COLLEGE OF APPLIED SCIENCE

(Managed by IHRD & Affiliated to University of Calicut)

VADAKKENCHERRY, PALAKKAD-678683

PROJECT REPORT

ON

DRIVER DROWSINESS DETECTION

A project report submitted in partial fulfillment of the requirement for the

Degree in

BACHELOR IN COMPUTER SCIENCE

SIMNA. M (IDATSCS011)

Sixth Semester BSc 2019-2022

Under the guidance of

Ms. SHIMA

CERTIFICATE

This is to certify that the project report submitted by SIMNA. M (IDATSCS011) to University Of Calicut for the award of Degree of Bachelor Science (B.Sc.) in Computer Science is a Bonafide record of the project work carried out by his/her under my supervision and guidance. The content of the report, in full or parts have not been submitted to any other institute or university for the award of any other degree or diploma

Signature
MS. Shima
Assistant Professor

Signature
Mr. Udayakumar
Head of the Department

Computer Science Department

College Of Applied Science

Vadakkencherry

Place:

Date:

(Office seal)

Certified that the candidate was examined by us in the Project Viva Voice Examination held on…… and his/her Register Number is……..……

Examiners:

1.

2.

CSEED
Centre for
Skill Enhancement &
Employability Development

An affiliate of National Council for Technology and Training -
An autonomous organization recognized by Govt of India - NCT

CERTIFICATE

This is to certify that the below mentioned student from dept. of **B.Sc. Computer Science**, **College of Applied Science, Vadakkencherry** has successfully completed the **Project Training** organized by **Centre for Skill Enhancement & Employability Development (CSEED)** for the academic year 2021 - 2022. This Hands-on Training Program Concentrated on **'Machine Learning'** based on the Project topic '**Driver Drowsiness Detection'**. As per our measurement, the candidate has been excellent during the training period.

Name	**Register Number**
SIMNA M	**IDATSCS011**

Centre for Skill Enhancement and Employability Development

MIDHUN LAL K B
PROJECT ENGINEER
CSEED

Evalin Jose 9/4/22

EVALIN JOSE
PROJECT MANAGER
CSEED

HO: Puthenpurakkal Road, Nr. SN Junction, Palarivattom, Cochin, PIN- 682025 | Br: Buhari Tower, MG Road, Nr. West Fort, Thrissur, PIN- 680004
Web: www.cseed.org | Email: mail@cseed.org | cseedkerala@gmail.com

ACKNOWLEDGEMENT

I solemnly thank all the helping hands that made me accomplish this project. First and foremost, I thank the Almighty who is the source of knowledge and my parents for the successful completion of my project work.

I express my gratitude to our Principal Dr Aisha for giving us the facility for this project.

I express my extreme gratitude to Head of the Department Mr. Udayakumar (Senior Lecturer, Department of Computer Science) for providing the relevant details, encouragement and co-operation, valuable suggestion and guidance throughout the project that helped the system successfully.

I am grateful to my project guide Ms.Shima (Asst. Prof, Department of Computer Science) for assisting us with whole hearted guidance and valuable information.

I am eternally grateful to my parents for their constant support without which I would not been what I am today. And last but not the least thank our co- student friends who have contributed a lot of support to me in order to bring out this project in time.

Date: SIMNA. M (IDATSCS011)

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person or material which has been accepted for the award of any degree or diploma of the university or other institute or higher learning, except where due acknowledgement has been made in text.

Place: SIMNA. M (IDATSCS011)

Date:

CONTENTS

ABSTRACT

The advancement of computing technology over the years has provided assistance to drivers mainly in the form of intelligent vehicle systems. Driver fatigue is a significant factor in a large number of vehicle accidents. Thus, driver drowsiness detection has been considereda major potential area so as to prevent a huge number of sleep induced road accidents. This paper proposes a vision based intelligent systemto detect driver drowsiness.

Driver's fatigue is one of the major causes of traffic accidents, particularly for drivers of large vehicles (such as buses and heavy trucks) due to prolonged driving periods and boredom in working conditions. In this paper, we propose a vision-based fatigue detection system for driver monitoring, which is easy and flexible for deployment in buses and large vehicles. The system consists of modules of face detection, eye detection, eye openness estimation, eyelid closure estimation, and fatigue level classification. The main techniques involved are as follows:

1) An approach to estimate the eyelid condition (whether it is open or close)

2) A continuous measure of eye closure and the driver states are classified on it.

3) A warning message for public security

CHAPTER 1

INTRODUCTION

INTRODUCTION

Driver fatigue is a significant factor in a large number of vehicle accidents. Fatalities have occurred as a result of car accidents related to driver inattention, such as distraction, fatigue, and lack of sleep.

