“DROWSINESS DETECTION ”

A

Minor Project Report

Submitted by

Anubhav Sharma

17BCON430

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

At

JECRC UNIVERSITY, JAIPUR

May 2020

Acknowledgments

Many people have supported me, in different ways, during the work with the thesis. I'd like to thank my guide Mrs Deepika Singh Shekhawat & HOD Dr. Naveen Hemrajni for their kind and active support and valuable guidance during the work process. My family has as always offered me their unconditional support, thank you! I have taken efforts in the Minor Project. However, it would not have been possible without the kind support and many individuals and organizations. I would like to extend my sincere thanks to each and every member related to JECRC University.

Anubhav Sharma

17BCON430

Candidate's Declaration

I, **Anubhav Sharma**, bearing roll number **17BCON430**, hereby declare that the work which is being presented in the Minor Project, entitled "**Drowsiness Detection**" in partial fulfilment for award of Degree of "**Bachelor of Technology**" in Department of **Computer Science Engineering** is submitted to the Department Computer Science & Engineering, JECRC University is a record of Minor Project work carried under the Guidance of Guide name, Department Computer Science & Engineering.

I have not submitted the matter presented in this work anywhere for the award of any other Degree.

Anubhav Sharma

Computer Science

Enrolment No.: 17BCON430

JECRC UNIVERSITY JAIPUR

Ramchandrapura, Sitapura Industrial Area Extn., Jaipur-303905(Raj.) India
www.jecrcuniversity.edu.in

Date: 24th May,2020

CERTIFICATE

Certified that the Project Report entitled "**Drowsiness Detection**" submitted by **Anubhav Sharma** bearing roll no.**17BCON430**. In partial fulfillment of the requirements for the award of the degree of Bachelor of Technology of JECRC University, Jaipur is a record of the student's own work carried out under my supervision and guidance. To the best of our knowledge, this Minor Project work has not been submitted to JECRC University or any other university for the award of the degree. It is further understood that by this certificate the undersigned does not endorse or approve of any statement made, opinion expressed, or conclusion drawn therein but approves Minor Project for the purpose for which it is submitted.

(Guide) (HOD, CSE)

Abstract

An important application of machine vision and image processing could be driver drowsiness detection systems due to its high importance. In recent years there have been many research projects reported in the literature in this field. In this paper, unlike conventional drowsiness detection methods, which are based on the eye states alone, we used facial expressions to detect drowsiness. There are many challenges involving drowsiness detection systems. Among the important aspects are: change of intensity due to lighting conditions, the presence of glasses and beard on the face of the person. 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.

LIST OF FIGURE

FIGURE Page

TABLE OF CONTENTS

CHAPTER 1

1.1. Introduction

Machine Learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

The name machine learning was coined in 1959 by Arthur Samuel. Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. "This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?". In Turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed. Machine Learning came to existence with the concept of artificial intelligence which gives the machine to become intelligent. It gives the machine a new power to learn and to predict things in the near future based on the learning.

2016 is the year when artificial intelligence (AI) came of age. With AlphaGo defeating the top human Go players, this has truly witnessed the huge potential in artificial intelligence (AI), and have begun to expect more complex, cutting-edge AI technology in many applications, including driverless cars, medical care, finance, etc. Today, AI technology is showing its strengths in almost every industry and walks of life. However, looking back at the development of AI, it is inevitable that the development of AI has experienced several ups and downs. AlphaGo in 2016 used a total of 3,00,000 games as training data to achieve the excellent results. With AlphaGo's success, people naturally hope that big data-driven AI like AlphaGo will be realized soon in all aspects of our lives. However, the real-world situations are somewhat disappointing: with the exception

of few industries, most fields have only limited data or poor quality data, making the realization of AI technology more difficult than we thought. Would it be possible to fuse the data together in a common site, by transporting the data across organizations? In fact, it is very difficult, if not impossible, in many situations to break the barriers between data sources. In general, the data required in any AI project involves multiple types. For example, in an AI-driven product recommendation service, the product seller has information about the product, data of the user's purchase, but not the data that describes the user's purchasing ability and payment habits. In most industries, data exists in the form of isolated islands. Due to industry competition, privacy security, and complicated administrative procedures, even data integration between different departments of the same company faces heavy resistance. It is almost impossible to integrate the data scattered around the country and institutions, or the cost is prohibited.

1.1.1. About Project

The project is about the detection of drowsiness in humans. It is done using the Artificial Neural Network, in which we have to train an artificial neural network to do the work of detection of drowsiness in humans. After the training of the model the ready model is used in the world to implement the real-life circumstance. It works like For detecting the face since the camera is focused on the automobile driver, we can avoid processing the image at the corners thus reducing a significant amount of processing required. Once the region of interest is defined, the face has been detected, the region of interest is now the face, as the next step involves detecting eyes. To detect the eyes, instead of processing the entire face region, we mark a region of interest within the face region which further helps in achieving the primary goal of the proposed system. Next we make use of Haarcascade Xml file constructed for eye detection, and detect the eyes by processing only the region of interest. Once the eyes have been detected, the next step is to determine whether the eyes are in an open/closed state, which is achieved by extracting and examining the pixel values from the eye region. If the eyes are detected to be open, no action is taken. But, if eyes are detected to be closed continuously for two seconds according to the code, that is a particular number of frames depending on the frame rate, then it means that the automobile driver is feeling drowsy and a sound alarm is triggered.

1.2. Literature Survey

Artificial intelligence is leading in every field of the world nowadays. The project depends on the artificial neural network in which we have trained a neural network to detect whether the person is drowsy or not. The previous version of the project was studied over the IEEE research paper it detects the face of the human and the eyes of the human. In this project we detect the facial expression of the human using the Haar Cascade algorithm. We also learn eye activity from Antoine Picot, Alice Caplier, and Sylvie Charbonnier's on-line detection of drowsiness paper in which we implement the opening and closing of eyes and also give a score which gives an indication of drowsiness.

1.3. Software Requirement Analysis

The software made by our team is required for the detection of the drowsiness of the human whether the human is drowsy or not. It helps policemen to catch people easily when people get drowsy and drive the vehicle. It also helps in many ways like it also detects whether the human gets sleepy or not. It alerts people whenever they get sleepy in between some tasks. It also helps in Bars, hotels, and shopkeepers whenever any drowsy person comes near the cash counter for robbery or fight.

1.4. Problem Occurs

In the world of human accidents, frequent things happen generally, to avoid any circumstance of the accident humans create rules one such rule is not to drive after drinking but many humans violate this law. Police always try to catch them red-handed but get failed due to lack of technology which detects instantly whether the human is drowsy or not. This lead to many accidents when the driver is driving the vehicle

Another challenge is faced by the shopkeeper and the bars and hotels where the manager gets troubled by the over drowsy person and till they detect that the person is drowsy, it's late and they might get robbed or beaten by the drowsy person.

1.5. Solution : Module & Functionality

See all the problems occur over We all team members come together and get to the solution to develop an alarm system that detects the person and instantly detects whether the person is drowsy or not. We use the Artificial Neural Network and train it to recognize the facial expression and eye activity of the human.

In this project, we check the facial expression and eye activity using following modules :-

- **OpenCV library**
- **Keras library**
- **OS library**
- **Numpy library**
- **Pygame library**
- **Time library**

All the modules are python libraries which work differently for this project. The functionality of the libraries are as follow:-

OS and Numpy library is used for data pre-processing and batch making.

