A Project Report

on

DROWSINESS DETECTION WITH OPENCV

Submitted in partial fulfilment of the

requirement for the award ofthe degree

of

BTECH CSE WITH SPECIALIZATION IN DA

Under The Supervision of
Ms. Nikita:

Ast. Professor

Submitted By

Bobby Tripathi
19021120008/19SCSE1120010

Harsh Kumar Singh

19021120012/19SCSE1120014

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA
Oct, 2019-2023

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"DROWSINESS DETECTION USING OPENCV"** in partial fulfillment of the requirements for the award of the Btech CSE with specialisation in Data Analytics submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

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

Bobby Tripathi – 19SCSE1120010

Harsh Kumar Singh – 19SCSE1120014

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

Supervisor Name: Ms. Nikita

Designation: Asst. Professor

CERTIFICATE

The Final Project Dissertation Viva-Voce examination of Bobby Tripathi : 19SCSE1120010 Harsh Kumar Singh: 19SCSE11200014 has been held on 22nd December, 2021 and there work is recommended for the award of Btech SCSE with specialisation in data analytics.

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

Signature of Project Coordinator

Signature of Dean

Date: 22nd December, 2021

Place: Greater Noida

ACKNOWLEDGEMENT

There were lots of individual who were certainly very helpful, kind, cooperative and generous along the development of this project. I express my appreciation to those concerned. This project and report would have been not possible if it was not for the kind support, tremendous support and continuous supervision of my learned project guide. The deep sense of sincere appreciation that I owe to my learned guide and supervisor “Ms. Nikita” cannot be expressed in words. Her impossible help, constant support, invaluable supervision, kind nature is among the very few consents that she bestowed upon us time to time from the beginning of this project.

Name
Bobby Tripathi
Harsh Kumar Singh

Abstract

With this python project, we will make Drowsiness Detection System. An incalculable number of individuals drive on the roadway day and night. Cabbies, transport drivers, transporter, and individuals voyaging significant distances experience a lack of sleep. Because of which it turns out to be extremely dangerous to drive when feeling lethargic. Most of the mishaps occur because of the tiredness of the driver. In this way, to prevent these mishaps we are constructing a framework using python, OpenCV, and Keras which will caution the driver when he feels drowsy.

Drowsiness Detection is a security innovation that can prevent mishaps that can be brought about by drivers who dozed off while driving. The goal of this halfway python project is to make a Drowsiness Detection system that will identify that an individual's eyes are shut for a couple of moments. The framework will caution the driver when tiredness is recognized.

List of Figures

Figure name

Table of content

Chapter 1: Introduction

1.1 Introduction

The country's automotive population is growing at an alarming rate. The most serious issue caused by increased traffic is a rise in the number of road accidents. Driver drowsiness, drinking, and recklessness are all major factors in an accident. Taking these elements into consideration, the driver behaviour state is a key problem for creating sophisticated driver assistance systems. Driver drowsiness detection is an automobile safety system that detects and avoids accidents when the driver becomes drowsy. Driver inattention may be the consequence of tiredness and distraction, as well as a lack of attentiveness when driving. In real time, the system informs the motorist through an alarm.

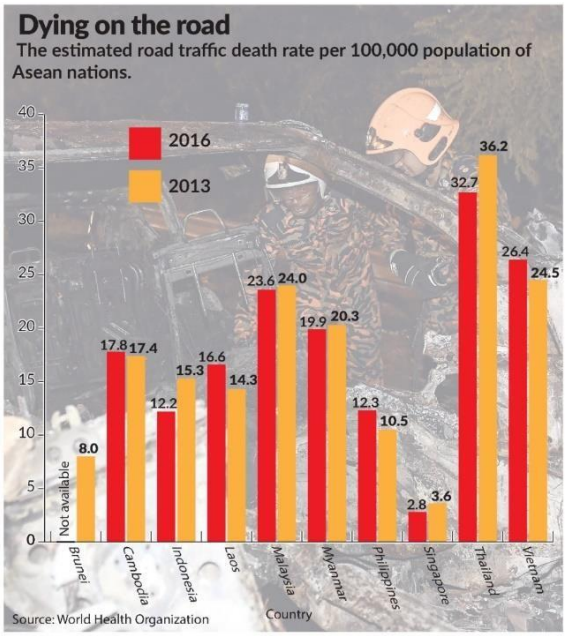

Figure 0: The statistic of road accidents of Asean Nations in 2013 and 2016

Drowsiness is a condition of reduced consciousness induced by a lack of sleep or exhaustion. Drowsiness causes the driver to lose control of the car, which may result in a collision. detour him/her from the road, resulting in serious consequences occurrences WHSTA stands for Women's Health and Safety at Work. According to statistics, tiredness is the leading cause of accidents. driver. However, with increased expansion throughout the years, motorization rate

