Department of Computer Applications

INVERTIS
UNIVERSITY BAREILLY
Established by Govt. of U.P. u/s 2(f) of UGC Act 1956 vide U.P. Act 22 of 2010.

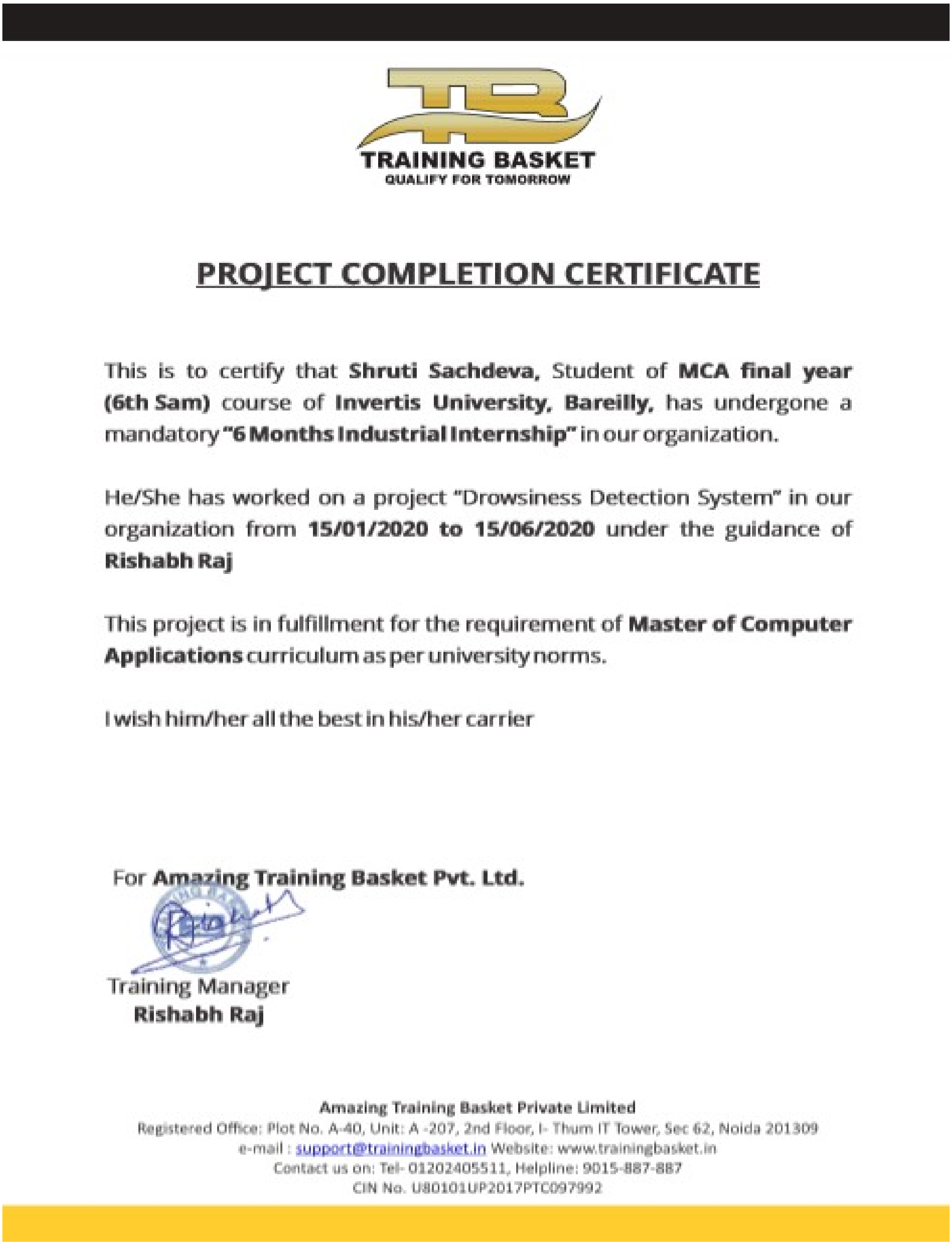
A Project Report
on

Drowsiness Detection System

Submitted in Fulfillment of the Requirement for the
Degree of
Master of Computer Applications
Invertis University, Bareilly

Project Guide
Ms. Shivangi Ghildiyal
Assistant Professor
Department of Computer Applications

Submitted By
Shruti Sachdeva
1850204017



TRAINING BASKET
QUALIFY FOR TOMORROW

PROJECT COMPLETION CERTIFICATE

This is to certify that **Shruti Sachdeva,** Student of **MCA final year (6th Sam)** course of **Invertis University, Bareilly,** has undergone a mandatory **"6 Months Industrial Internship"** in our organization.

He/She has worked on a project "Drowsiness Detection System" in our organization from **15/01/2020 to 15/06/2020** under the guidance of **Rishabh Raj**

This project is in fulfillment for the requirement of **Master of Computer Applications** curriculum as per university norms.

I wish him/her all the best in his/her carrier

For **Amazing Training Basket Pvt. Ltd.**

Training Manager
Rishabh Raj

Amazing Training Basket Private Limited
Registered Office: Plot No. A-40, Unit: A -207, 2nd Floor, I- Thum IT Tower, Sec 62, Noida 201309
e-mail : support@trainingbasket.in Website: www.trainingbasket.in
Contact us on: Tel- 01202405511, Helpline: 9015-887-887
CIN No. U80101UP2017PTC097992

ACKNOWLEGDEMENT

I extend my thanks of **Department of Computer Applications, Invertis University, Bareilly** who has given me opportunity to work on this Project.

Today I am feeling a great sense of Excitement on my way to successfully complete my project on **“Drowsiness Detection System”** under the guidance of **“Rishabh Raj”**.

I sincerely thank him for responding great confidence and faith in my work and being with me to encourage and guide me to successful project completion.

I should also like to thank **Mr. Jitendra Chaudhary, HOD, Department of Computer Applications** for their support and all our friends and colleagues who have created an atmosphere to encourage me from time to time making our work easy.

Thank You…..

Shruti Sachdeva

1850204017

INDEX

LIST OF FIGURES

1.) Introduction of Project

Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving. The objective of this project is to build a drowsiness detection system that will detect that a person's eyes are closed for a few seconds. Driver drowsiness detection is a car safety technology which prevents accidents when the driver is getting drowsy. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads. Driver fatigue is a significant factor in a large number of vehicle accidents. Recent statistics estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes.

The development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident avoidance systems. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects. Driver inattention might be the result of a lack of alertness when driving due to driver drowsiness and distraction.

Driver distraction occurs when an object or event draws a person's attention away from the driving task. Unlike driver distraction, driver drowsiness involves no triggering event but, instead, is characterized by a progressive withdrawal of attention from the road and traffic demands. Both driver drowsiness and distraction, however, might have the same effects, i.e., decreased driving performance, longer reaction time, and an increased risk of crash involvement. Based on Acquisition of video from the camera that is in front of driver perform real-time processing of an incoming video stream in order to infer the driver's level of fatigue if the drowsiness is Estimated then it will give the alert by sensing the eyes.

2.) Problem Definition

Today drowsy driving is a serious problem that leads to thousands of accidents each year. Motor vehicle collisions lead to significant death and disability as well as significant financial cost to both security and individual due to the driver impairments. Drowsiness is one of the factors for collisions. In India, no monitoring device is used to measure the drowsiness of driver. Some kind of systems like driver fatigue monitor, real time vision based on driver state monitoring system, seeing driver assisting system, user center drowsiness driver detection and working system are implemented in foreign countries. All the systems focus either changes in eye movement, physiological measures or driver performance measure. Due to illumination variation, the traditional systems have some defects.

Problem : A new system to monitor the driver fatigue.

3.) System Analysis & Feasibility Study

As the name implies, a feasibility analysis is used to determine the viability of an idea , such as ensuring a project is legally and technically feasible as well as economically justifiable. In simple terms, a feasibility study involves taking a judgment call on whether a project is doable. The two criteria to judge feasibility are cost required and value to be delivered. A well-designed study should offer a historical background of the business or project, a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, such studies precede technical development and project implementation.

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions.

Types of Feasibility Study :

Technical Feasibility

Technical Feasibility assessment is centered on the technical resources available to the organization. It helps organizations access if the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves evaluation of the hardware and the software requirements of the proposed system.

Economic Feasibility

Economic Feasibility helps organizations assess the project's viability, cost, and benefits associated with the projects before financial resources are allocated. It also serves as an independent project assessment, and enhances project credibility, as a result. It helps decision-makers determine the positive economic benefits to the organization that the proposed system will provide, and helps quantify them. This assessment typically involves a cost/ benefits analysis of the project.

Operational Feasibility

Operational Feasibility involves undertaking a study to analyze and determine whether your business needs can be fulfilled by using the proposed solution. It also measures how well the proposed system solves problems and takes advantage of the opportunities identified during scope definition. Operational feasibility studies also analyze how the project plan satisfies the requirements identified in the requirements analysis phase of system development. To ensure success, desired operational outcomes must inform and guide design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, disposability, sustainability, affordability, and others.

Scheduling Feasibility

Scheduling Feasibility is the most important for project success. A project will fail if not completed on time. In scheduling feasibility, we estimate how much time the system will take to complete.

Legal Feasibility

Legal Feasibility investigates if the proposed system conflicts with legal requirements like data protection acts or social media laws.

4.) Scope of the Proposed System

The main idea behind this project is to develop a nonintrusive system which can detect fatigue of any human and can issue a timely warning. Drivers who do not take regular breaks when driving long distances run a high risk of becoming drowsy a state which they often fail to recognize early enough. According to the expert’s studies show that around one quarter of all serious motorway accidents are attributable to sleepy drivers in need of a rest, meaning that drowsiness causes more road accidents than drink-driving. This system will monitor the driver eyes using a camera and by developing an algorithm we can detect symptoms of driver fatigue early enough to avoid the person from sleeping. So, this project will be helpful in detecting driver fatigue in advance and will give warning output in form of alarm and popups.

5.) System Requirements

A computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as system requirements.

Operating System : Windows, Linux, Mac

5.1.) Hardware Requirements

The section of hardware requirements is an important task related to the software development insufficient random access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

- Intel Pentium i3 processor or equivalent or higher
- 4gb RAM or Higher
- 500gb HDD or Higher
- Network Connectivity

5.2.) Software Requirements

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should befeasible for the system. This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allows the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

- Python 3.7 or above versions
- Anaconda software

5.2.1.) Front End Tool

- **GUI Tkinter :** Tkinter commonly comes bundled with Python, using Tk and is Python's standard GUI framework. It is famous for its simplicity and graphical user interface. It is open-source and available under the Python License.

5.2.2.) Back End Tool

- **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.

- **OpenCV :** OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

- **TensorFlow :** An end-to-end open source python platform for everyone. TensorFlow is a python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

- **Keras :** Keras is a minimalist python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development.

- **Pygame :** Pygame is a python wrapper module for the SDL(Simple DirectMedia Layer) multimedia library. It contains python functions and classes that will allow you to use SDL's support for playing cdroms, audio and video output, and keyboard, mouse and joystick input.

6.) System Logical Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

6.1.) Data Flow Diagram

Data flow diagram graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between user and system designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams.

Data flow diagrams (DFDs) reveal relationships among and between the various components in a program or system. DFDs are an important technique for modeling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs consist of four major components: entities, processes, data stores, and data flows. The symbols used to depict how these components interact in a system are simple and easy to understand; however, there are several DFD models to work from, each having its own semiology.