OpenCV is used to take the input from the Camera and to give Output on the computer screen.

Keras is used for working with the neural network model.

Pygame is used for the sound/alarm and the red border of the screen also shows the score of the human’s drowsiness.

Time is used for making the model work in real-time.

Functionality :- The Model is trained over many images now when it comes to the work it starts watching over the human and detects its facial expression also it detects the eye activity of the human. When the score of the human goes above 16 the alarm is triggered and the screen becomes red.

Video acquisition: Video acquisition mainly involves obtaining the live video feed of the automobile driver. Video acquisition is achieved, by making use of a camera.

Dividing into frames: This module is used to take live video as its input and convert it into a series of frames/ images, which are then processed.

Face detection: The face detection function takes one frame at a time from the frames provided by the frame grabber, and in each and every frame it tries to detect the face of the automobile driver. This is achieved by making use of a set of predefined Haar Cascade samples.

Eyes detection: Once the face detection function has detected the face of the automobile driver, the eyes detection function tries to detect the automobile driver's eyes. This is achieved by making use of a set of predefined Haar Cascade samples.

Drowsiness detection: After detecting the eyes of the automobile driver, the drowsiness detection function detects if the automobile driver is drowsy or not, by taking into consideration the state of the eyes, that is, open or closed and the blink rate.

As the proposed system makes use of OpenCV libraries, there is no necessary minimum resolution requirement on the camera. The schematic representation of the algorithm of the proposed system is depicted in Fig 2. In the proposed algorithm, first video acquisition is achieved by making use of an external camera placed in front of the automobile driver. The acquired video is then converted into a series of frames/images. The next step is to detect the automobile driver's face, in each and every frame extracted from the video.

As indicated in Figure 2, we start by discussing face detection which has 2 important functions (a) Identifying the region of interest, and (b) Detection of the face from the above region using Haar Cascade. To avoid processing the entire image, we mark the region of interest. By considering the region of interest it is possible to reduce the amount of processing required and also speeds up the processing, which is the primary goal of the proposed system.

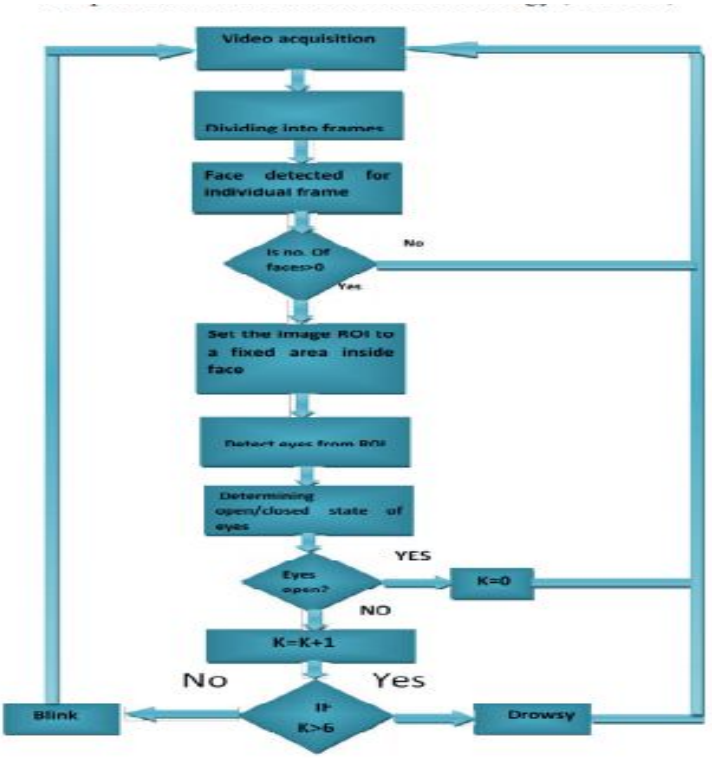


Fig1.1: Flow Chart of Drowsiness Detection

For detecting the face, since the camera is focused on the automobile driver, we can avoid processing the image at the corners thus reducing a significant amount of processing required. Once the region of interest is defined face has been detected, the region of interest is now the face, as the next step involves detecting eyes. To detect the eyes, instead of processing the entire face region, we mark a region of interest within the face region which further helps in achieving the primary goal of the proposed system. Next we make use of Haarcascade Xml file constructed for eye detection, and detect the eyes by processing only the region of interest. Once the eyes have been detected, the next step is to determine whether the eyes are in an open/closed state, which is achieved by extracting and examining the pixel values from the eye region. If the eyes are detected to be open, no action is taken. But, if eyes are detected to be closed continuously for two seconds according to the code, that is a particular number of frames depending on the frame rate, then it means that the automobile driver is feeling drowsy and a sound alarm is triggered.

CHAPTER 2

2.1. Software Design

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

Here, we design software that detects whether a person is drowsy or not. So, to prevent unnecessary accidents from happening in the near future. It also helps the police to make sure whether the driver is drunk or not.

2.2. UML Design

UML is an acronym which stands for Unified Modeling Language. It is a diagrammatic representation of software components. Mainly UML is used for general-purpose modeling language in software development. In this project, we develop many files that are collaboratively shown in the UML diagrams.

2.2.1. Class Diagram

A Class diagram gives an overview of a system by showing its classes and the relationships among them. Class diagrams are static -- they display what interacts but not what happens when they do interact. This class diagram models whether a person is drowsy or not. The central class is the Detector. Associated with it is the Predictor making the prediction whether the person is drowsy or not.