Autonomous systems designed to analyze driver exhaustion and detect driver drowsiness can be an integral part of the future intelligent vehicle so as to prevent accidents caused by sleep. A variety of techniques have been employed for vehicle driver fatigue and exhaustion detection. Driver operation and vehicle behavior can be implemented by monitoring the steering wheel movement, accelerator or brake patterns, vehicle speed, lateral acceleration, and lateral displacement. These arenon-intrusive ways of driver drowsiness detection, but are limited to the type of vehicle and driver conditions. Another set of techniques focuses on monitoring physiological characteristics of the driver such as heart rate, pulse rate, and Electroencephalography (EEG) . Research in these lines have suggested that as the alertness level decreases EEG power of the alpha and theta bands increase, hence providing indicators of drowsiness. Although the use of these physiological signals yields better detection accuracy, these are not accepted widely because of less practicality. A third set of techniques is based on computer vision systems which can recognize the facial appearance changes occurring during drowsiness .Physiological feature–based approaches are intrusive because the measuring equipment must be attached to the driver. Thus, visual feature–based approaches have recently become preferred because of their non-intrusive nature. In this paper, we propose a scheme based on extraction of eye features from the data without human intervention. These visual features have been learnt using a machine learning algorithm (SVM).

1.1 PROBLEM DEFENITION

Driver fatigue is the leading cause of a significant number of road accidents. Recent figures suggest that annually there are around 1,300 deaths and 75,000 injuries due to overtiredness. Trying to develop technologies to identify or avoid drowsiness is the most crucial problem in accident prevention systems. Since the danger that drowsiness poses on the road,techniques to counteract its effects need to be created. Driver negligence may result from driving with a lack of attentiveness or due to driver sleepiness and diversion. Driver interruption occurs when any activity or an object distracts the driver's attention away from the duty of driving. Unlike driver fatigue, driver drowsiness does not require any triggering

incident but is instead characterized by the gradual withdrawal of focus from road and traffic demand.

Nevertheless, both driver sleepiness and distraction might have the same consequences, i.e., more reaction time diminished driving efficiency and raised risk of involvement in the accident. If the Fatigue level of the driver and the estimation of drowsiness are determined, then the output is sent to the detection system, and the warning will be triggered.

1.2 OBJECTIVES :

- Driver drowsiness detection is a car safety technology which helps to save the life of the driver by preventing accidents when the driver is getting drowsy.
- The main objective is to first design a system to detect driver's drowsiness by continuously monitoring retina of the eye.
- The system works in spite of driver wearing spectacles and in various lighting conditions.
- To alert the driver on the detection of drowsiness by using buzzer or alarm.
- To alert the driver by using a vibrator under the seat or near the seat
- Traffic management can be maintained by reducing the accidents.

1.3 SCOPE:

The proposed system is used to avoid various road accidents caused by drowsy driving.This system is used for security purpose of a driver to caution the driver if any fire accident or any gas leakage occur.Drowsy driver detection methods can form the basis of a system to potentially reduce the number of crashes related to drowsy driving.In future we can implement drowsiness detection system in aircraft in order to alert pilot.The alcoholic sensor is also used for drunk drivers.In the near future, the system can be modified for further enhancement which are mentioned below:

- The system will include two modes; day light And night mode. These modes will be Automatically set by the use of a specific sensors, depending on the light available.
- After drowsiness has been detected the vehicle Will be set to self-driving mode and will be Controlled using CNN.
- The system will be Developed using Open CV to avoid delay in Processing

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The paper proposes a fatigue driving detection algorithm based on facial multi feature fusion combining driver characteristics. First, we introduce an improved YOLOv3-tiny convolutional neural network to capture the facial regions under complex driving conditions, eliminating the inaccuracy and affections caused by artificial feature extraction. Second, on the basis of the Dlib toolkit. We introduce the Eye Feature Vector(EFV) and Mouth Feature Vector(MFV), which are the evaluation parameters of the driver's eye state and mouth state, respectively. Then, the driver identity information library is constructed by offline training, including driver eye state classifier library, driver mouth state classifier library, and driver biometric library. Finally, we construct the driver identity verification model and the driver fatigue assessment model by online assessment. After passing the identity verification, calculate the driver's closed eyes time, blink frequency and yawn frequency to evaluate the driver's fatigue state.

2.2 DISADVANTAGES

- If the person is in deep sleep conditionor drowsy, he will not wake up only with the alarm message.
- It will be more complex.
- When we detect the drowsines by their yawn there might be a chance for an error. Because if the person is laughing the system may detect it as yawn so the prediction may go wrong.

2.3 PROPOSED SYSTEM

In the proposed work, pre-existing features for facial landmark detection is implemented to identify the state or drowsiness and fatigue.facial landmark predefined landmark helps in shape prediction to clearly identify the various regions of the face like eye brows, eye, mouth region etc

High vision camera's are embedded to monitor and capture to extract frames one by one and generate the alerts accordingly. Each extracted frame is analyzed at time to study the pattern of facial features; using Haar Cascade Classifiers and determined Eye Aspect Ratio(EAR) and Mouth Aspect Ratio (MAR) for each frame EAR and MAR values exceed their respective threshold values, a blink and a yawn is considered respectively. The system

alerts the driver by playing an alarm if eye blinking rate and yawns are suspected for a certain number of consecutive frames. The alarm is activated to grab the driver's attention and keep on ringing until driver wakes up.If the driver didn't wakeup even after the the alarm started ringing..we provide a vibrator motor to alert the drowsy person .so it can easily reduce the drowsines

2.4 ADVANTAGES

- Easy adaptable system.
- Vibrator motor is used as an advanced feature, so it will be more effective to wake up a drowsy person.