- DFD (level-0)
- DFD (level-1)
- DFD (level-2)

DFD (level-0)

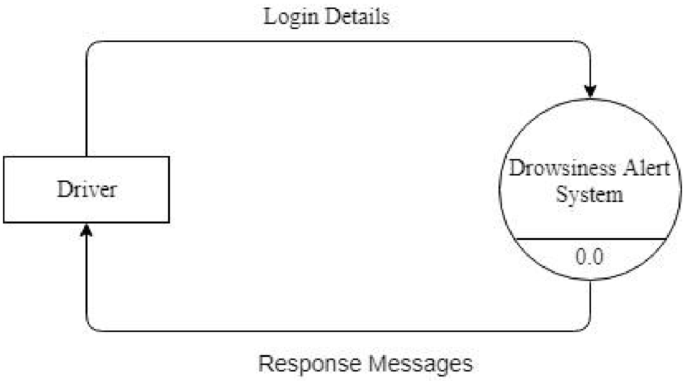


Fig.1. DFD (level-0)

DFD (level-1)

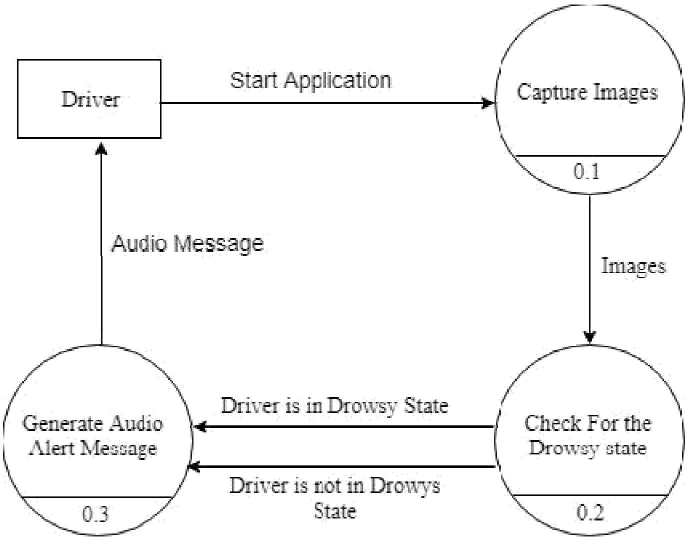

Driver
Start Application
Capture Images
0.1
Audio Message
Images
Generate Audio Alert Message
0.3
Driver is in Drowsy State
Driver is not in Drowys State
Check For the Drowsy state
0.2

Fig.2. DFD (level-1)

DFD (level-2)

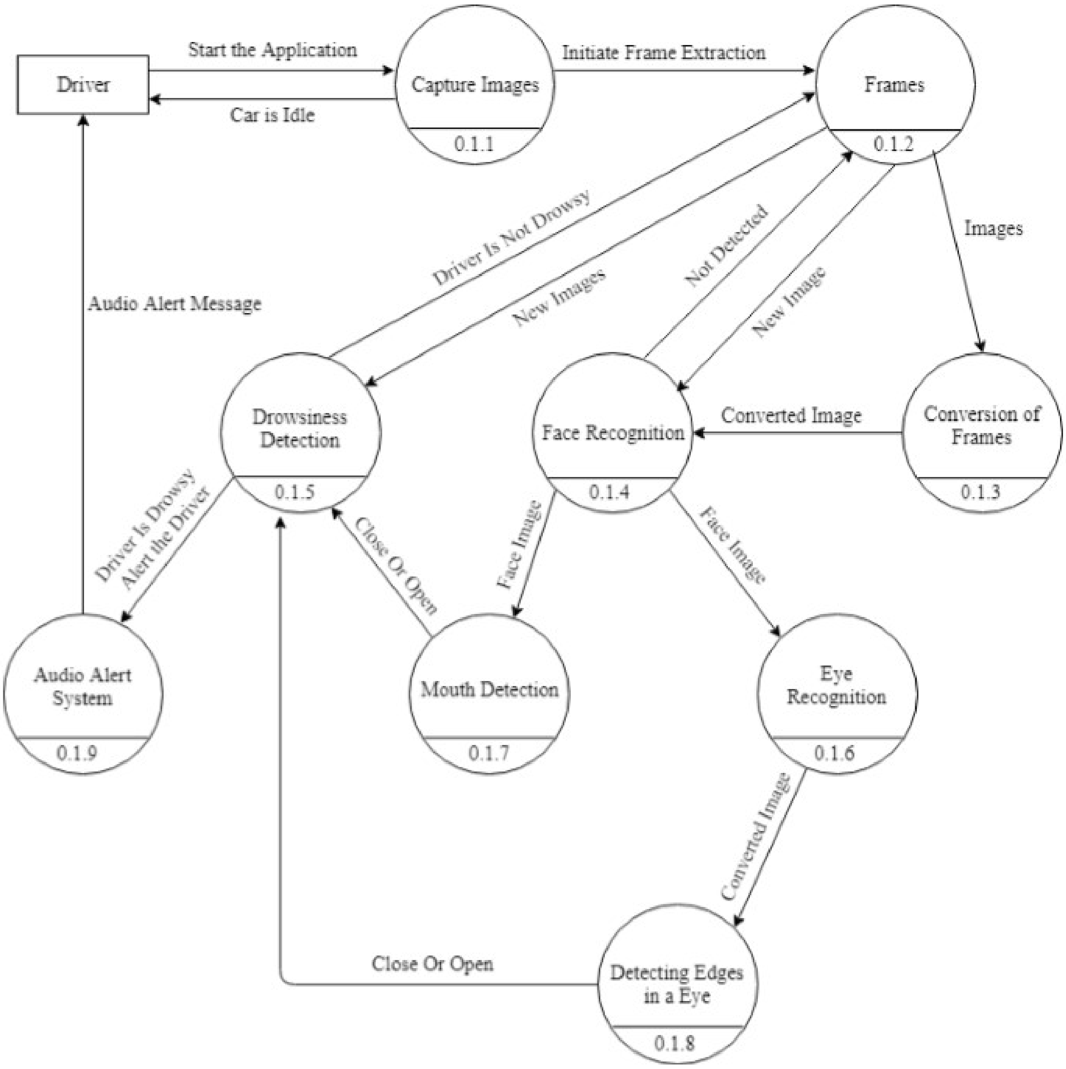


Fig.3. DFD (level-2)

6.2.) ER Diagram

Entity Relationship Diagrams are a major data modelling tool and will help organize the data in your project into entities and define the relationships between the entities. This process has proved to enable the analyst to produce a good database structure so that the data can be stored and retrieved in a most efficient manner. By using a graphical format it may help communication about the design between the designer and the user and the designer and the people who will implement it.