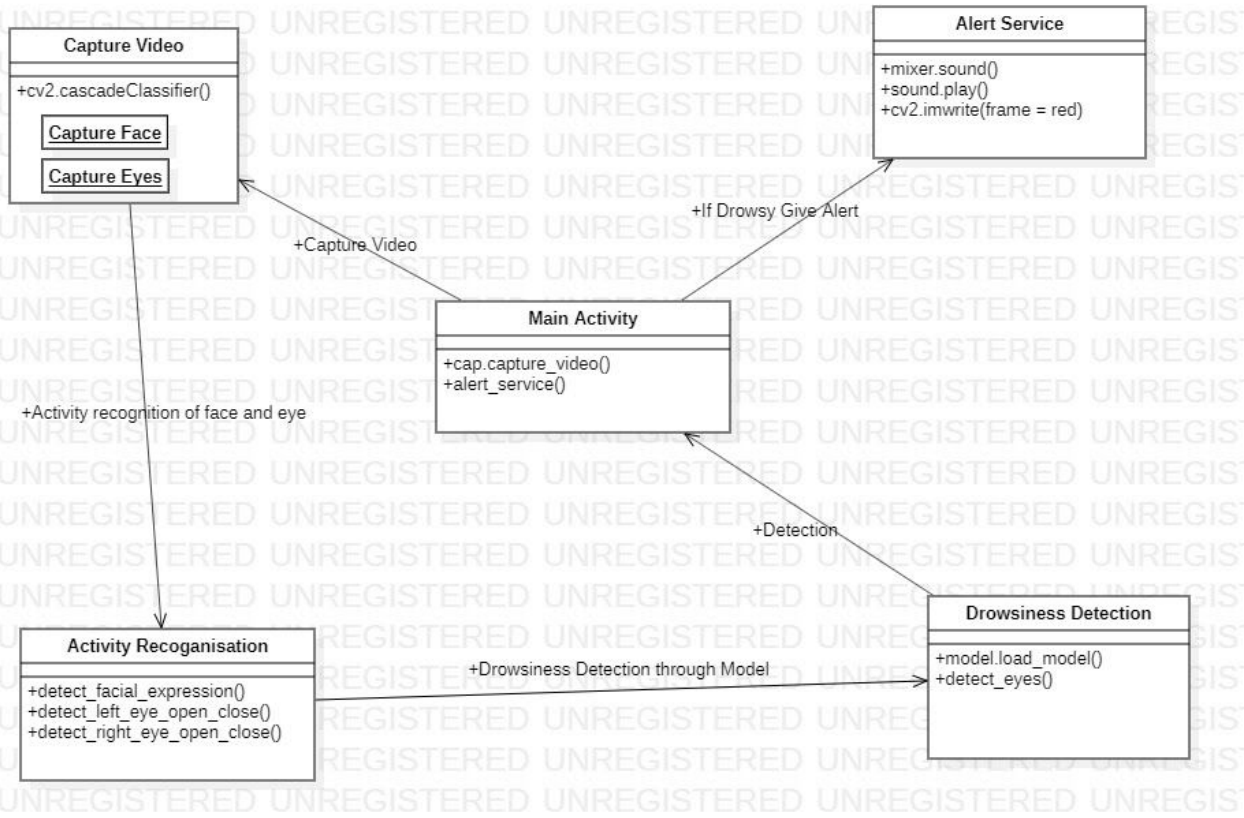


Fig2.1: Class Diagram

2.2.2. Interaction Diagram

This interactive behavior is represented in UML by two diagrams known as the Sequence diagram and collaboration diagram. The basic purpose of both diagrams are similar

I. Sequence Diagram

The Sequence diagram has three diagrams which include objects that are Camera, person to detect, and the output screen.

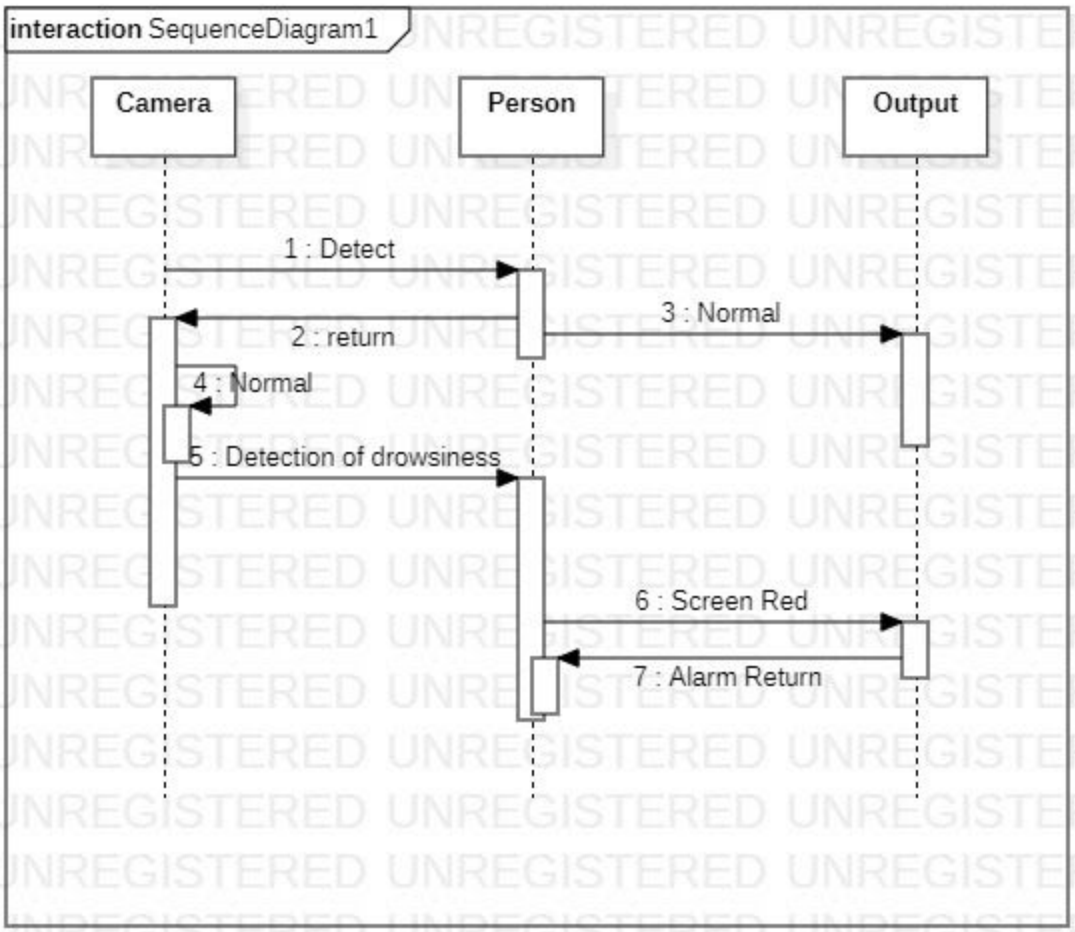


Fig2.2: Sequence Diagram

II. Collaborative Diagram

The second interaction diagram is the collaboration diagram. It shows the object organization as seen in the following diagram. In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another.

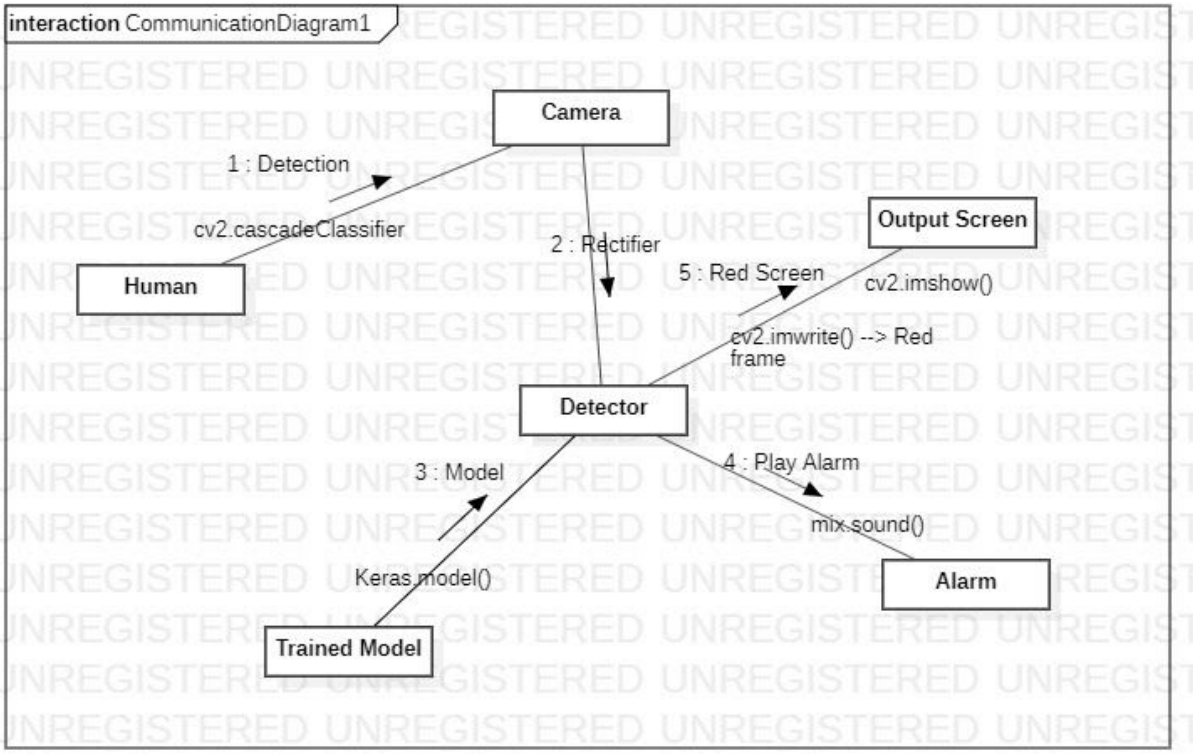


Fig2.3: Collaboration Diagram

2.2.3. Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating. Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consist of actors, use cases, and their relationships. The diagram is used to model the system/subsystem of an application. A single-use case diagram captures a particular functionality of a system.