accompanied by road network India is experiencing major consequences as a result of its rapid growth and urbanization. on the degree of road safety, The total number of roads in India is Accidents have increased by 3% from 4,90,400 in 2014 in 2016 to 5, 01,424 in 2017. The examination of traffic accident data of in 2015, there were around 1374 accidents and 300 fatalities every day on Indian highways. A few of the key reasons for this massive loss include alcohol intake while driving and driver tiredness. Advanced driver assistance techniques can be utilized to reduce a large number of accidents. The driver is monitored in two ways: directly and indirectly. Head movement and facial expressions are collected using sensors such as cameras in the direct monitoring approach. Indirect strategies for assessing tiredness include driver behaviors and responses to certain situations. A sequence of actions made by the driver while driving includes eye activity, the frequency and length of time during which the eyes were closed, and head movement concerning the center of gravity, which aids in determining the driver's present state.

Figure 1: Example of drowsiness condition while driving

This can be an essential safety measure because studies show that accidents caused by tired or sleepy drivers account for roughly 20% of all accidents, and on particular long-distance highways, it can be up to 50%. It is a severe problem, and most individuals who have driven for lengthy periods of time at night can attest to the fact that exhaustion and a temporary state of unconsciousness can occur to everyone and everyone.

There has been an increase in the number of safety devices in automobiles and other vehicles, and many are now required, but none of them can help if a driver falls asleep behind the wheel, even for a little minute. As a result, that is what we are going to construct today - a Driver Drowsiness Detection System.

In this Python project, we will utilize OpenCV for gathering the images from webcam and feed them into a Deep Learning model which will arrange whether the individual's eyes are 'Open' or 'Shut'. The methodology we will use for this Python project is as per the following:

Step 1 – Take picture as contribution from a camera.

Step 2 – Detect the face in the picture and make a Region of Interest (ROI).

Step 3 – Detect the eyes from ROI and feed it to the classifier.

Step 4 – Classifier will sort whether eyes are open or shut.

Step 5 – Calculate score to check whether the individual is sleepy.

1.2 Requirement of the system

1. HARDWARE REQUIRMENTS
    - Camera
    - Personal computer

2. SOFTWARE REQUIREMENTS
    - Python 2.7 or above versions
    - Anaconda software

1.3 PRESENT SYSTEM

The current driver sleepiness detection technology has the following drawbacks. The system primarily employs two cameras, one for tracking head movement and the other for monitoring facial expressions.

Another downside is sensor ageing, since all of these sensors are tied to the driver's body, which may have an effect on the driving. So, in order to solve all of these limitations, we devised a system in which a live camera is utilised to monitor the driver's drowsy status and inform the driver, thereby reducing road accidents.

1.4 PROPOSED METHOD

There are several different algorithms and methods for eye tracking, and monitoring. Most of them in some way relate to features of the eye (typically

reflections from the eye) within a video image of the driver. The original aim of this project was to use the retinal reflection as a means to finding the eyes on the face, and then using the absence of this reflection as a way of detecting when the eyes are closed. Applying this algorithm on consecutive video frames may aid in the calculation of eye closure period. Eye closure period for drowsy drivers are longer than normal blinking. It is also very little longer time could result in severe crash. So we will warn the driver as soon as closed eye is detected.

1.5 ADVANTAGES

- The discovered aberrant behaviour is remedied in real time via alerts.
- The component readily develops interaction with other drivers.
- The driver's life can be spared by informing him via the alarm system.
- The vehicle's speed may be regulated.
- By lowering accidents, traffic management can be maintained.
- Usable in practice