2.5 FEASIBILITY STUDY

Feasibility analysis is an analysis of possible alternative solution to a problem and a recommendation on the best alternative. Before the request is to be approved, it has to be checkedwhether the system is feasible or not with respect to the following areas. It is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. The objective of feasibility study is not to solve the problem but to acquire a sense of its scope.

Feasibility analyses involve 8 steps:

1. From a project team and appoint a project leader.

2. Prepare system flow charts.

3. Enumerate potential candidate systems.

4. Describe and identify characteristics of candidate systems.

5. Determine and evaluate performance and cost effectiveness of each candidate system.

6. Weigh system performance and cost data.

7. Select the best candidate system.

8. Repair and report final project directive to management.

Three key considerations are involved in the feasibility analysis:

- Technical
- Economical
- Operational.

TECHNICAL FEASIBILITY

The technical issue that is usually raised during the feasibility stage of the investigation includes the following:

Accuracy, reliability, ease of maintenance is considered. Exist to do what is suggested. Technical capacity to hold the data required to use the new system. Adequate response to inquiries regardless of the number of locations of users. Data security is measured.

This study is performed to check whether the proposed system is technically feasible or not. Any system Developed must not have a high demand on the available technical resources. This will lead to high demand on the Available technical resources. This will lead to high demand being placed on the client. Technical feasibility centers On the existing computing system (hardware, software etc.) and to what extent it can support the proposed addition. This is technically feasible, all requirements used for this software are reachable.

ECONOMICAL FEASIBILITY :

This study is the most frequently used method for evaluating the effectiveness of a candidate system more commonly known as cost /benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. This analysis phase determines how much cost is needed to produce the proposed system. This system is economically feasible since it does not require any initial set up cost, as the organization has required machines and supporting programs for the application to execute itself. No extra charges are needed for this project. Economic analysis used a method for evaluating the effectiveness of the considered system. The cost benefit analysis is done, a procedure to determine.

The benefit and saving those are expected from a candidate system and compare them with cost.The selected candidate system is economically feasible. Proposed system is developed with the available resources. Since cost input for the software is almost nil, output of the software is always a profit. Hence this proposed system, the risa number of employees to be evolved is reduced drastically. So the proposed system is said to be feasible.

OPERATIONAL FEASIBILITY :

This is performed to check whether the system is operationally feasible or not. Using command button throughout the application programs enhanced operational feasibility. So maintenance and modification is found to be easier. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. A network admin can himself do all things in this system

Proposed projects are beneficial only if they can be turned into an information system which will meet the organization's operating requirements. Operation feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of the project. There is sufficient support for the management from the users. Will the system be used and work properly if it is being developed and implemented.

CHAPTER 3

SYSTEM SPECIFICATIONS

3.1 HARDWARE REQUIREMENTS:

- Processor: Intel core i3 or above
- RAM: 8GB
- USB Cable
- Microcontroller:Arduino Uno :Atmega 328p
- Vibration Motor
- Buzzer

3.2 SOFTWARE REQUIREMENTS:

- Operating System: Windows
- Python
- Anaconda Navigator
- Spyder IDE

3.3 SOFTWARE DESCRIPTION

PYTHON

Python is an interpreted high-level general-purpose programming language. Python's design Philosophy emphasizes code readability with its notable use of significant indentation. Its language Constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and Large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, Including structured (particularly, procedural), object-oriented and functional programming. Python is often Described as a "batteries included" language due to its comprehensive standard library.

Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very Attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing Components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of Program maintenance. Python supports modules and packages, which encourages program modularity and code

reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all Major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is No compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad Input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows Inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through The code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective Power. On the other hand, often the quickest way to debug a program is to add a few print statements to the Source: the fast edit-test-debug cycle makes this simple approach very effective.

CHAPTER IV

SYSTEM IMPLEMENTATION

4.1DATA FLOW DIAGRAM

Data flow diagrams are used widely for modeling the requirements. DFD's show the flow of data through a system. The system may be a company, an organizational set of procedures, a computer hardware system, a software system or any combination of the proceedings.

The DFD also known as a data flow graph or a bubble chart. The following observations about DFDs are important:

1. All names should be unique. This makes easier to refer items in the DFD. Remember that a DFD is not a flow chart.

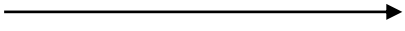
2. Arrows in a flow chart represent the order of events; arrows in DFD represent flowing data. A DFD doesn't imply any order of events.

3. Suppress logical decisions (A diamond shape box is used in flow chart to represent decision points with multiple exit paths of which only one is taken). This implies an ordering of events, which makes no sense in DFD.

BASIC DFD SYMBOLS

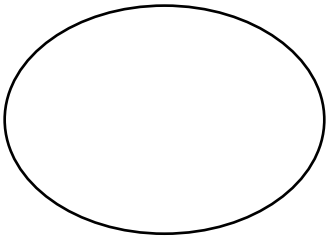
1. ARROW

A data flow is a route, which enables packets of data to travel from one point to another. An arrow indicates data flow-data in motion. It is a pipeline through which information flows.