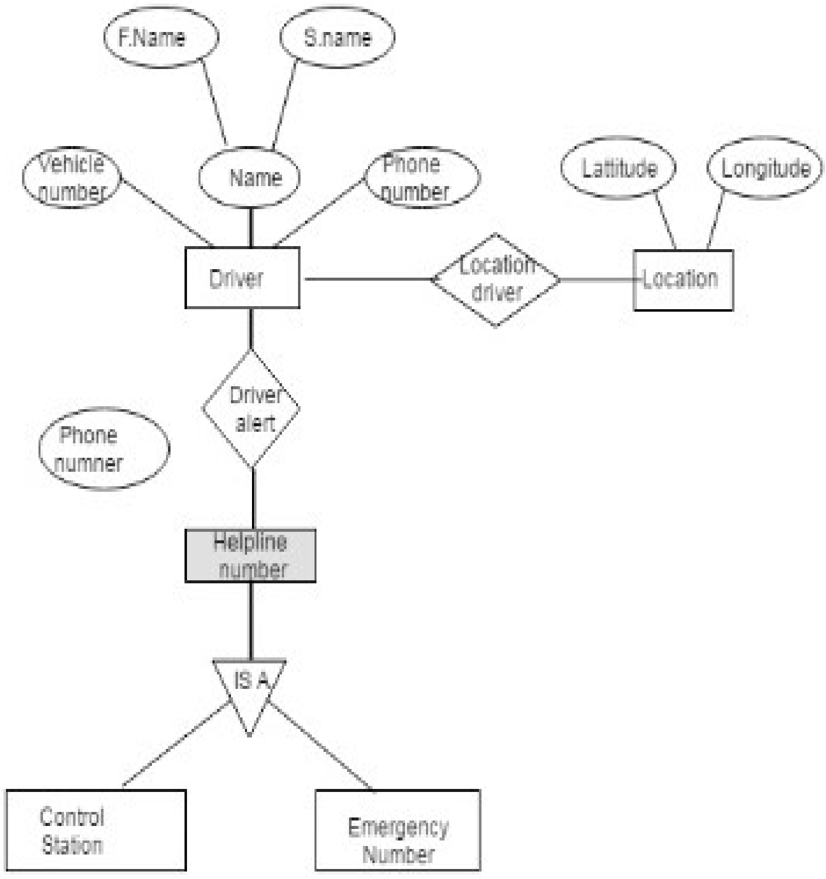


Fig.4. ER Diagram

6.3.) Use Case Diagram

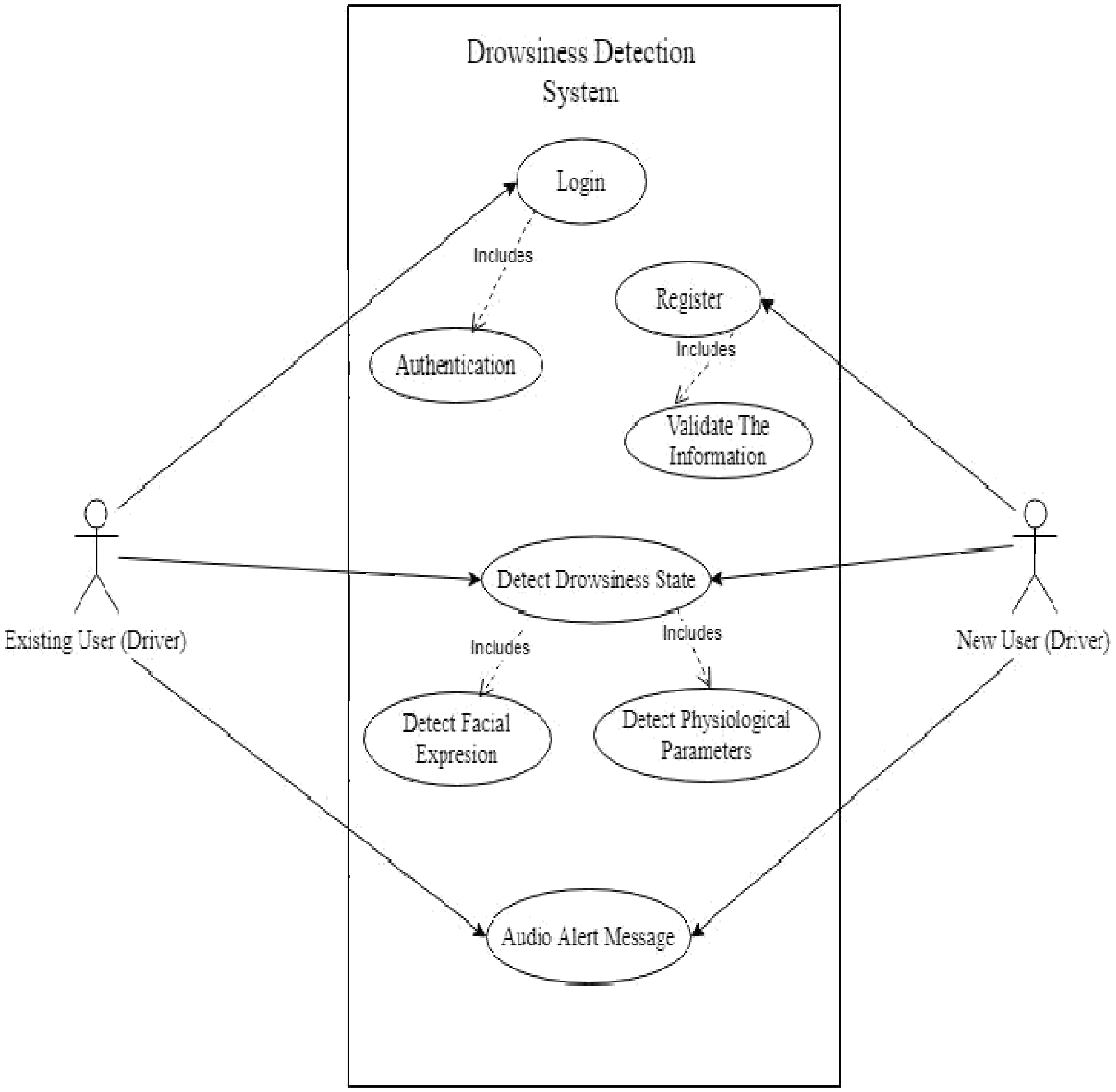


Fig.5. Use Case Diagram

6.4.) Sequence Diagram

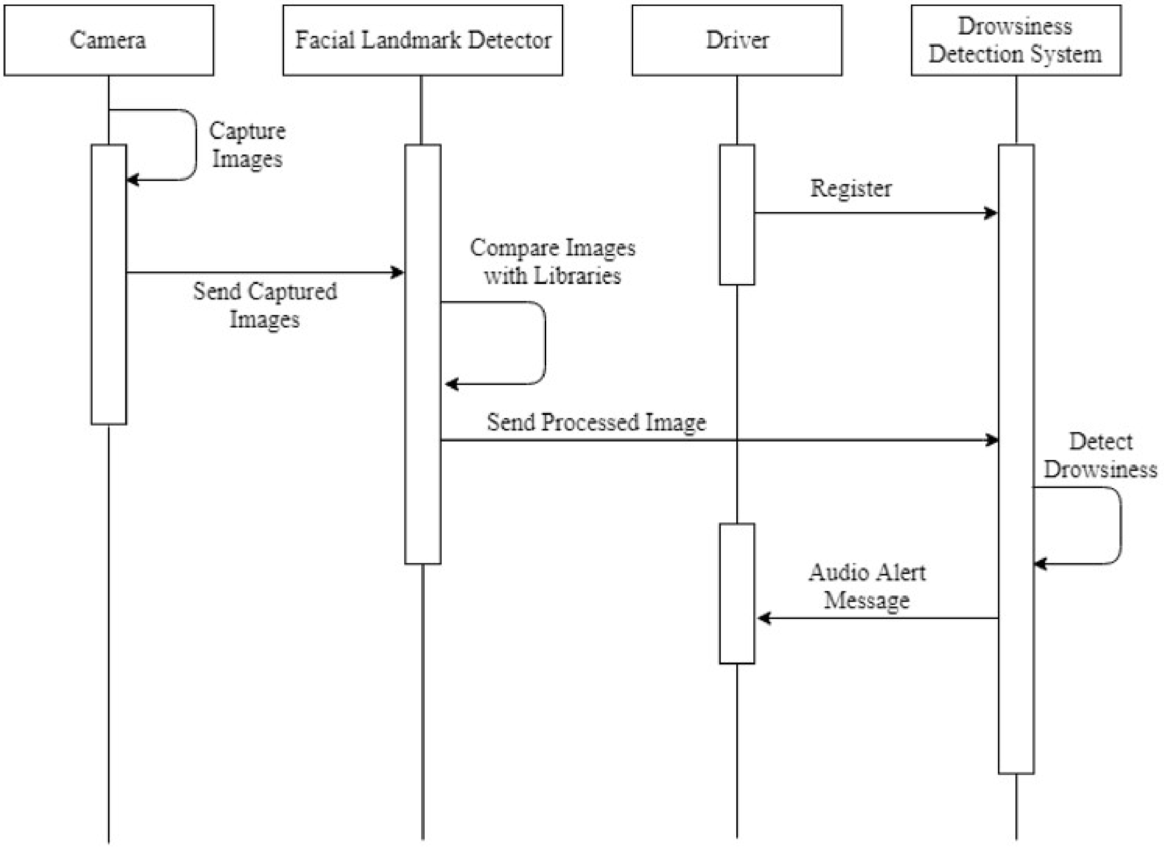


Fig.6. Sequence Diagram

6.5.) Flow Chart of System

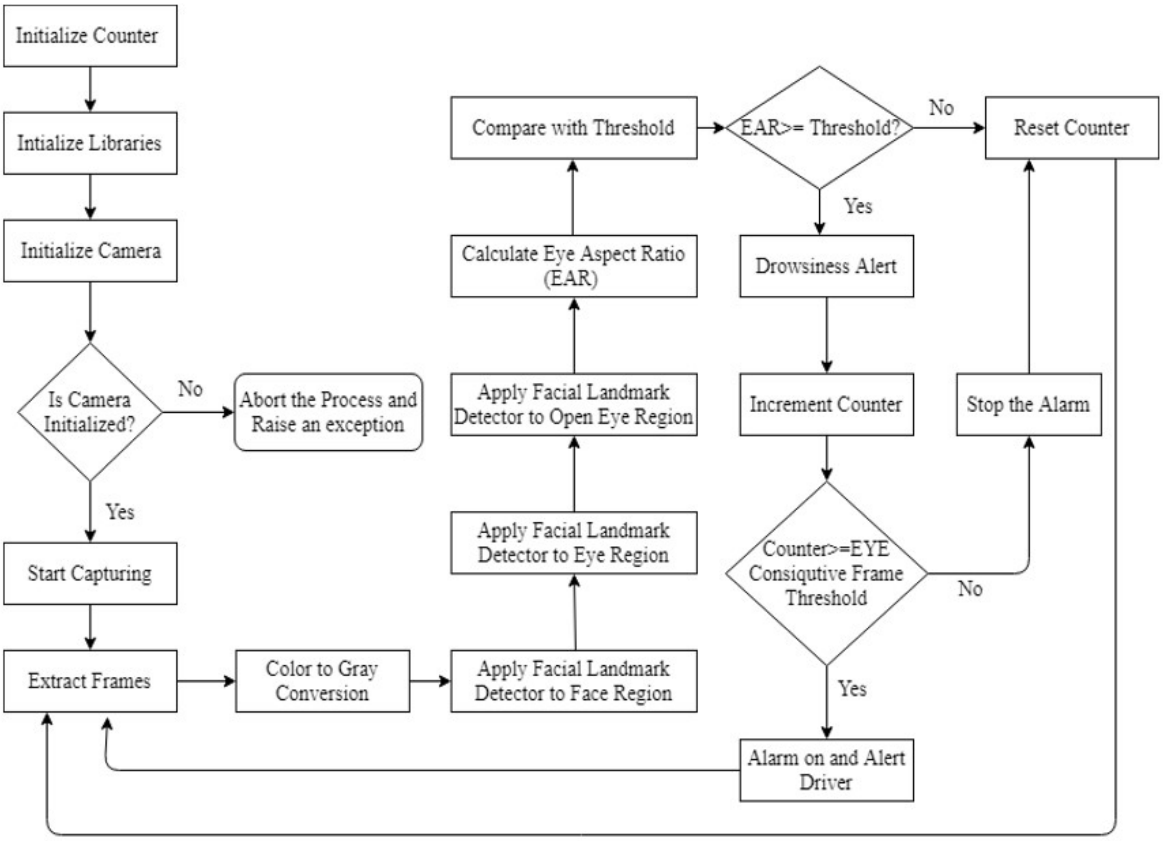
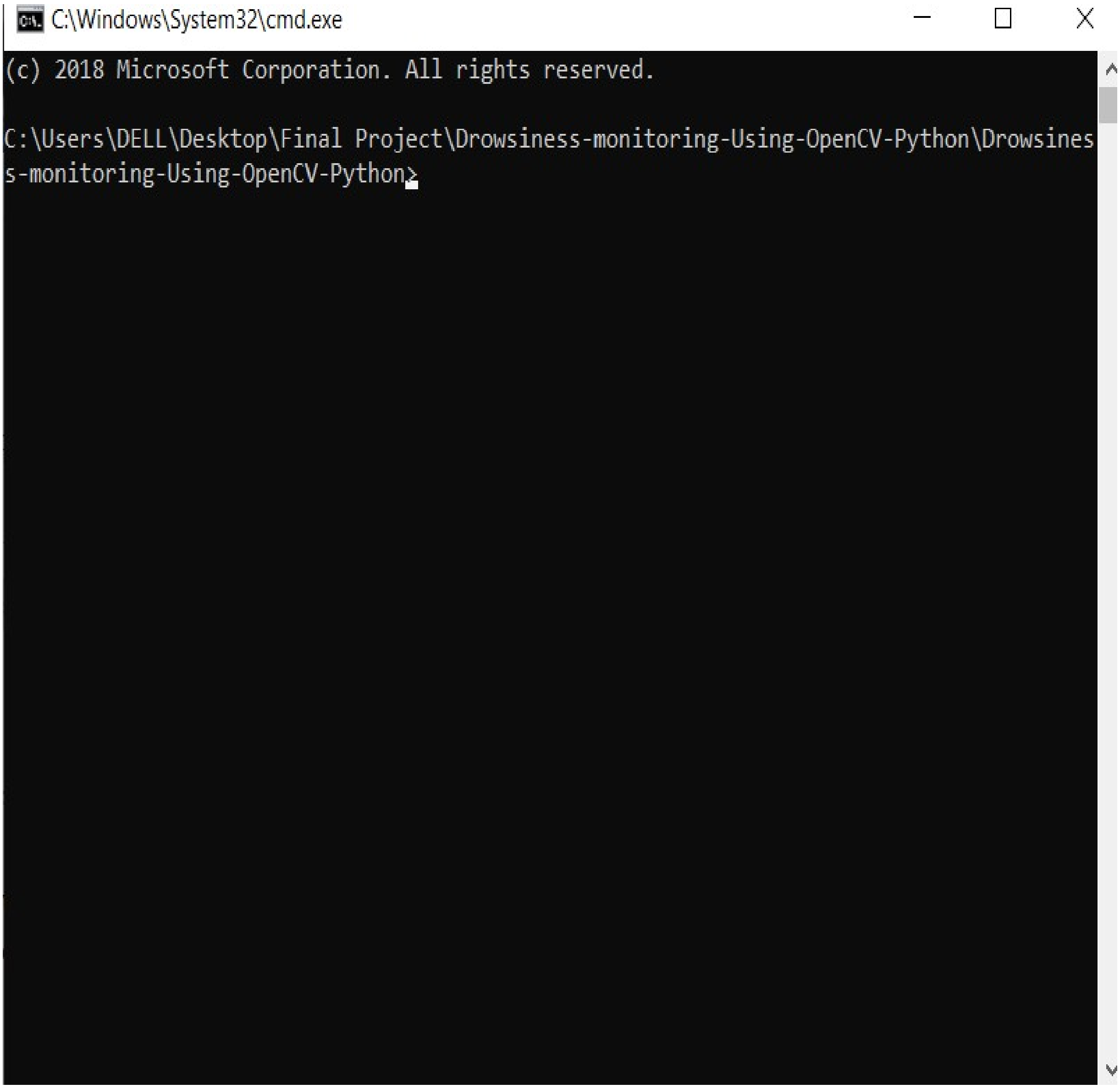

Initialize Counter
Intialize Libraries
Initialize Camera
Is Camera Initialized?
No
Abort the Process and Raise an exception
Yes
Start Capturing
Extract Frames
Color to Gray Conversion
Apply Facial Landmark Detector to Face Region
Apply Facial Landmark Detector to Eye Region
Apply Facial Landmark Detector to Open Eye Region
Calculate Eye Aspect Ratio (EAR)
Compare with Threshold
EAR>= Threshold?
No
Reset Counter
Yes
Drowsiness Alert
Increment Counter
Counter>=EYE Consiqutive Frame Threshold
No
Stop the Alarm
Yes
Alarm on and Alert Driver