Chapter 2: Literature Review/Project Design

2.1 Literature Review

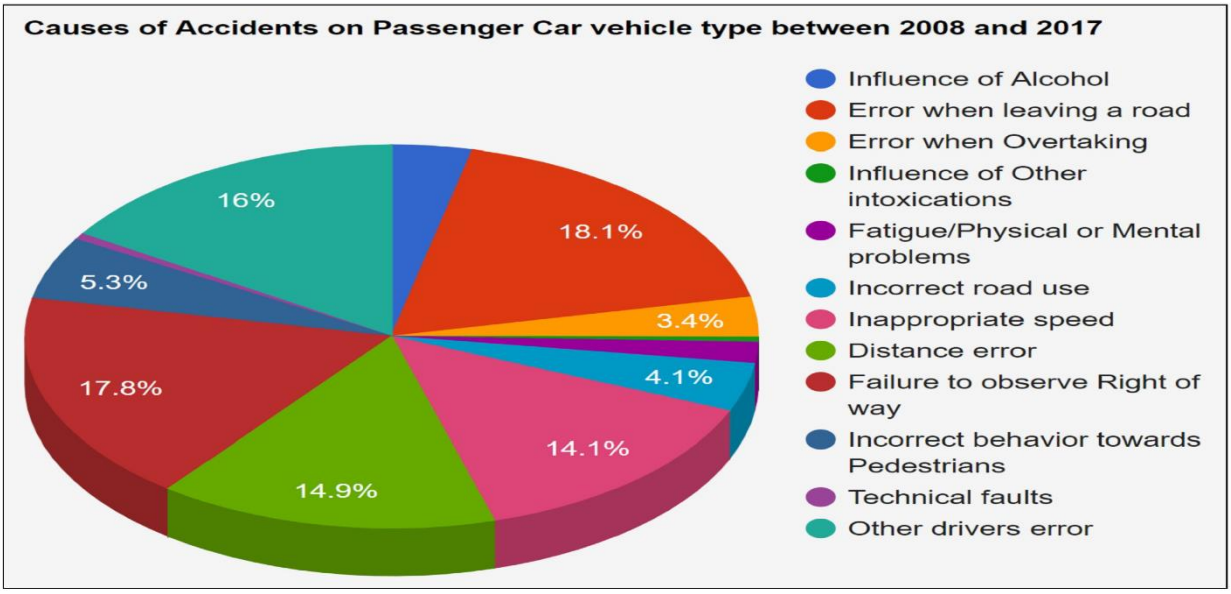

Figure 2: Statistics of Car Accidents

In today's world accident are one of the serious concerns, and, one of the main reasons behind it is tiredness which is caused by travelling for a long distance without any rest, stress and many more. These accidents don't only affect the driver but also their family. According to PubMed.gov there was an article published that was "Road accidents caused by drivers falling asleep" in that about 29600 Norwegian accident that involves driver were sent questionnaire in which they were asked about the last accident reported to their insurance company. About 9200 drivers (31%) returned the questionnaire. The questionnaire which was sent to them contains the question regarding sleep or fatigue as contributing factors to the accident. In addition, the drivers reported whether or not they had fallen asleep some time whilst driving. and what the consequences had been. Sleep or drowsiness was a contributing factor in 3.9% of all accidents, as reported by drivers who were at fault for the accident. This factor was strongly over-represented in night-time accidents (18.6%), in running-off-the-road accidents (8.3%), accidents after driving more than 150 km on one trip (8.1%), and personal injury accidents (7.3%). A logistic regression analysis showed that the following

additional factors made significant and independent contributions to increasing the odds of sleep involvement in an accident. A total of 10% of male drivers and 4% of females reported to have fallen asleep while driving during the last 12 months.

To reduce this accident to a very extent we introduce this python project which is “DROWSINESS DETECTION With OpenCV”. This project will alert the driver when the system detects drowsiness. It is using different python packages that are OpenCV, keras that will alert the driver when he feels sleepy.