2. PROCESS

A process represents transformations where incoming data flows are changed into outgoing data flows. A “circle” or a “bubble” represents a process that transforms incoming data flows into outgoing data flows



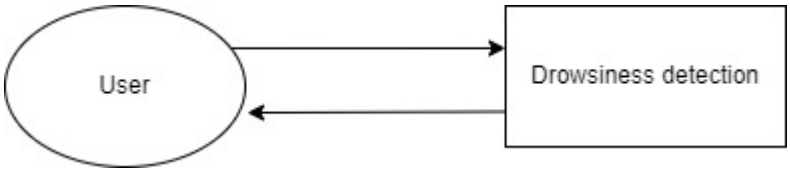
3. DATA STORE

A data store represents a repository of data that is to be stored for use by one or more processes may be symbol as buffer or queue or sophisticated as relational database. They should have clear names

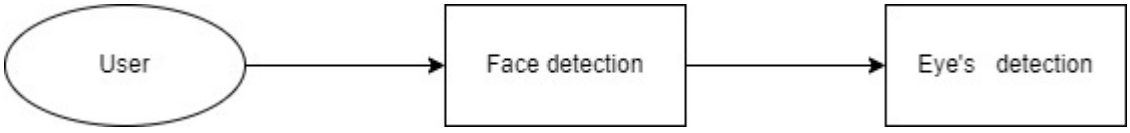
4. SOURCE

A source or sink is a person or part of an organization which enter or receives information from the system but it is considered to be outside the context of data flow models.

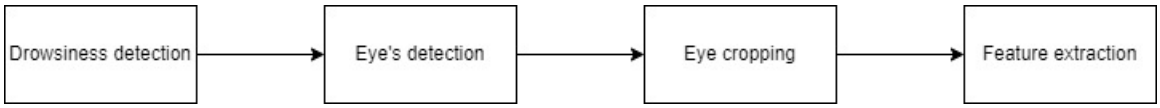
LEVEL 0



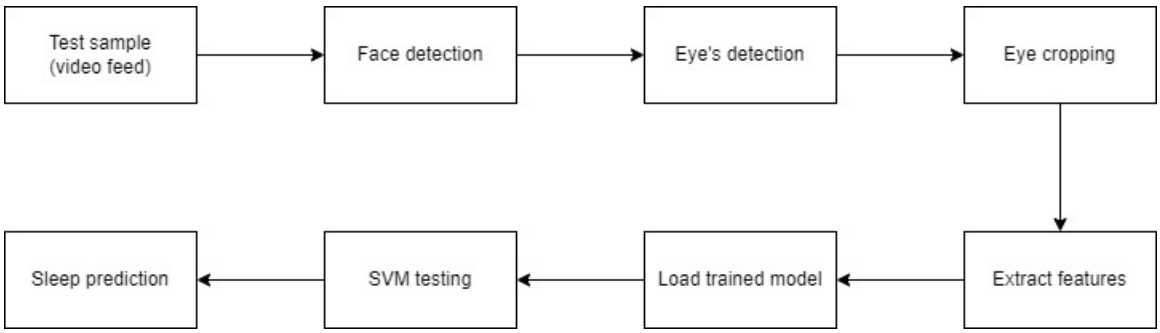
LEVEL 1



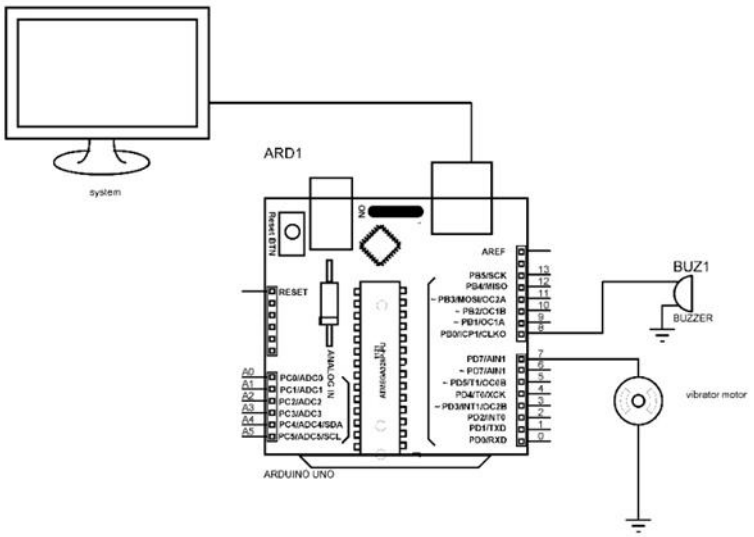
LEVEL 2



LEVEL 3



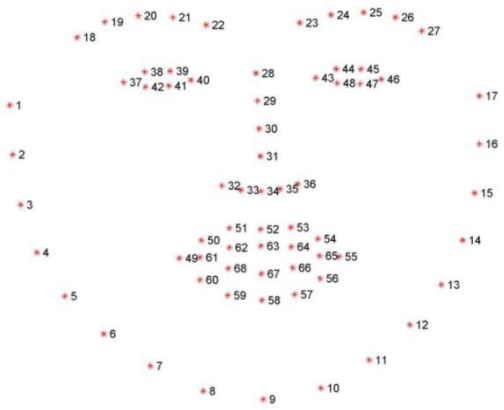
4.2 CIRCUIT DIAGRAM



4.3DATASET

shape_predictor_68_face_landmarks model is used for detecting rapid facial points. It is a pre-trained model