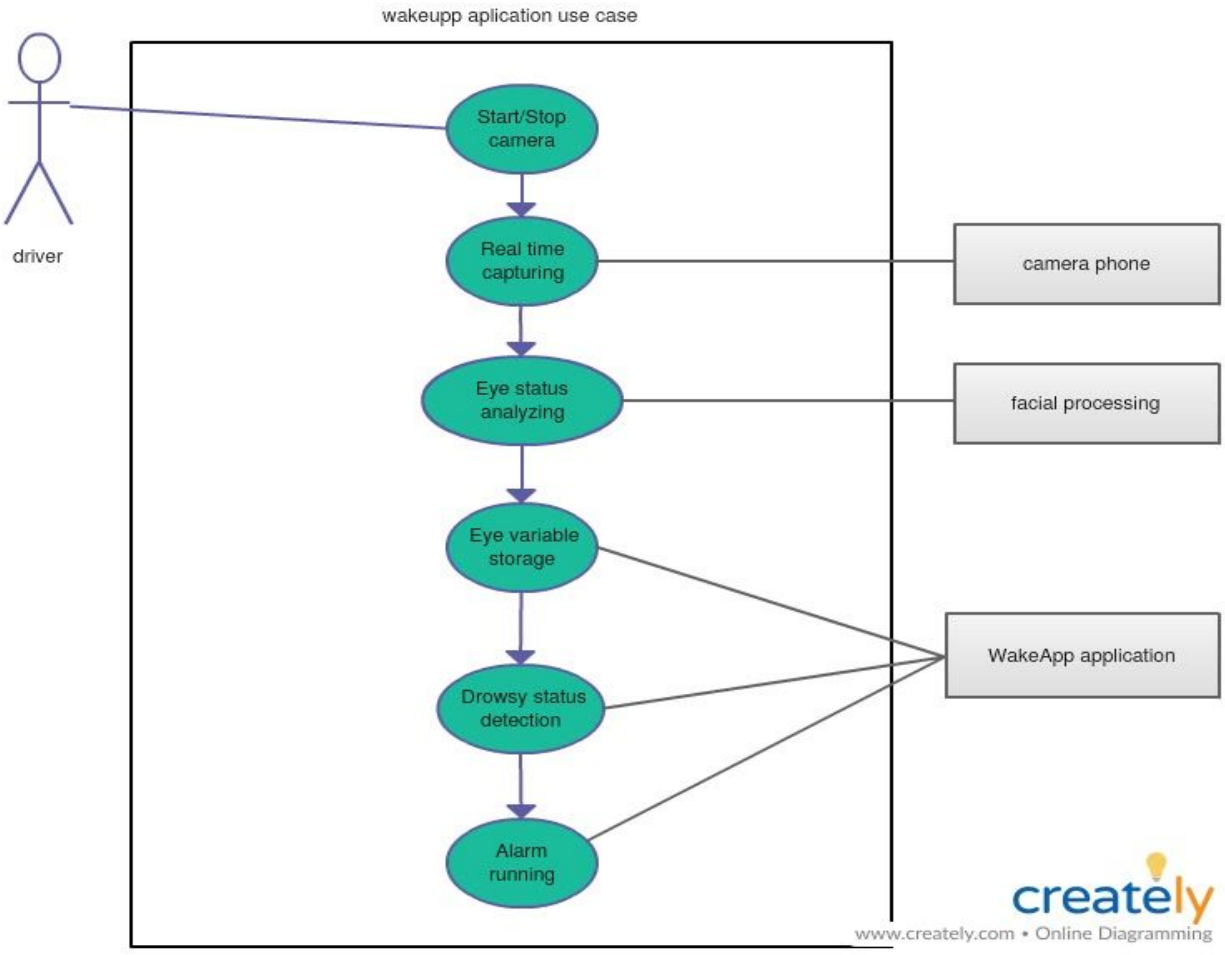


Fig2.4: Use Case Diagram

2.3. System Diagram

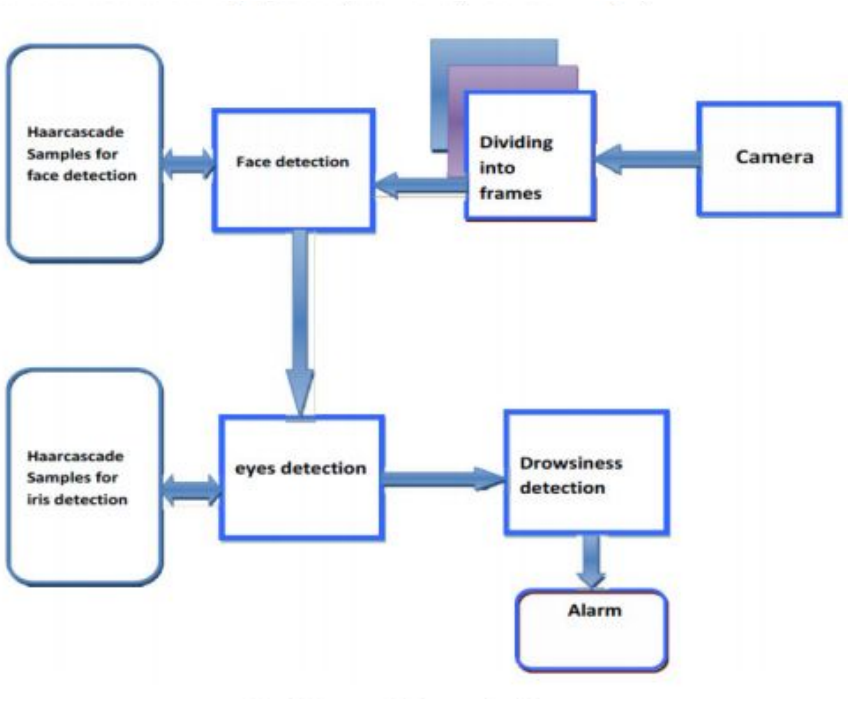


Fig2.5: System Diagram

System Diagrams are models used to visually express the dynamic forces acting upon the components of a process and the interactions between those forces. System Diagrams are more than process flow charts. In this system diagram, we have to represent the full process of the Drowsiness detection starting from the video acquaintance to the drowsiness detection

CHAPTER 3

3.1. Software Requirement

The language used to make the software in Python. So the requirements to run the file are as follow:-