Fig.7. Flow Chart of System

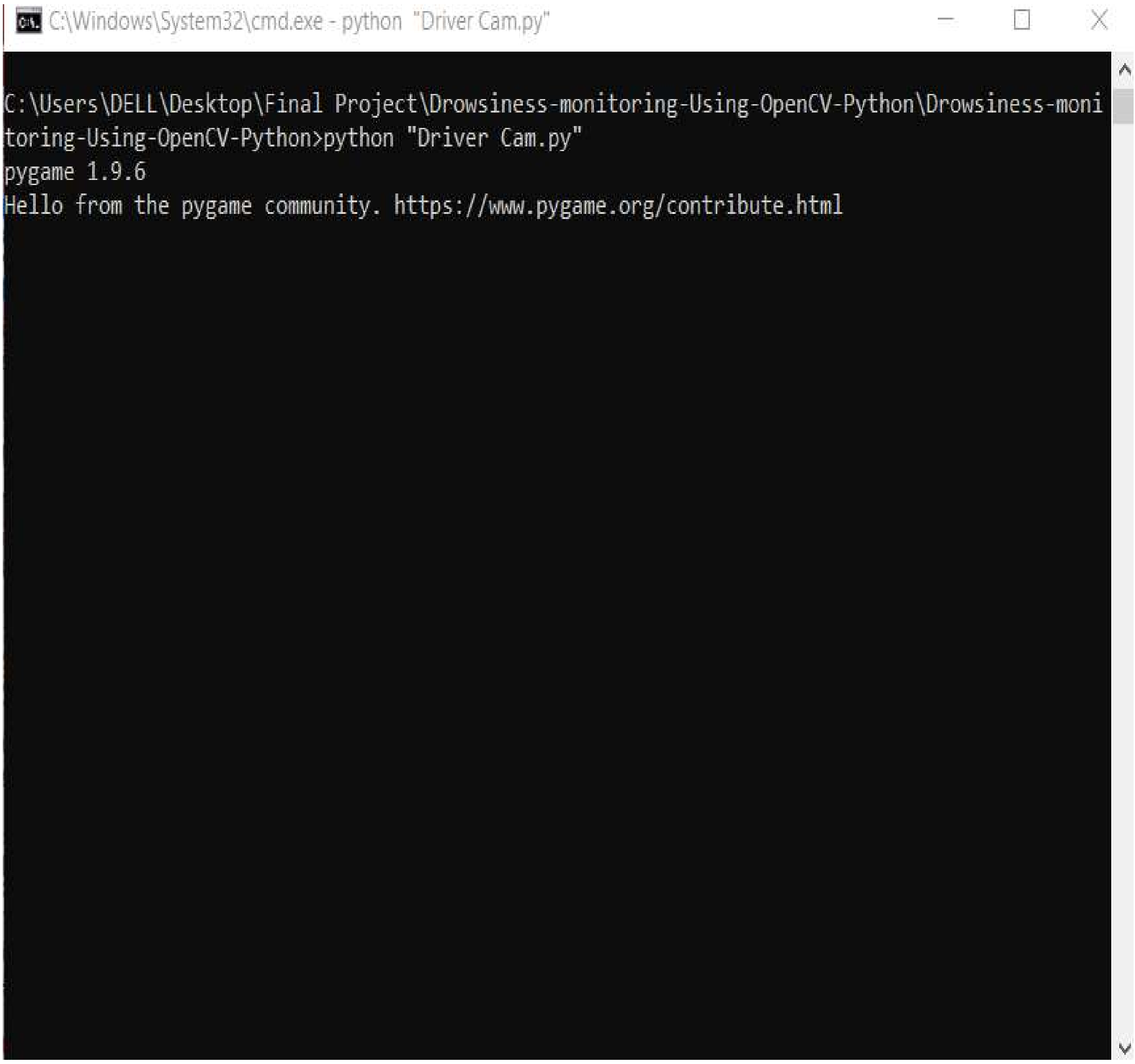
7.) Input Screen

7.1.)



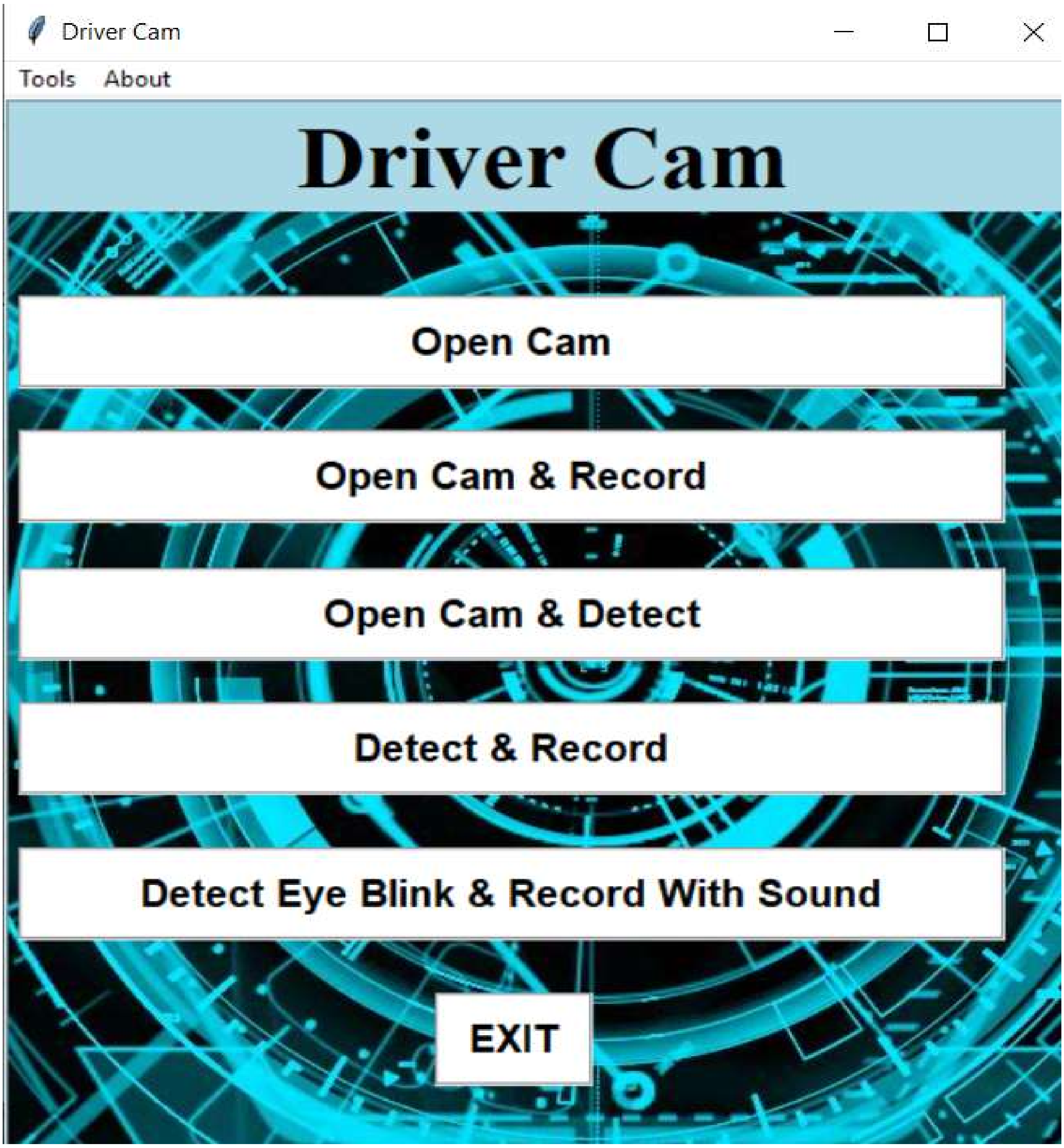
```
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\DELL\Desktop\Final Project\Drowsiness-monitoring-Using-OpenCV-Python\Drowsines
s-monitoring-Using-OpenCV-Python>
```

7.2.)

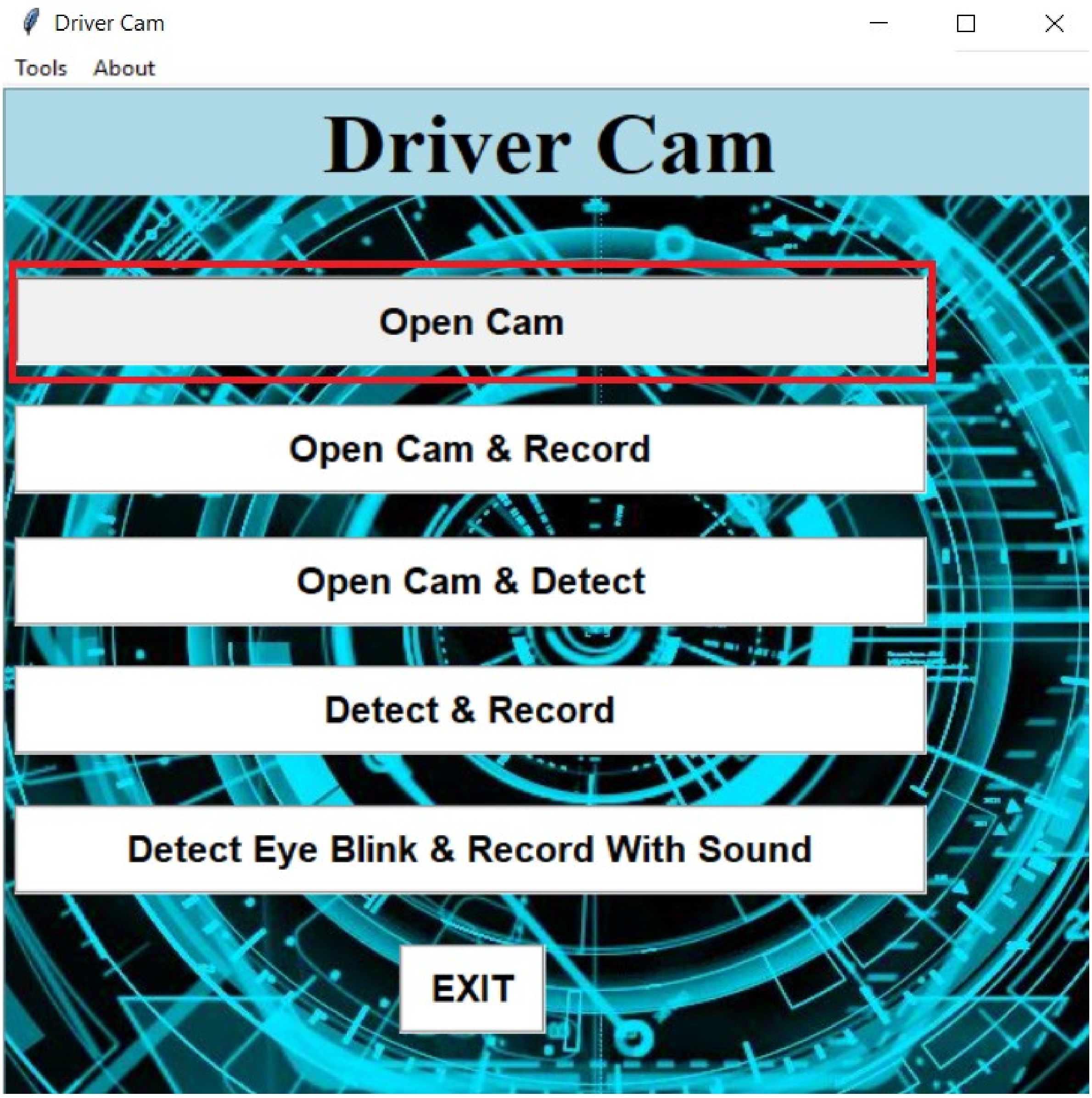


```
C:\Users\DELL\Desktop\Final Project\Drowsiness-monitoring-Using-OpenCV-Python\Drowsiness-moni
toring-Using-OpenCV-Python>python "Driver Cam.py"
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
```