dlib library is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face

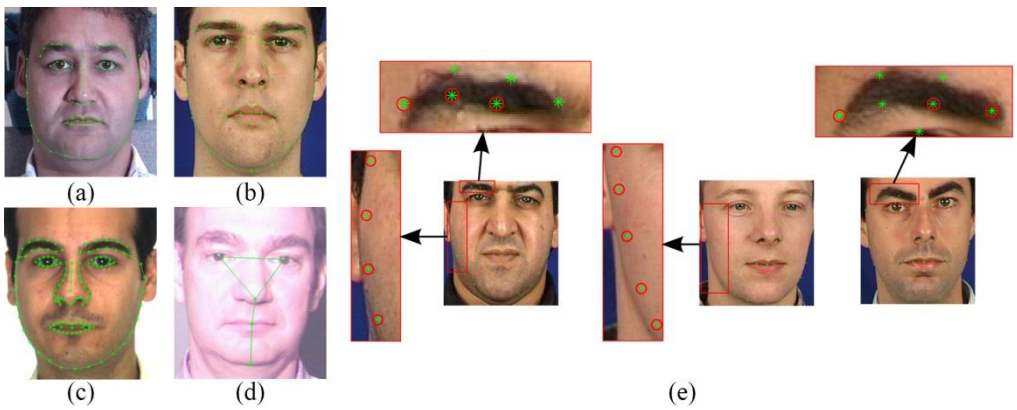


It's a landmark's facial detector

Data set

iBUG300-W dataset was used to train This model

300 Indoor and 300 Outdoor images with different faces, different face size and different skin colored and in different lighting condition and it also consist of 399 unique images



4.4 MODULE DESCRIPTION

- **Capturing Camera feed and Face detection**
- **Detecting facial points and calculating EAR value**
- **Hardware Motor and Buzzer**

Module1: Capturing camera feed and face detection

Capturing camera feed is done using python and open CV. Face detection is done by using Harcascadeclassifier.it is a model works with machine learning.

Module2: Detecting facial points and calculating EAR value

From face we detect our necessary facial points e.g.(eye facial points) using facial land mark 68.The points are detected to calculate EAR value(Eye Aspect Ratio).A threshold is set based on EAR value. Drowsiness is predicted and a signal is send to hardware.

Module3: Hardware Motor and Buzzer

In hardware there is a motor and buzzer and an Arduino. Whenever an incoming signal occurs, first it start buzzer and wait for the driver to wake up. If the driver doesn't wakeup then it activates the vibration motor.

CHAPTER V

SYSTEM TESTING

..

SYSTEM TESTING

Testing is vital to the success of the system. System testing makes a logical assumption that if all the Parts of the system are correct, the goal will be successfully achieved. System testing is the stage of Implementation that we aimed at assuring that the system works accurately and efficiently before live operation commences.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of developed system using various kinds of data

The candidate system is subject to a variety of tests. In the response security and usability is tested. A series of testing is performed for the proposed system, before the system is ready for user acceptance Testing.

5.1 UNIT TESTING

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as module testing. The modules are tested separately. These tests are carried out during the programming stage itself. In this testing step, each module is found to be working satisfactory as regards to the expected outputs from the module. A unit is the smallest testable part of an application.

In this project Drowsiness detection detection system each module is separately tested. Enter input data to each form and test working of data and observe the output data. All decision branches and internal code flow should be validated in the project. Conform the properworking of each and every unit in my project as the part of unit testing.

5.2 INTEGRATION TESTING

Integration testing is a systematic testing for constructing the program structure while at the same time conducting tests to uncover errors associated with the interfaces; all the modules are combined and tested as a whole. When conducting this sort of tests, all analysts may devise the test data, which is the collection of data that analysts used when providing system accuracy. Another source of test data is the user.

Integration testing in project Driver drowsiness detection, individual modules are combined and tested in a group. Enter input data and test the output data of input. This testing done in a project at the time of database related operation occurred.

5.3 SYSTEM TESTING

System testing is the software or hardware testing conducted on a complete,integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing,and as such, should require no knowledge of the inner design of the code or logic.

System testing ensures that the entire integrated Driver drowsiness detection System project software system meets requirements. It tests a configuration to ensure known and predictable results.

5.4 VALIDATION TESTING

Validation testing is where requirements established as a part of software requirements analysis are Validated against the software that has been constructed. This test provides the final assurance that the Software meets all functional, behavioral and performance requirements.

CHAPTER VI

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The existing driver drowsiness detection system does not have a vibrator motor, if the person is in deep sleep condition or drowsy, he will not wake up only with the alarm message. Which is overcome by this system,using such a system we can wake up the person in addition to the alarm message.Machine learning allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so.Our project aims to reduce accidents and provide safe life to driver's and vehicles safety.A systemfor vehicle and driver safety is present only luxurious vehicles.Using drowsiness detection system, driver safety can be implemented in normal vehicles also.

6.2 FUTURE WORK

The project driver drowsiness detection system can be implemented in vehicles where we witness thousands of accidents due to drowsy driving. This project will be very useful in reducing the number of accidents in our daily life.The system has Been designed at the maximum possible excellence.Proper documentation of the code helps in easy reuse or modifications of the code. This project can be further developed in future.