1. Windows OS/ Linux OS/ Mac OS
2. Python 3.7.6
3. Libraries Require:-
    - OS Library
    - Numpy Library
    - Keras Library
    - shutil Library
    - Random Library
    - Matplotlib Library
    - Pygame Library
    - OpenCV Library
    - Time Library

3.2. Hardware Requirement

The hardware required to run the software are as follow:-

1. Intel i3 7th gen processor or Above
2. Nvidia Geforce 900M series of Above/AMD Risen 3 or above
3. Web Camera

CHAPTER 4

4.1. Coding/Code Templates

There are two main files which are used in the developing of the software they are as follow:-

1. model.py
2. drowsiness detection.py

The model.py contains the trained model which is used to give the weights to detect whether the person is drowsy or not

```
#Importing the libraries

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

#Defining Functions and variables

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)

# Defining Model
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')
])

#Compiling Model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accura
cy'])
```

```
#Fitting the Model
model.fit_generator(train_batch,
validation_data=valid_batch,epochs=15,steps_per_epoch=SPE
,validation_steps=VS)

#Saving the model in cnnCat2.h5 file
model.save('models/cnnCat2.h5', overwrite=True)
```

The second file is the of the drowsiness detection in which the trained model function to get the drowsiness

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

#Calling Functions for Facial and Eye Activity variables
mixer.init()
sound = mixer.Sound('alarm.wav')

face=cv2.CascadeClassifier('haarcascadefiles\haarcascade_frontalface_alt.xml')
leye=cv2.CascadeClassifier('haarcascadefiles\haarcascade_lefteye_2splits.xml')
reye=cv2.CascadeClassifier('haarcascadefiles\haarcascade_righteye_2splits.xml'
)

#defining labels
lbl=['Close','Open']

#Calling model and defining variables
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]

#Algorithm to detect
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.FILL
        ED)

        #Detection of facial expression
        for (x,y,w,h) in faces:
                cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )

        #Detection of right eye
        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

        #detection of left eye
        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,c
        v2.LINE_AA)

        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)
        #Starting alarm and give red alert on screen
        if(score>15):
                #person is feeling sleepy so we beep the alarm
                cv2.imwrite(os.path.join(path,'image.jpg'),frame)
                sound.play()