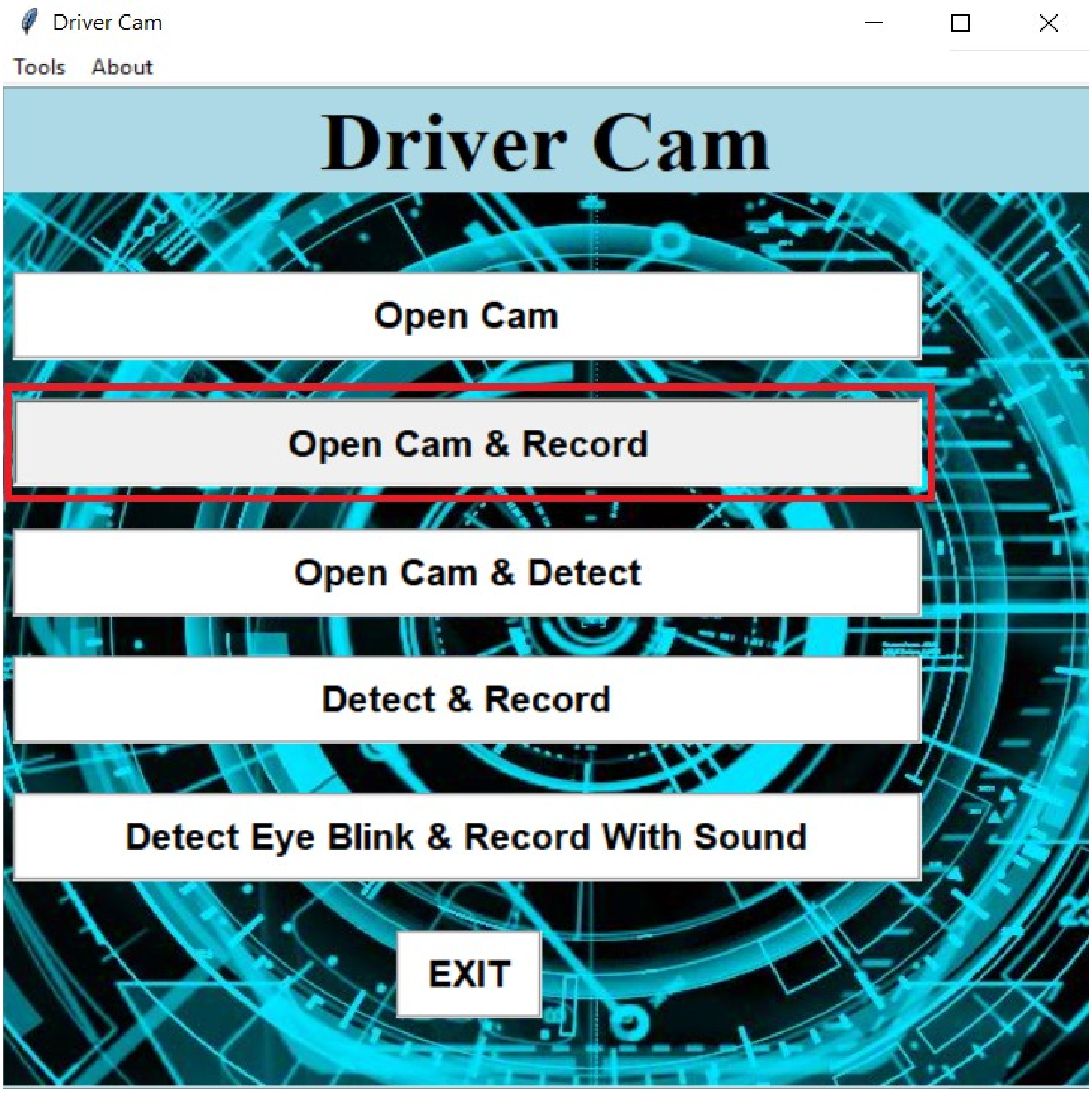
7.3.)


Driver Cam
Tools
About
Driver Cam
Open Cam
Open Cam & Record
Open Cam & Detect
Detect & Record
Detect Eye Blink & Record With Sound
EXIT

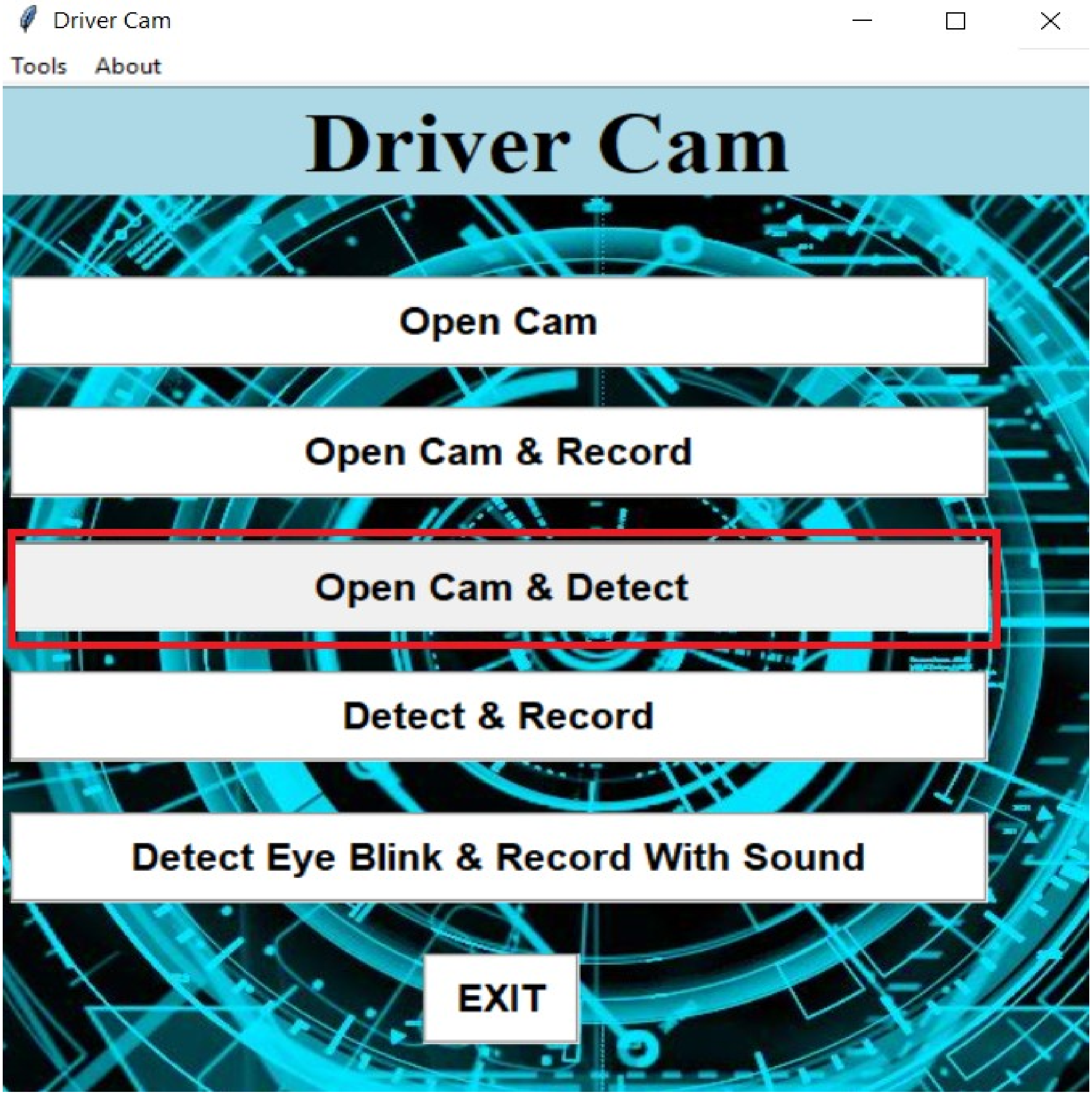
7.4.)


Driver Cam
Tools
About
Driver Cam
Open Cam
Open Cam & Record
Open Cam & Detect
Detect & Record
Detect Eye Blink & Record With Sound
EXIT

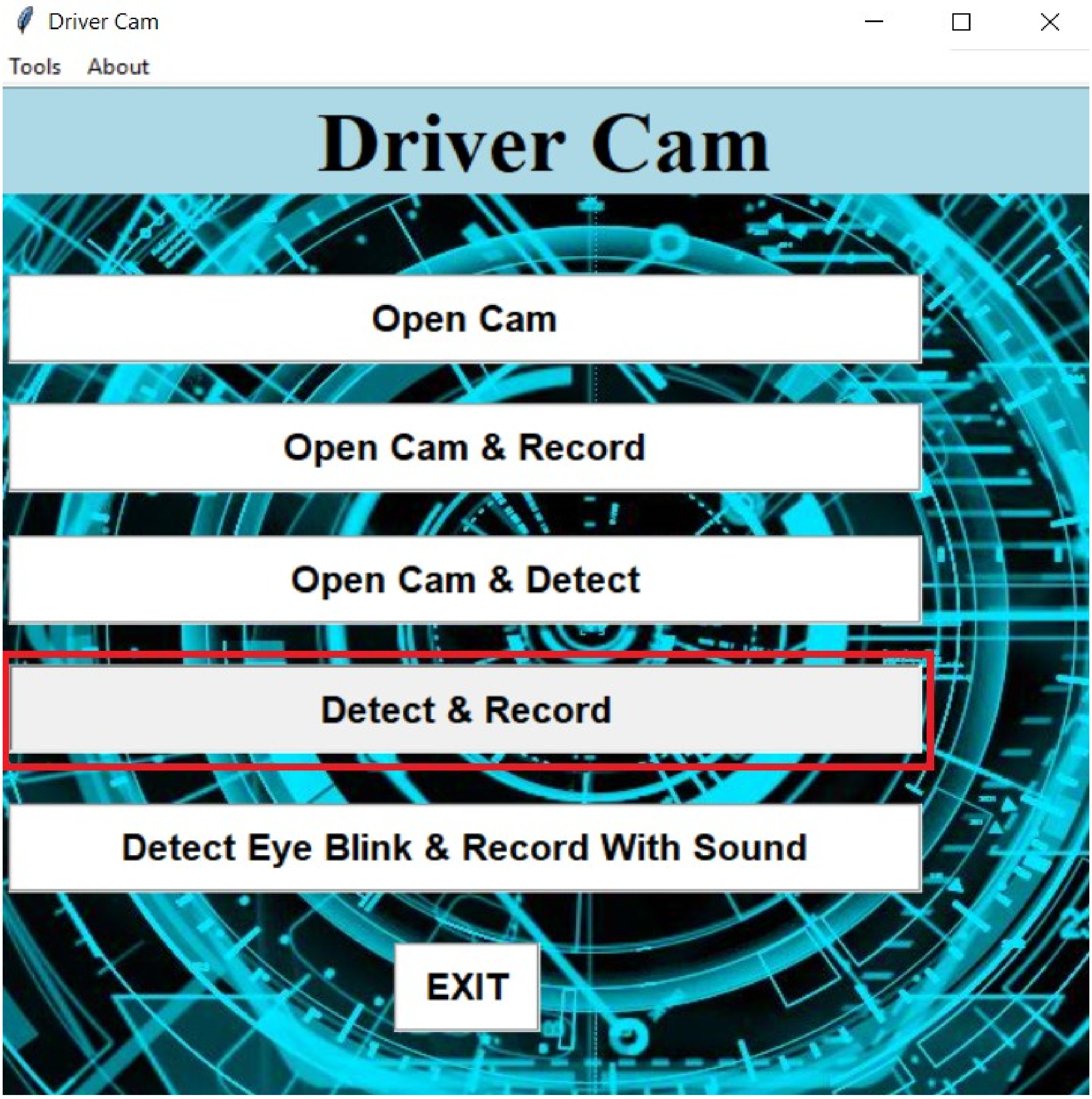
7.5.)