- Development of a hybrid microcontroller for a vehicle which also consists of an alcohol detector which will sense if the driver is drunk and would not start theVehicle.
- If the person who is driving takes alcohol then the vehicle will be stopped immediately by giving alarms.
- Integrate the alarm into the car's stereo system
- Create a fully wireless system (with a wireless bracelet)
- The system will be developed using openCV to avoid delay in processing

APPENDIX A

SOURCE CODE

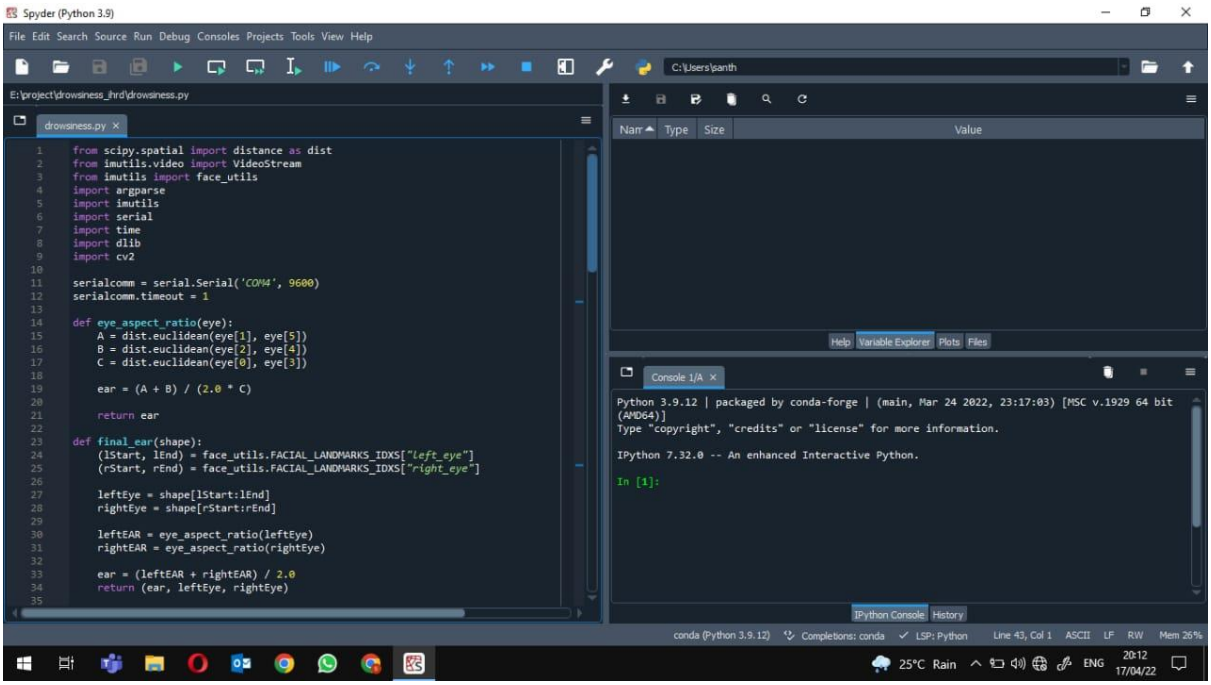
```
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
import argparse
import imutils
import serial
import time
import dlib
import cv2

serialcomm = serial.Serial('COM25', 9600)
serialcomm.timeout = 1

def eye_aspect_ratio(eye):
A = dist.euclidean(eye[1], eye[5])
B = dist.euclidean(eye[2], eye[4])
C = dist.euclidean(eye[0], eye[3])

ear = (A + B) / (2.0 * C)

return ear

def final_ear(shape):
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

 leftEye = shape[lStart:lEnd]
 rightEye = shape[rStart:rEnd]

 leftEAR = eye_aspect_ratio(leftEye)
 rightEAR = eye_aspect_ratio(rightEye)

 ear = (leftEAR + rightEAR) / 2.0
 return (ear, leftEye, rightEye)



#ap = argparse.ArgumentParser()
#ap.add_argument("-w", "--webcam", type=int, default=1,
          #help="index of webcam on system")
#args = vars(ap.parse_args())

EYE_AR_THRESH = 0.33
EYE_AR_CONSEC_FRAMES = 30
COUNTER = 0

print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
```