                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()
```

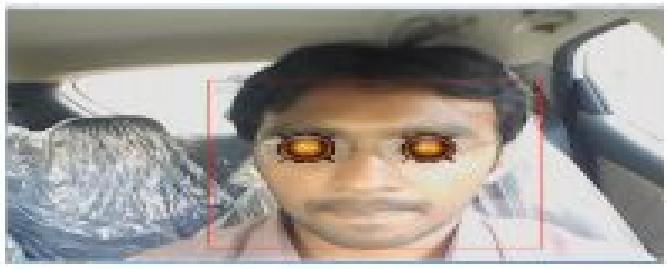
This is the second file which is used to detect the drowsiness of the human.

CHAPTER 5

5.1. Testing

The tests were conducted in various conditions including:
1. Different lighting conditions.
2. Drivers' posture and position of the automobile drivers face.
3. Drivers with spectacles

5.1.1. Test case 1: When there is ambient light



.

Fig5.1: Test1: Ambient Light

Result: As shown in Fig 3, when there is an ambient amount of light, the automobile driver's face and eyes are successfully detected.

5.1.2. Test case 2: Position of the automobile drivers face

5.1.2.1 Center Positioned

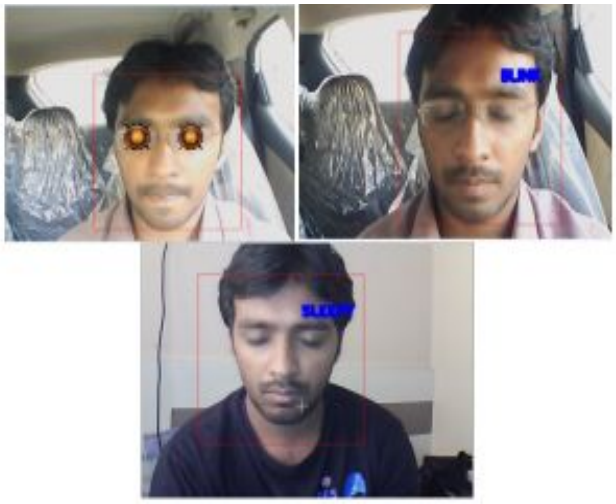


Fig5.2: Test2: Center Position

RESULT: As shown in Fig 4, When the automobile driver's face is positioned at the Centre, the face, eyes, eye blinks, and drowsiness was successfully detected.

5.1.2.2 Right Positioned:

Fig5.3: Test2.2: Right Position

RESULT: As shown in Fig 5, When the automobile driver's face is positioned at the Right, the face, eyes, eye blinks, and drowsiness was successfully detected.

5.1.2.3 Left Positioned:

Fig5.4: Test2.3: Left Position

RESULT: As shown in the screen snapshot in Fig 6,when the automobile driver's face is positioned at the Left, the face, eyes, eye blinks, and drowsiness was successfully detected.

5.1.3 Test case 3: When the automobile driver is wearing spectacles

Fig5.5: Test3: Driver Wear Spectacles

RESULT : As shown in the screen snapshot in Fig 7, When the automobile driver is wearing spectacles, the face, eyes, eye blinks, and drowsiness was successfully detected.

5.1.4. Test case 4: When the automobile driver’s head tilted