Driver Cam
Tools
About
Driver Cam
Open Cam
Open Cam & Record
Open Cam & Detect
Detect & Record
Detect Eye Blink & Record With Sound
EXIT

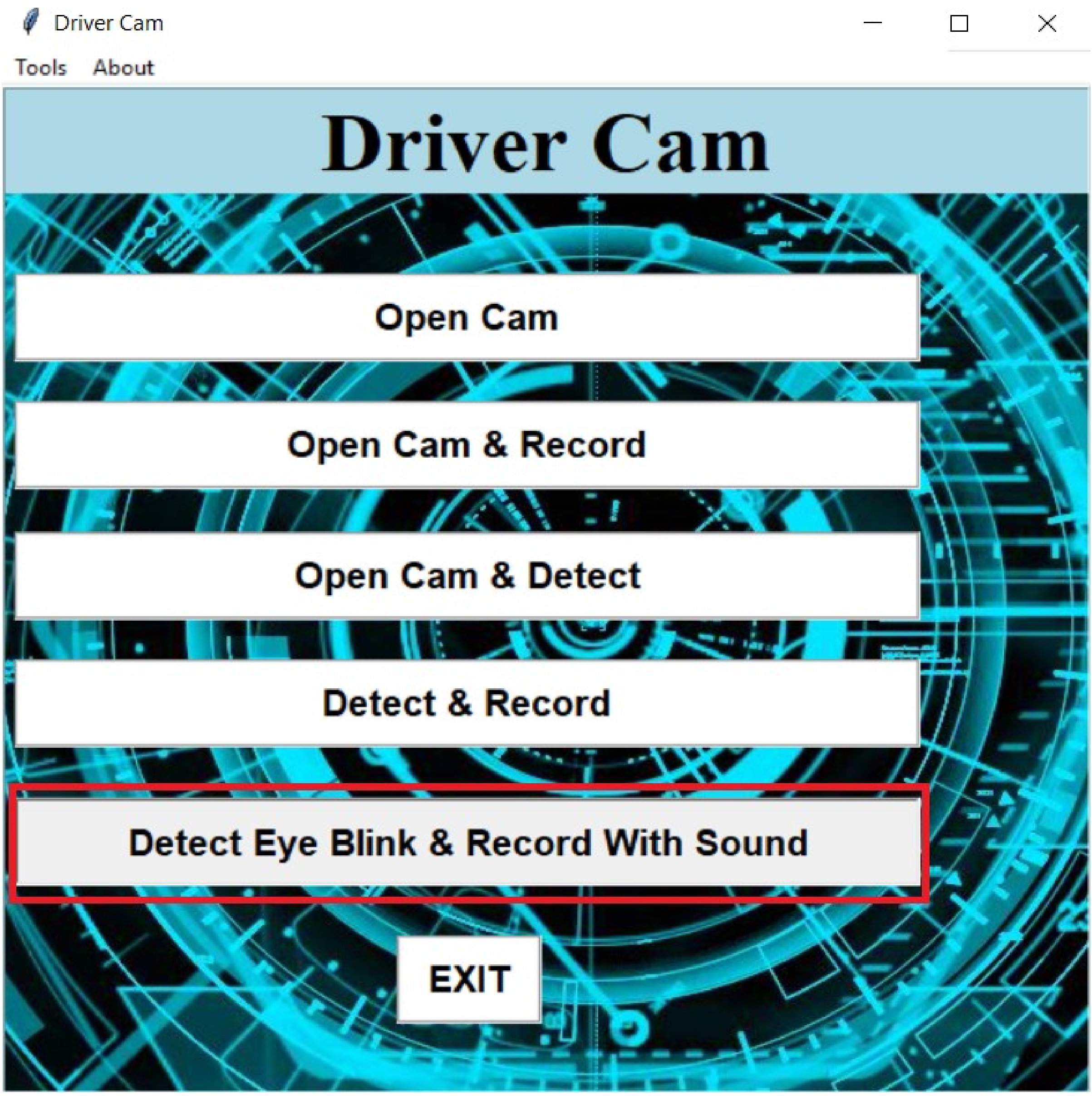
7.6.)


Driver Cam
Tools
About
Driver Cam
Open Cam
Open Cam & Record
Open Cam & Detect
Detect & Record
Detect Eye Blink & Record With Sound
EXIT

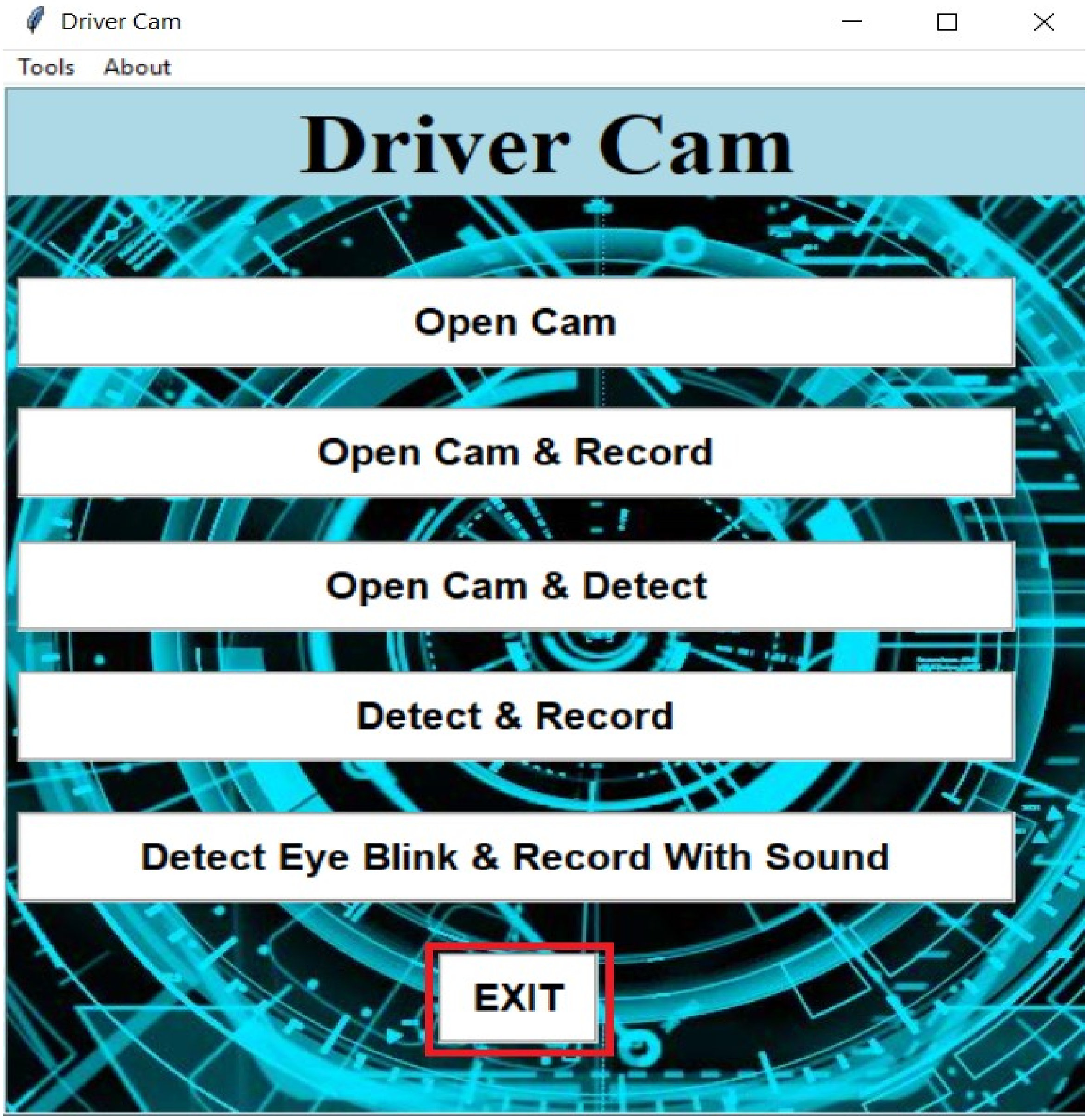
7.7.)


Driver Cam
Tools
About
Driver Cam
Open Cam
Open Cam & Record
Open Cam & Detect
Detect & Record
Detect Eye Blink & Record With Sound
EXIT

7.8.)

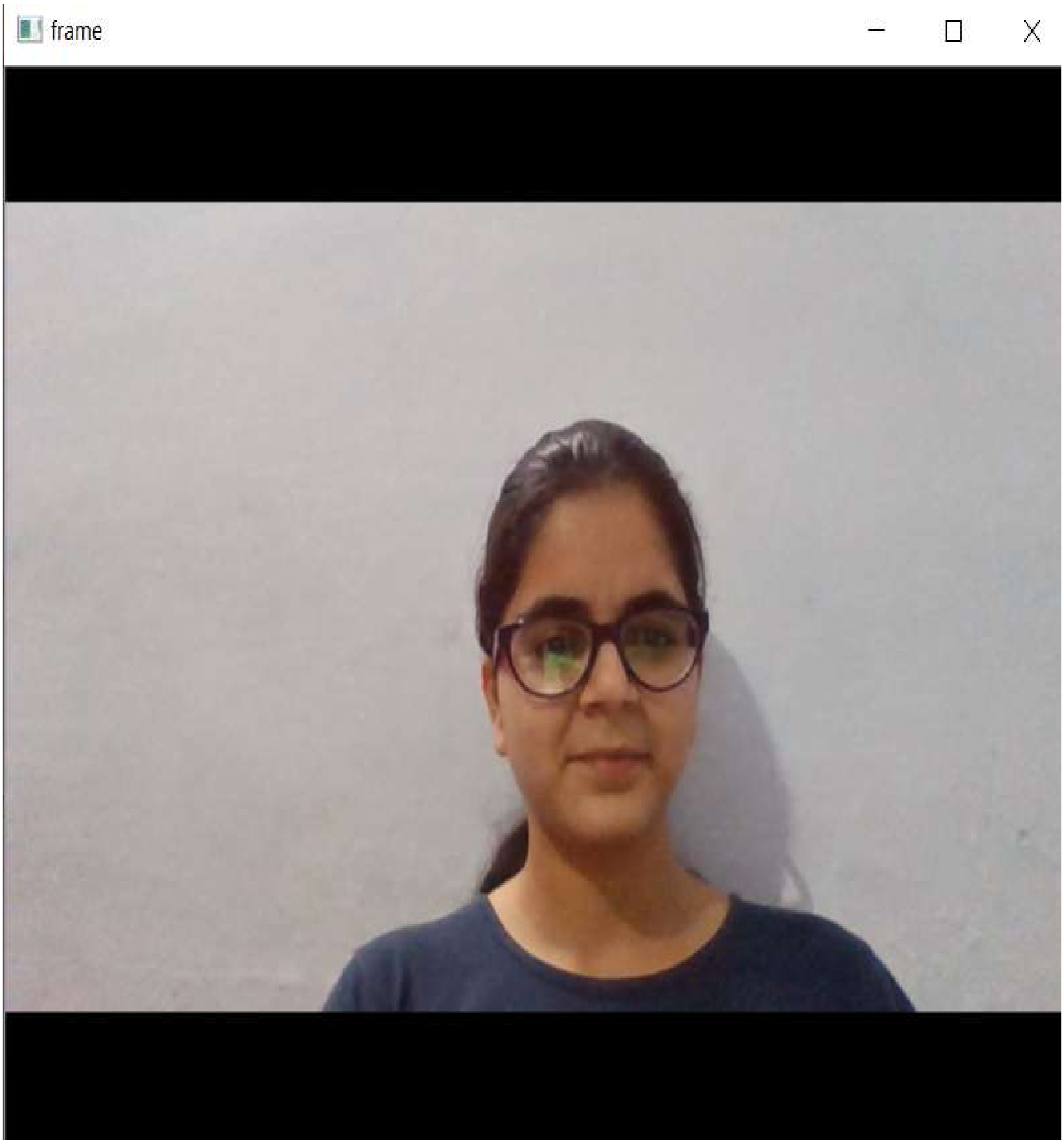

Driver Cam
Tools
About
Driver Cam
Open Cam
Open Cam & Record
Open Cam & Detect
Detect & Record
Detect Eye Blink & Record With Sound
EXIT

7.9.)