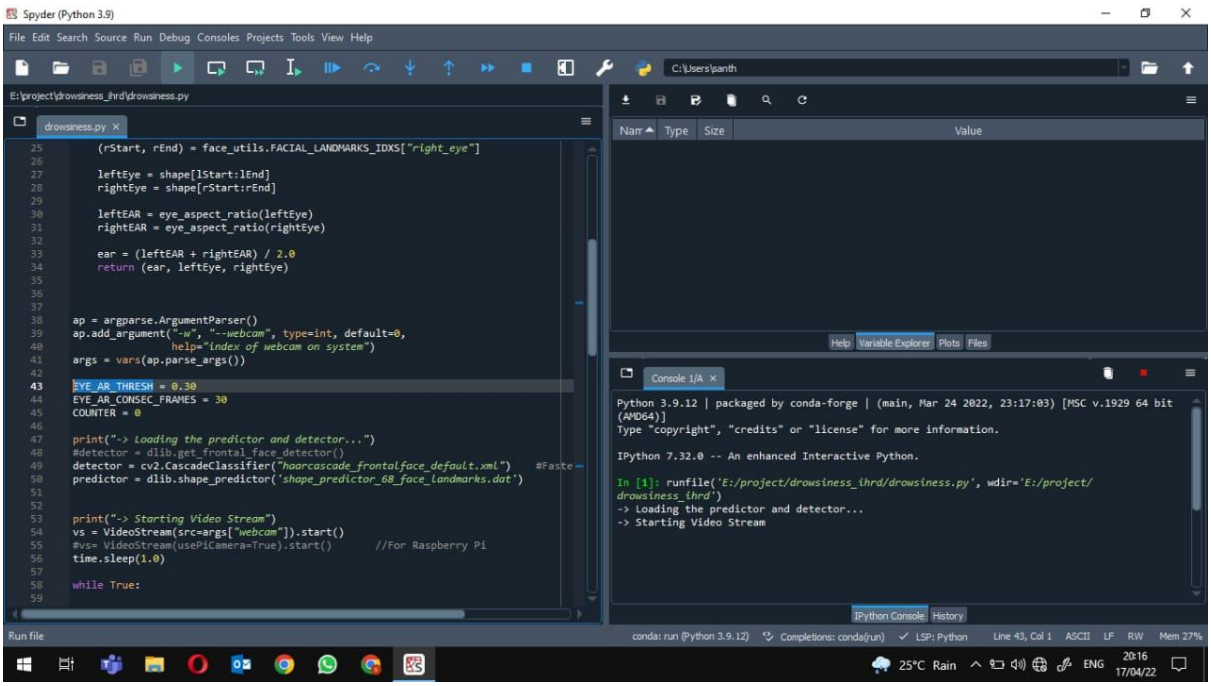
```
detector =
cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
#Faster but less accurate

predictor =
dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")


print("-> Starting Video Stream")
#vs = VideoStream(src=args["webcam"]).start()
vs = cv2.VideoCapture(0)
#vs= VideoStream(usePiCamera=True).start()     //For Raspberry Pi
time.sleep(1.0)

while True:

frame = vs.read()
frame = imutils.resize(frame, width=450)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

#rects = detector(gray, 0)
 rects = detector.detectMultiScale(gray, scaleFactor=1.1,
 minNeighbors=5, minSize=(30, 30),
 flags=cv2.CASCADE_SCALE_IMAGE)

#for rect in rects:
 for (x, y, w, h) in rects:
  rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))

  shape = predictor(gray, rect)
  shape = face_utils.shape_to_np(shape)

  eye = final_ear(shape)
  ear = eye[0]
  leftEye = eye [1]
  rightEye = eye[2]



  leftEyeHull = cv2.convexHull(leftEye)
  rightEyeHull = cv2.convexHull(rightEye)
  cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
  cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)



  if ear < EYE_AR_THRESH:
      COUNTER += 1

      if COUNTER >= EYE_AR_CONSEC_FRAMES:

          i="on"
          serialcomm.write(i.encode())
          time.sleep(0.6)
          print("alert")
          print(serialcomm.readline().decode('ascii'))
```

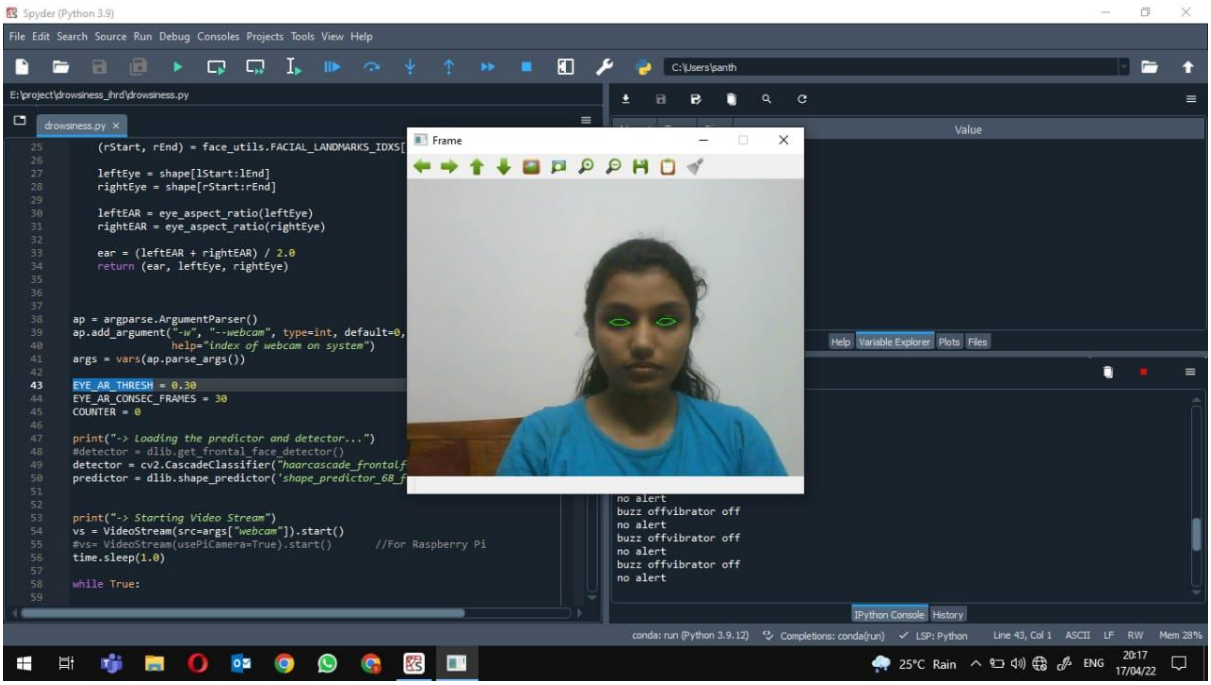
```
        cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


    else:
            COUNTER = 0

            i="off"
            serialcomm.write(i.encode())
            time.sleep(0.6)
            print("no alert")
            print(serialcomm.readline().decode('ascii'))



            cv2.imshow("Frame", frame)
            key = cv2.waitKey(1) & 0xFF

            if key == ord("q"):
            break

            serialcomm.close()
             cv2.destroyAllWindows()
             vs.stop()
```