Fig5.6: Test4: Head Tilted

RESULT: As shown in the screen snapshot in Fig 8, when the automobile driver's face is tilted for more than 30 degrees from the vertical plane, it was observed that the detection of face and eyes failed. The system was extensively tested even in real-world scenarios, this was achieved by placing the camera on the visor of the car, focusing on the automobile driver. It was found that the system gave positive output unless there was any direct light falling on the camera.

5.2. Output Screen

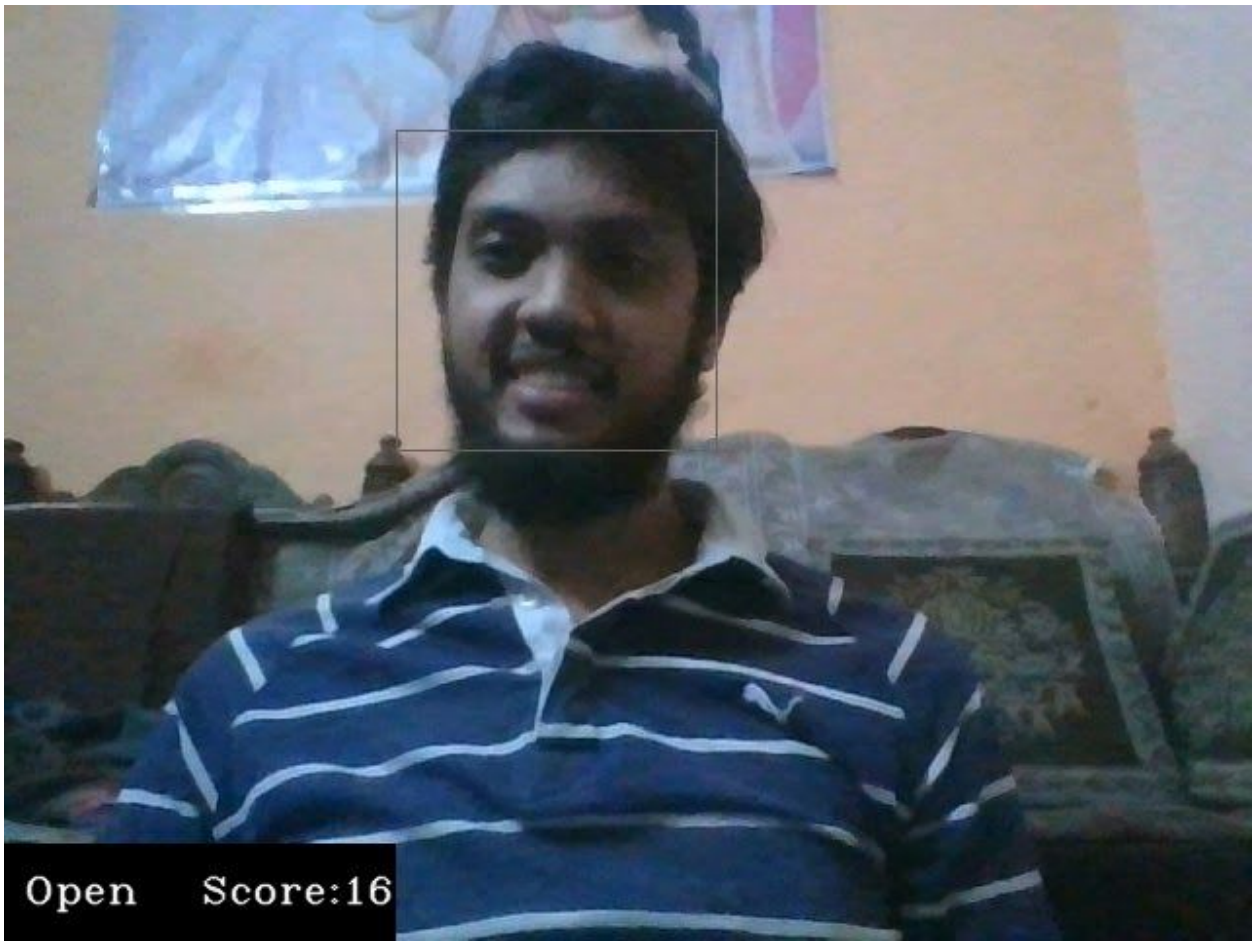


Fig5.7: Output Screen from software

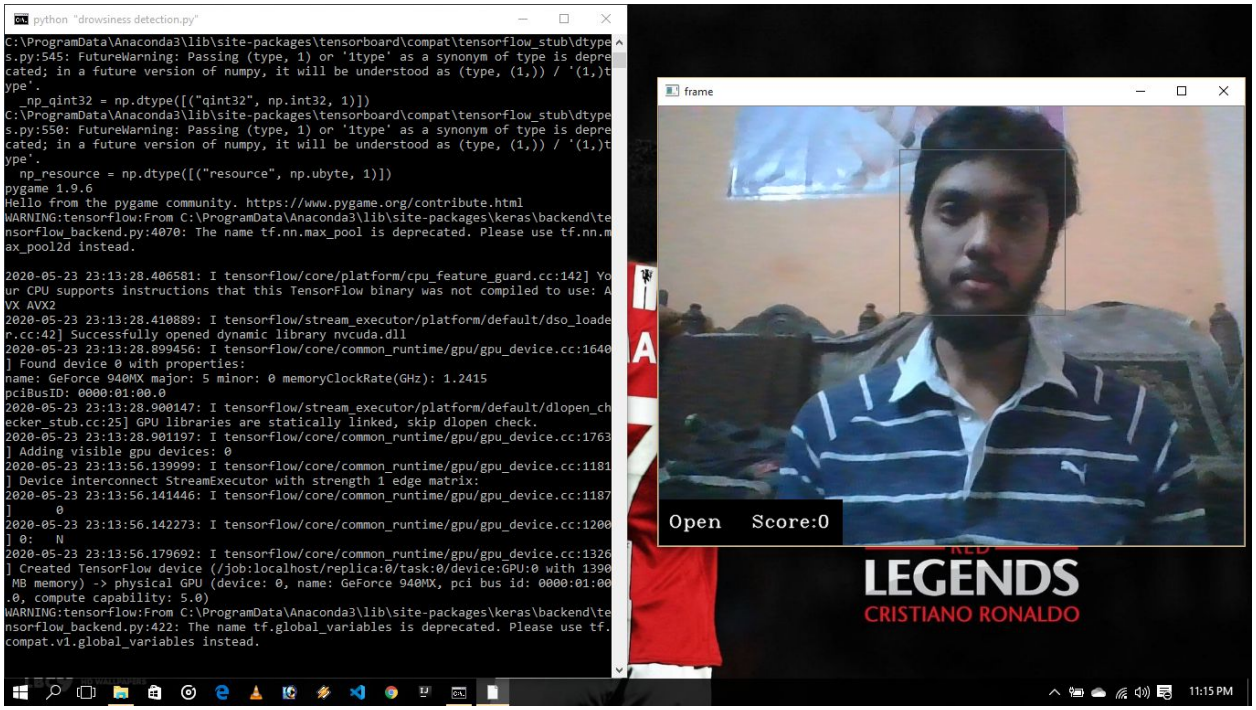


Fig5.8: Output Screen1: Eye Open(Center Position)

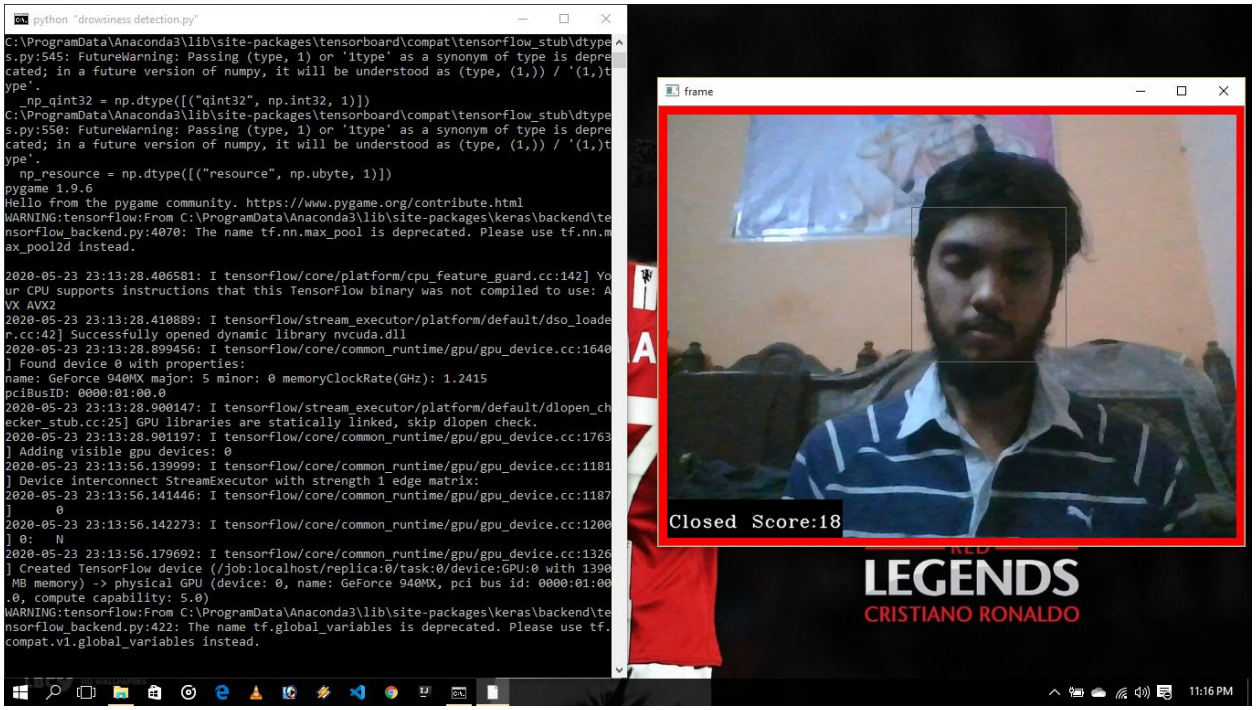


Fig5.9: Output Screen1: Eye Close (Center Position)

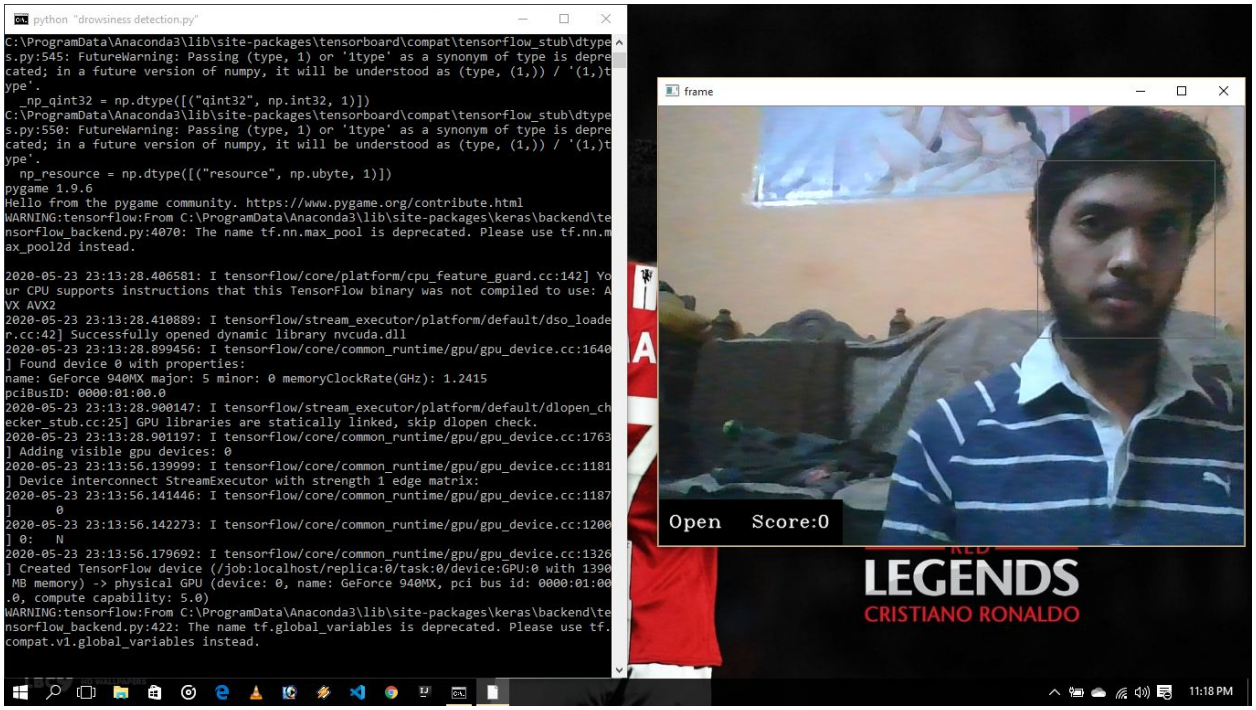


Fig5.10: Output Screen1: Eye Open (Left Position)

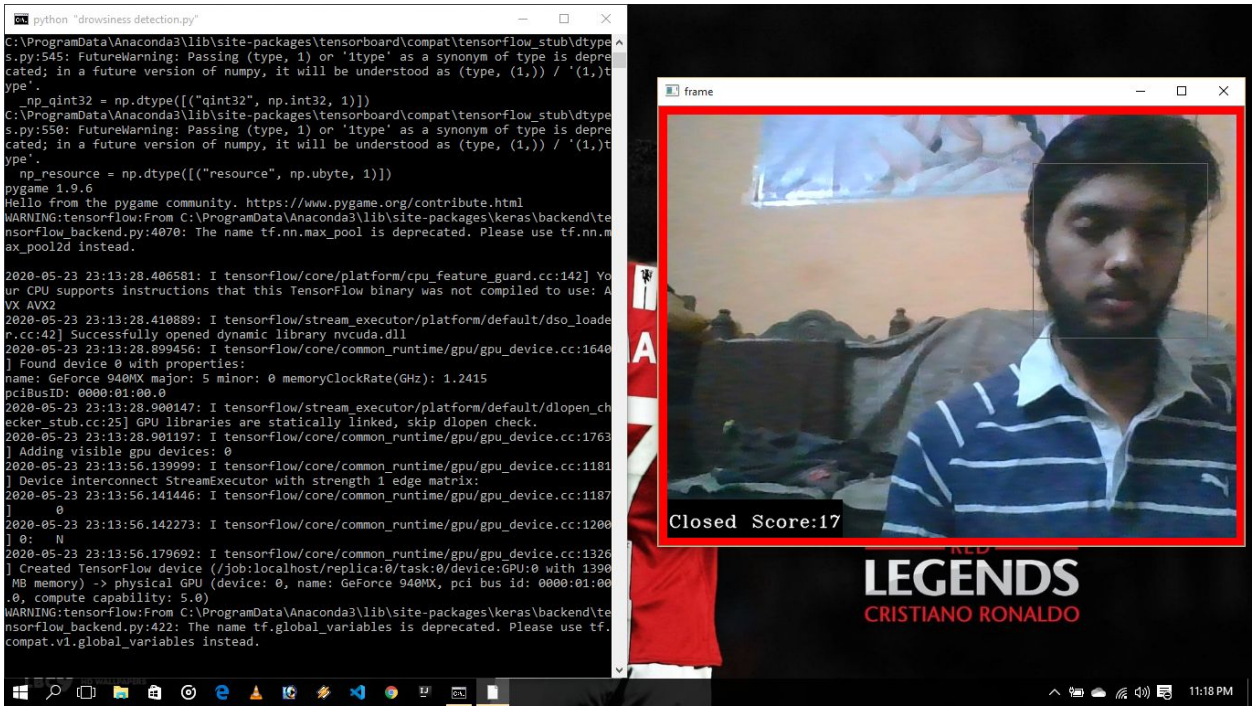


Fig5.11: Output Screen1: Eye Close (Left Position)

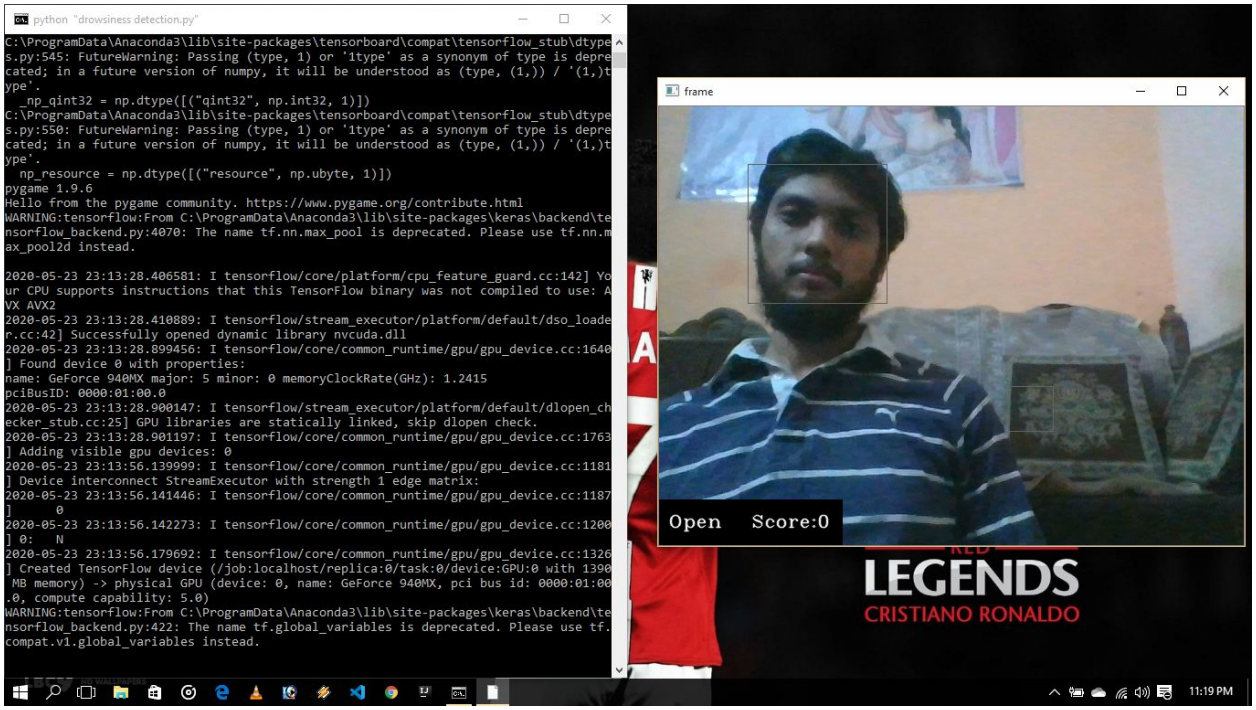


Fig5.12: Output Screen1: Eye Open (Right Position)

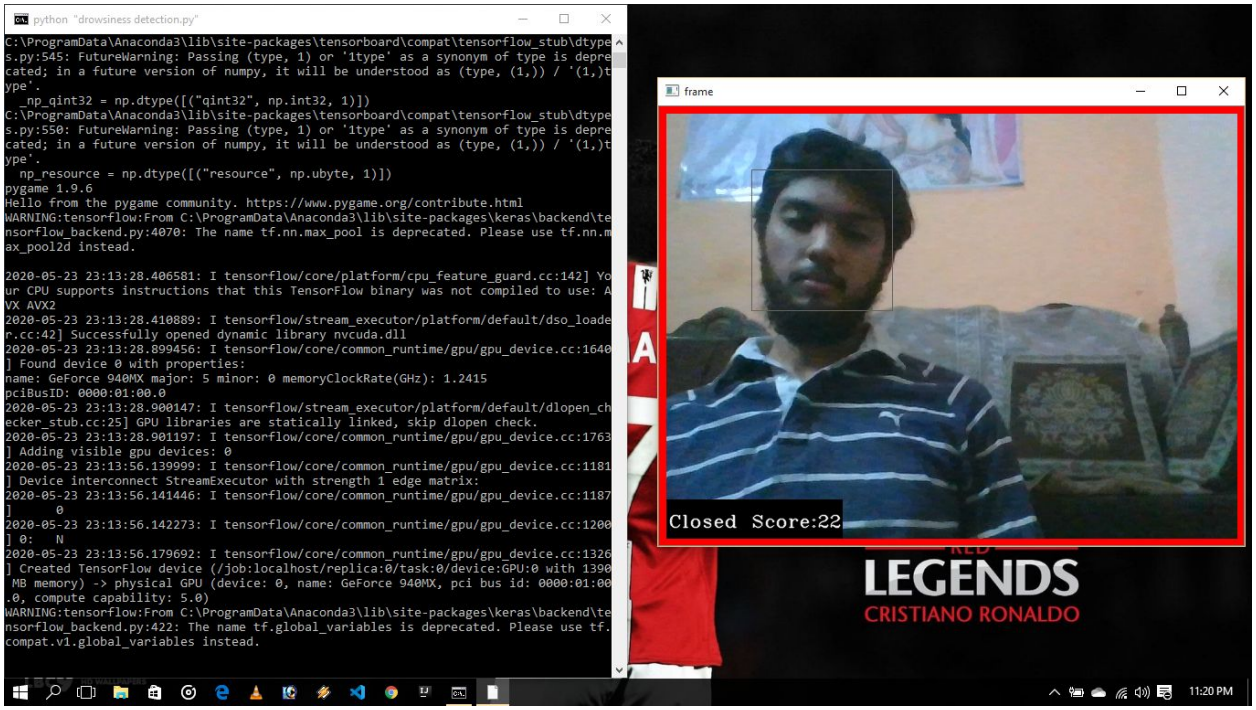


Fig5.13: Output Screen1: Eye Close (Right Position)

5.3. Conclusion

The primary goal of this project is to develop a real-time drowsiness monitoring system in automobiles. We developed a simple system consisting of 5 modules namely(a) video acquisition, (b) dividing into frames, (c) face detection, (d) eye detection, and (e)drowsiness detection. Each of these components can be implemented independently thus providing a way to structure them based on the requirements.

5.4. Further Enhancement/Recommendations