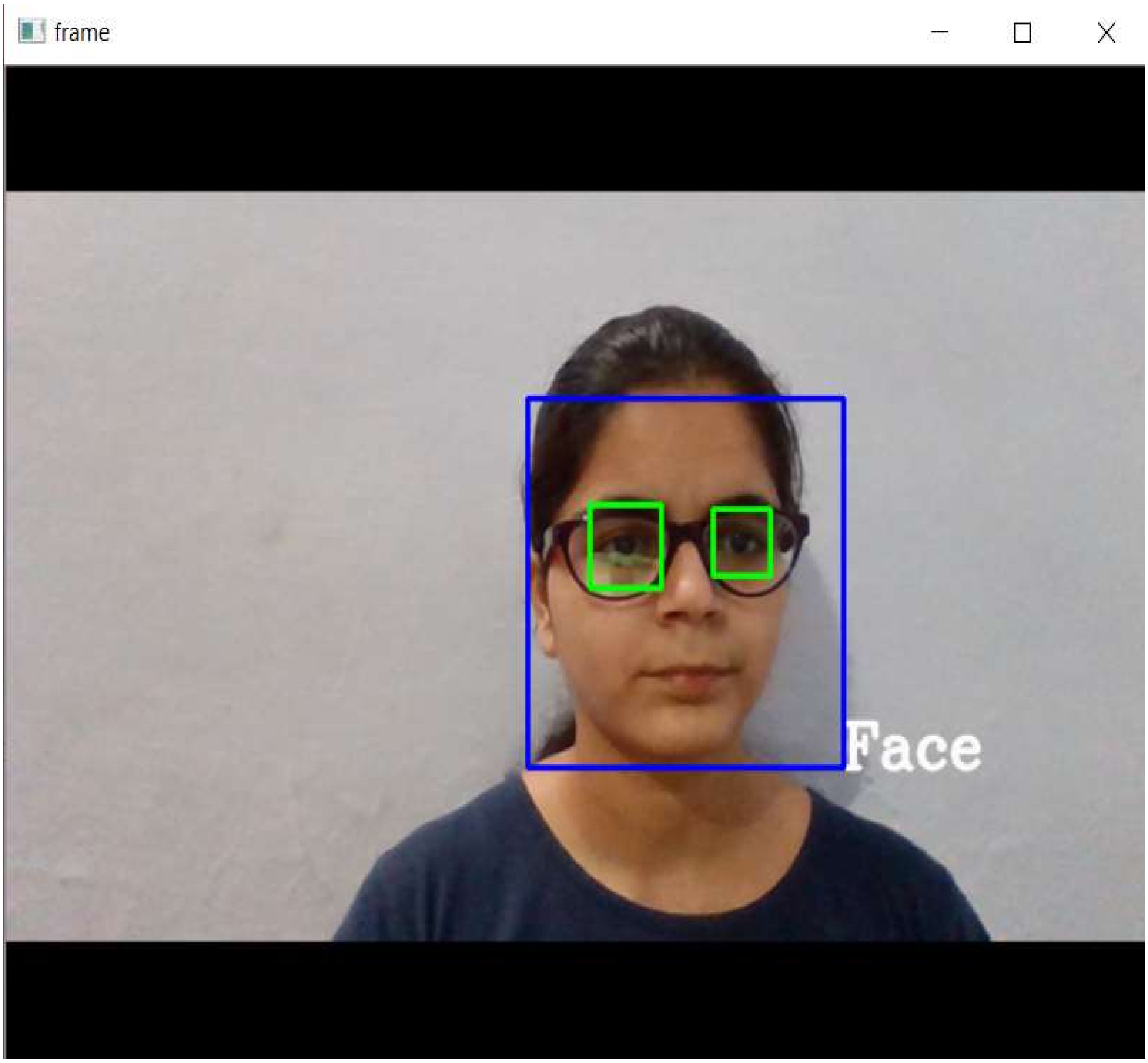

Driver Cam
Tools
About
Driver Cam
Open Cam
Open Cam & Record
Open Cam & Detect
Detect & Record
Detect Eye Blink & Record With Sound
EXIT

8.) Output Screen

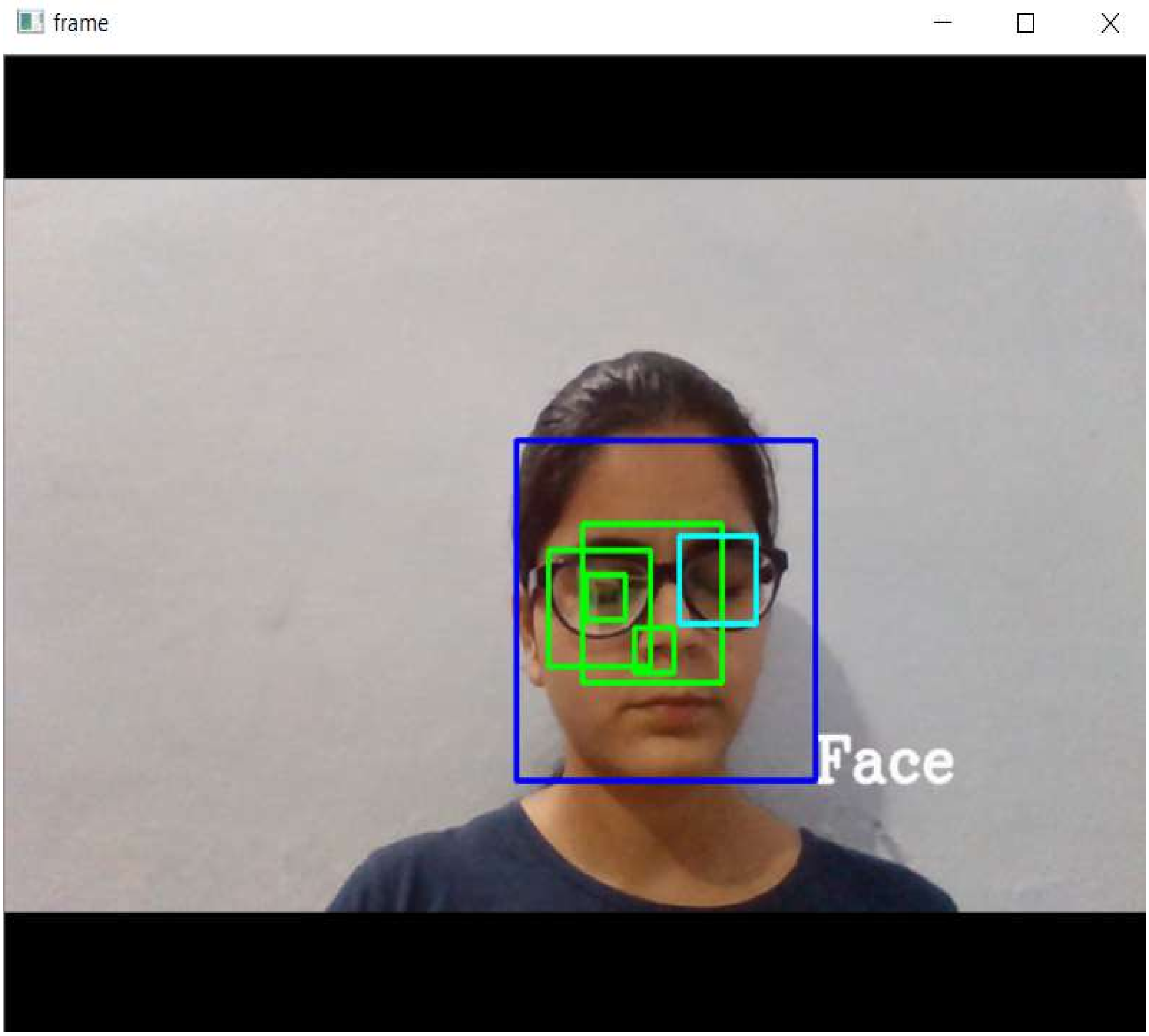
8.1.)



8.2.)



8.3.)



9.) Testing

9.1.) Test Cases

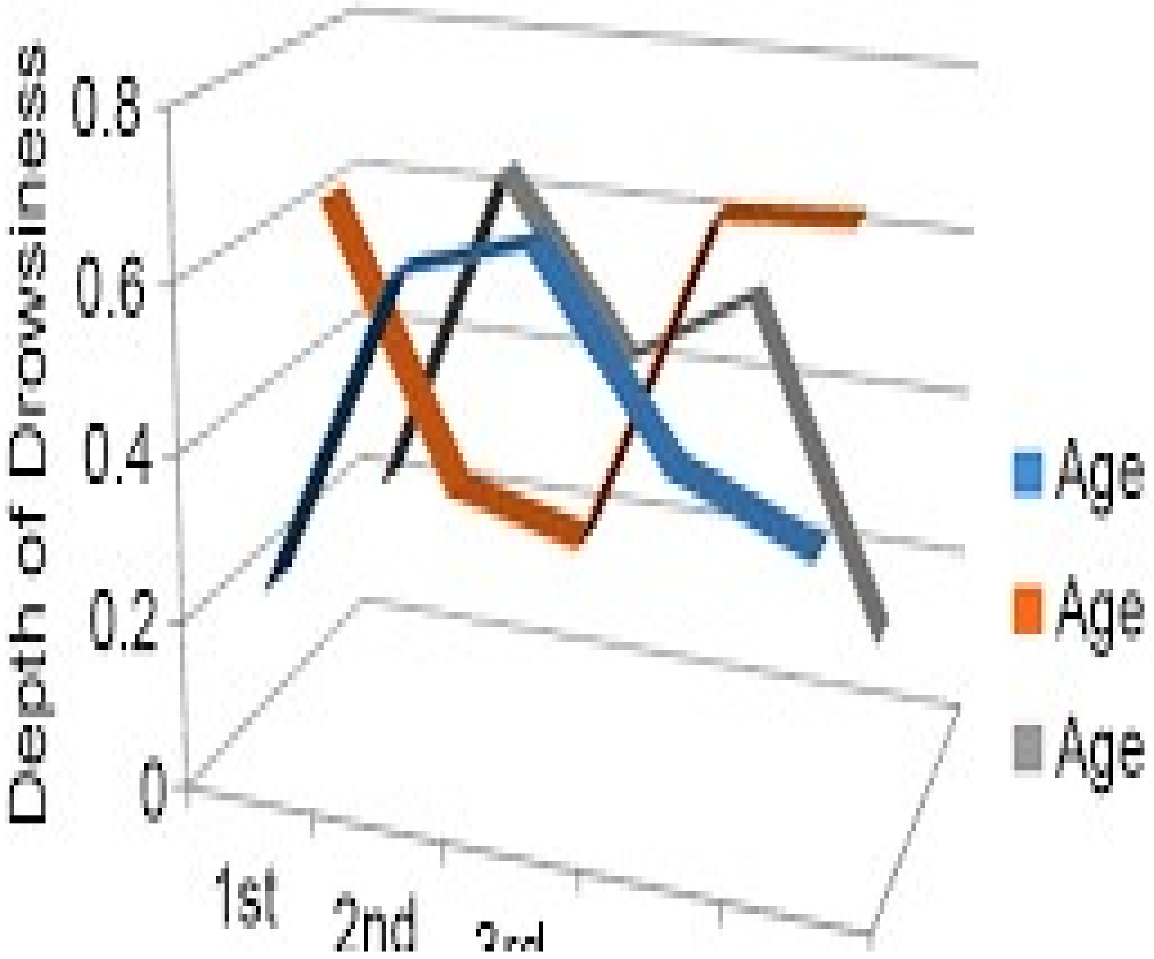


Fig. Test Cases

9.2.) Methodology

The different types of methodologies have been developed to find out drowsiness.