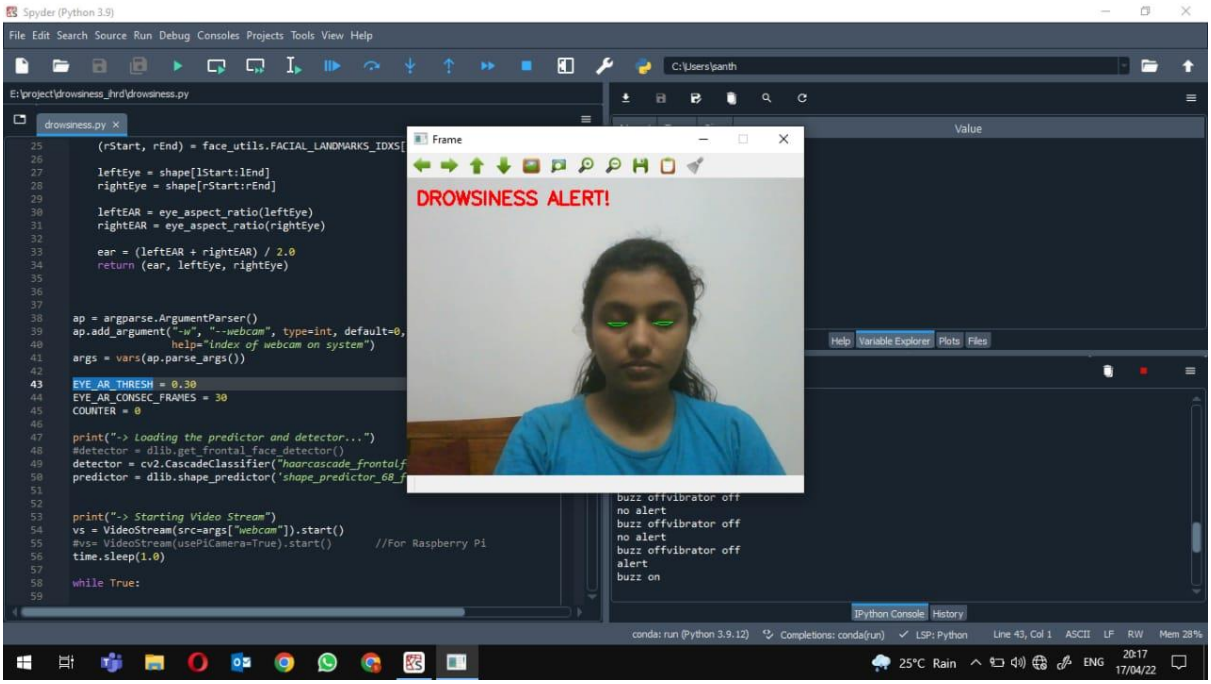
APPENDIX B

SCREENSHOTS






Frame
drowsiness.py
EYE_AR_THRESH = 0.30
EYE_AR_CONSEC_FRAMES = 30
COUNTER = 0
print("-> Loading the predictor and detector...")
print("-> Starting Video Stream")
time.sleep(1.0)
while True:
no alert
buzz offvibrator off
no alert
buzz offvibrator off
no alert
buzz offvibrator off
no alert


Frame
DROWSINESS ALERT!
drowsiness.py
EYE_AR_THRESH = 0.30
EYE_AR_CONSEC_FRAMES = 30
COUNTER = 0
print("-> Loading the predictor and detector...")
print("-> Starting Video Stream")
time.sleep(1.0)
while True:
buzz offvibrator off
no alert
buzz offvibrator off
no alert
buzz offvibrator off
alert
buzz on

BIBLIOGRAPHY

- “Learn Python The Hard Way”-Zed Shaw
- C language reference model, Author: Guido van Rossum, and Fred L. Drake, Jr
- Shape_predictor_68_face_landmarks model
- iBUG300 -W Dataset
- Harcascade model

- https://www.python.org/
- http://www.c_language.org/