The system at this stage is a “Proof of Concept” for a much substantial endeavor. This will serve as the first step towards a distinguished technology that can bring about an evolution aimed at ace development. The developed system has a special emphasis on real-time monitoring with flexibility, adaptability, and enhancements as the foremost requirements.

Future enhancements are always meant to be items that require more planning, budget, and staffing to have them implemented. There following are a couple of recommended areas for future enhancements:

- **Standalone product:** It can be implemented as a standalone product, which can be installed in an automobile for monitoring the automobile driver.
- **Smartphone application:** It can be implemented as a smartphone application, which can be installed on smartphones. And the automobile driver can start the application after placing it at a position where the camera is focused on the driver.

REFERENCE/BIBLIOGRAPHY

- Driver drowsiness detection using face expression recognition by Mohammad Amin Assari & Mohammad Rahmati :- https://ieeexplore.ieee.org/document/6144162
- Drowsiness detection with OpenCV by Adrain RoseBrock :- https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/
- Intermediate Python Project – Driver Drowsiness Detection System with OpenCV & Keras by DataFlair Team :- https://data-flair.training/blogs/python-project-driver-drowsiness-detection-system/
- Drowsiness Detection with Machine Learning by Grant Zhong :- https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a
- On-Line Automatic Detection of Driver Drowsiness Using a Single Electroencephalographic Channel by Antoine Picot, Alice Caplier, and Sylvie Charbonnier :- https://www.researchgate.net/publication/23932724_On-Line_Automatic_Detection_of_Driver_Drowsiness_Using_a_Single_Electroencephalographic_Channel