9.2.1) Physiological Level Approach

This technique is an intrusive method wherein electrodes are used to obtain pulse rate, heart rate and brain activity information. ECG is used to calculate the variations in heart rate and detect different conditions for drowsiness. The correlation between different signals such as ecg (electrocardiogram), EEG (electroencephalogram), and EMG (electromyogram) are made and then the output is generated whether the person is drowsy or not.

9.2.2) Behavioral Based Approach

In this technique eye blinking frequency, head pose, etc. of a person is monitored through a camera and the person is alerted if any of these drowsiness symptoms are detected.

9.3) Technology

The different types of technologies have been used to find out drowsiness.

9.3.1) Python

Python is an open source, high-level programming language. Python is a powerful language that we can use to create games, write GUIs, and develop web applications. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms.

Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to www browsers to games.

Features of Python

- **Easy to learn :** Python has few keywords, simple structure, and a clearly defined syntax. This allows us to pick up the language quickly.
- **Easy to read :** Python code is more clearly defined and visible to the eyes.
- **Easy to maintain :** Python's source code is fairly easy-to-maintain.
- **A broad standard library :** Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode :** Python has support for an interactive mode which allows interactive testing and debugging.

- **Portable :** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Databases :** Python provides interfaces to all major commercial databases.

- **GUI Programming :** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable :** Python provides a better structure and support for large programs than shell scripting.

9.3.2) Machine Learning

Machine learning is the kind of programming which gives computers the capability to automatically learn from data without being explicitly programmed. This means in other words that these programs change their behavior by learning from data. Python is clearly one of the best languages for machine learning. Python does contain special libraries for machine learning namely scipy, pandas and numpy which great for linear algebra and getting to know kernel methods of machine learning. The language is great to use when working with machine learning algorithms and has easy syntax relatively.

10.) Coding

```
import numpy

from pygame import mixer

import time

import cv2

from tkinter import *

import tkinter.messagebox

root=Tk()

root.geometry('500x570')

frame = Frame(root, relief=RIDGE, borderwidth=2)

frame.pack(fill=BOTH,expand=1)

root.title('Driver Cam')

frame.config(background='light blue')

label = Label(frame, text="Driver Cam",bg='light
blue',font=('Times 35 bold'))

label.pack(side=TOP)
```

```
filename = PhotoImage(file="demo.png")

background_label = Label(frame,image=filename)

background_label.pack(side=TOP)




def hel():

   help(cv2)


def Contri():

   tkinter.messagebox.showinfo("Contributors","\ Made by
Shruti\n")



def anotherWin():
   tkinter.messagebox.showinfo("About",'Driver Cam version
v1.0\n Made Using\n-OpenCV\n-Numpy\n-Tkinter\n In
Python 3')




menu = Menu(root)

root.config(menu=menu)


subm1 = Menu(menu)

menu.add_cascade(label="Tools",menu=subm1)

subm1.add_command(label="Open CV Docs",command=hel)


subm2 = Menu(menu)

menu.add_cascade(label="About",menu=subm2)
```

```
subm2.add_command(label="Driver
Cam",command=anotherWin)
subm2.add_command(label="Contributors",command=Contri)




def exitt():
   exit()



def web():
   capture =cv2.VideoCapture(0)
   while True:
      ret,frame=capture.read()
      gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
      cv2.imshow('frame',frame)
```

```
    if cv2.waitKey(1) & 0xFF ==ord('q'):
      break
  capture.release()
  cv2.destroyAllWindows()

def webrec():
  capture =cv2.VideoCapture(0)
  fourcc=cv2.VideoWriter_fourcc(*'XVID')
  op=cv2.VideoWriter('Sample1.avi',fourcc,11.0,(640,480))
  while True:
    ret,frame=capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('frame',frame)
    op.write(frame)
    if cv2.waitKey(1) & 0xFF ==ord('q'):
      break
  op.release()
  capture.release()
  cv2.destroyAllWindows()


def webdet():
  capture =cv2.VideoCapture(0)
  face_cascade =
cv2.CascadeClassifier('lbpcascade_frontalface.xml')
  eye_glass =
cv2.CascadeClassifier('haarcascade_eye_tree_eyeglasses.xml')



  while True:
    ret, frame = capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray)



    for (x,y,w,h) in faces:
        font = cv2.FONT_HERSHEY_COMPLEX

cv2.putText(frame,'Face',(x+w,y+h),font,1,(250,250,250),2,cv
2.LINE_AA)
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]



        eye_g = eye_glass.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in eye_g:
```

```
cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)



       cv2.imshow('frame',frame)

       if cv2.waitKey(1) & 0xff == ord('q'):

          break

   capture.release()

   cv2.destroyAllWindows()

def webdetRec():

   capture =cv2.VideoCapture(0)

   face_cascade =
cv2.CascadeClassifier('lbpcascade_frontalface.xml')

   eye_glass =
cv2.CascadeClassifier('haarcascade_eye_tree_eyeglasses.xml')

   fourcc=cv2.VideoWriter_fourcc(*'XVID')
```

```
 op=cv2.VideoWriter('Sample2.avi',fourcc,9.0,(640,480))


 while True:
    ret, frame = capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray)



    for (x,y,w,h) in faces:
        font = cv2.FONT_HERSHEY_COMPLEX

cv2.putText(frame,'Face',(x+w,y+h),font,1,(250,250,250),2,cv
2.LINE_AA)
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]
        eye_g = eye_glass.detectMultiScale(roi_gray)

        for (ex,ey,ew,eh) in eye_g:


    cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

      op.write(frame)

      cv2.imshow('frame',frame)

      if cv2.waitKey(1) & 0xff == ord('q'):

        break

  op.release()

  capture.release()

  cv2.destroyAllWindows()
```

```
def alert():
   mixer.init()
   alert=mixer.Sound('beep-07.wav')
   alert.play()
   time.sleep(0.1)
   alert.play()


def blink():
   capture =cv2.VideoCapture(0)
   face_cascade =
cv2.CascadeClassifier('lbpcascade_frontalface.xml')
   eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
   blink_cascade =
cv2.CascadeClassifier('CustomBlinkCascade.xml')


   while True:
    ret, frame = capture.read()

    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(gray)


    for (x,y,w,h) in faces:

      font = cv2.FONT_HERSHEY_COMPLEX


cv2.putText(frame,'Face',(x+w,y+h),font,1,(250,250,250),2,cv2.LINE_AA)

      cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

      roi_gray = gray[y:y+h, x:x+w]

      roi_color = frame[y:y+h, x:x+w]


      eyes = eye_cascade.detectMultiScale(roi_gray)

      for(ex,ey,ew,eh) in eyes:
```

```
cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)


         blink = blink_cascade.detectMultiScale(roi_gray)

         for(eyx,eyy,eyw,eyh) in blink:


cv2.rectangle(roi_color,(eyx,eyy),(eyx+eyw,eyy+eyh),(255,25
5,0),2)

            alert()

      cv2.imshow('frame',frame)

      if cv2.waitKey(1) & 0xFF ==ord('q'):

          break




   capture.release()

   cv2.destroyAllWindows()
```

```
but1=Button(frame,padx=5,pady=5,width=39,bg='white',fg='b
lack',relief=GROOVE,command=web,text='Open
Cam',font=('helvetica 15 bold'))

but1.place(x=5,y=104)


but2=Button(frame,padx=5,pady=5,width=39,bg='white',fg='b
lack',relief=GROOVE,command=webrec,text='Open Cam &
Record',font=('helvetica 15 bold'))

but2.place(x=5,y=176)


but3=Button(frame,padx=5,pady=5,width=39,bg='white',fg='b
lack',relief=GROOVE,command=webdet,text='Open Cam &
Detect',font=('helvetica 15 bold'))

but3.place(x=5,y=250)
```

```
but4=Button(frame,padx=5,pady=5,width=39,bg='white',fg='b
lack',relief=GROOVE,command=webdetRec,text='Detect &
Record',font=('helvetica 15 bold'))
but4.place(x=5,y=322)


but5=Button(frame,padx=5,pady=5,width=39,bg='white',fg='b
lack',relief=GROOVE,command=blink,text='Detect Eye Blink
& Record With Sound',font=('helvetica 15 bold'))
but5.place(x=5,y=400)


but5=Button(frame,padx=5,pady=5,width=5,bg='white',fg='bl
ack',relief=GROOVE,text='EXIT',command=exitt,font=('helve
tica 15 bold'))
but5.place(x=210,y=478)


root.mainloop()
```

11.) Conclusion

The real time system has been successfully created to detect the face and hence the eyes and mouth of the driver to check whether he is blinking or yawning to acquire information about his level of alertness. The system has been tried and tested in different lighting conditions and with different people with varied facial characteristics. It has been experimentally found that absolute accuracy is achieved when the lighting conditions are bright and favourable. The biggest drawback experienced till now is the presence of beard or sunglasses or spectacles on the driver's face. This interferes with the detection of eyes and mouth and may lead to false triggering. This system is real time and checks the state of the driver all through the journey.

12.) Limitations

The limitations of the system are as follows:

- **Use of spectacles**

  In case the user uses spectacle then it is difficult to detect the state of the eye. As it hugely depends on light hence reflection of spectacles may give the output for a closed eye as opened eye. Hence for this purpose the closeness of eye to the camera is required to avoid light.

- **Multiple face problem**

  If multiple face arises in the window then the camera may detect more number of faces undesired output may appear. Because of different condition of different faces. So, we need to make sure that only the driver face come within the range of the camera. Also, the speed of detection reduces because of operation on multiple faces.

- **Dependence on ambient light**

  The system has been tried and tested in different lighting conditions and with different people with varied facial characteristics. It has been experimentally found that absolute accuracy is achieved when the lighting conditions are bright and favourable.

13.) Future Scope

The future scope for this project includes increasing the speed of operation of the system and hence increase the accuracy rate. Further, this concept can be extended to provide an inexpensive solution for commercial vehicles.

The difficulties faced due to bad lighting that may occur while driving during night time is a potent problem that needs to be taken care of. Bearded men and people wearing spectacles too should be able to use this system accurately. This is drawback that needs to be mended as future scope.

Another area for further work would include involving the dynamics and working of the vehicle when the buzzer is set off. This, if used with extreme discretion, can help further reduce road accidents. The speed of the vehicle can be reduced or a remote terminal can be warned about fatigue detected in the driver.

14.) Bibliography

IEEE standard Journal Paper reference,

- Facial Features Monitoring for Real Time Drowsiness Detection by Manu B.N, 2016 12th International Conference on Innovations in Information Technology (IIT) [Pg. 78-81] https://ieeexplore.ieee.org/document/7880030

- Real Time Drowsiness Detection using Eye Blink Monitoring by Amna Rahman Department of Software Engineering Fatima Jinnah Women University 2015 National Software Engineering Conference (NSEC 2015) https://ieeexplore.ieee.org/document/7396336

Names of Websites reference,

- https://realpython.com/face-recognition-with-python/
- https://docs.opencv.org /3.4/d7/d8b/tutorial_py_face_detection.html
- https://www.learnopencv.com/training-better-haar-lbp-cascade-eye-detector-opencv/
- https://docs.opencv.org/master/d9/df8/tutorial_root.ht
- https://docs.opencv.org/master/d9/df8/tutorial_root.ht
- https://docs.opencv.org/master/d9/df8/tutorial_root.ht