver_drowness.py --shape-predictor shape_predictor_68_face_landmarks.dat --alarm alarm.wav
[INFO] loading facial landmark predictor...
[INFO] starting video stream thread...
Traceback (most recent call last):
  File "driver_drowness.py", line 76, in <module>
    rects = detector(gray, 0)
KeyboardInterrupt

(home35) C:\Users\inspi\Desktop\driver>_
```


Anaconda Prompt (anaconda3) - python driver_drowness.py --shape-predictor shape_predictor_68_face_landmarks.dat --alarm alarm.wav
7:13 PM
5/14/2021

```
(base) C:\Users\rizwana>anaconda activate home35
usage: anaconda [-h] [--disable-ssl-warnings] [--show-traceback] [-v] [-q] [-V] [-t TOKEN] [-s SITE]  ...
anaconda: error: argument : invalid choice: 'activate' (choose from 'auth', 'label', 'channel', 'config', 'copy', 'download', 'groups', 'login', 'logout', 'move', 'notebook
', 'package', 'remove', 'search', 'show', 'upload', 'whoami')

(base) C:\Users\rizwana>conda activate home35

(home35) C:\Users\rizwana>cd desktop

(home35) C:\Users\rizwana\Desktop>cd driver

(home35) C:\Users\rizwana\Desktop\driver>cd driver

(home35) C:\Users\rizwana\Desktop\driver\driver>python driver_drowness.py --shape-predictor shape_predictor_68_face_landmarks.dat --alarm alarm.wav
```



DROWSINESS ALERT! EAR: 0.19

Landmarks around eyes detects if the person is drowsy or not.

6. CONCLUSION:

Working on this project we get the understanding about drowsiness detection. We gained knowledge of python and open CV. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

7. REFERENCES :

- https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/
- Wreggit, S. S., Kim, C. L., and Wierwille, W. W., Fourth Semi-Annual Research Report”, Research on Vehicle-Based Driver Status Performance Monitoring”, Blacksburg, VA: Virginia Polytechnic Institute and State University, ISE Department, January 2013.
- H. Singh, J. S. Bhatia, and J. Kaur, “Eye tracking based driver fatigue monitoring and warning system”, in Proc. IEEE IICPE, New Delhi, India, Jan. 2014.