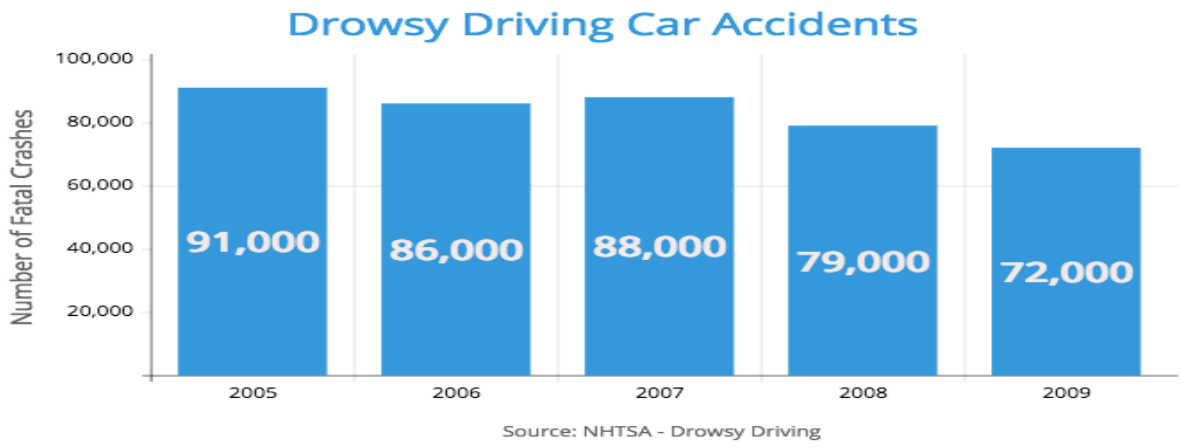

Figure 3: Bar Graph of Drowsiness Driving Car Accidents

2.1 Block Diagram

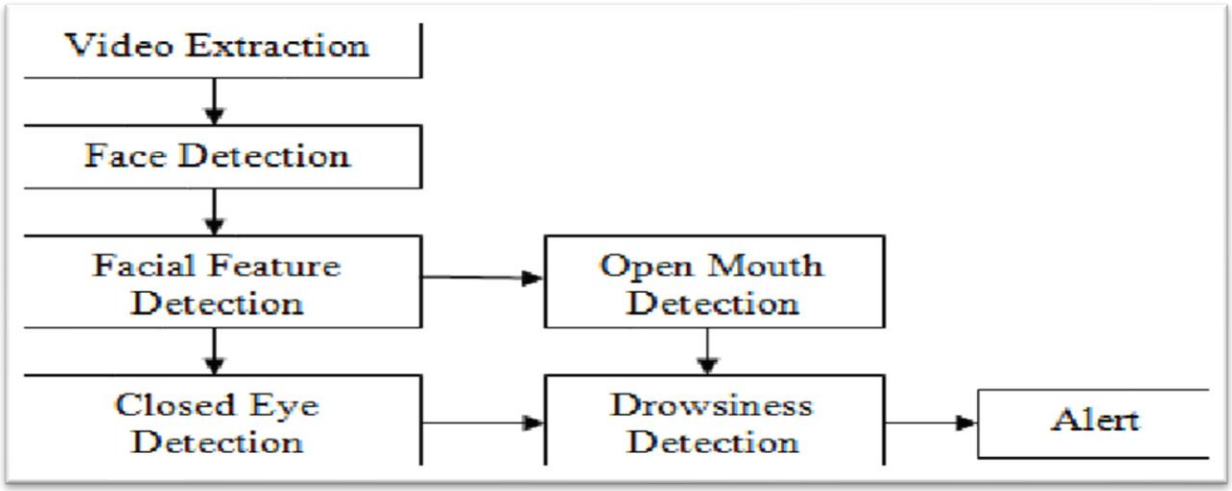

Figure 4: Block Diagram of drowsiness detection

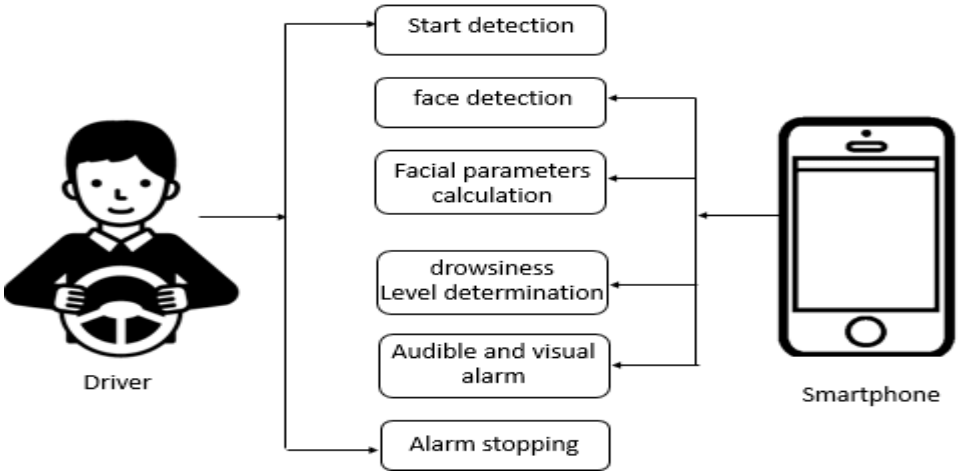

Figure 5: Block Diagram of drowsiness detection

2.2 Driver Drowsiness Detection Dataset

The dataset utilized for this model is taken from the Kaggle. The dataset, we have taken from Kaggle is composed a content that catches eyes from a camera and stores in the disk. It has separated the datasets in different labels of 'Open' or 'Shut'. It was physically cleaned by eliminating the undesirable pictures which were excessive for building the model. The information involves around 7000 pictures of individuals' eyes under various lighting conditions. After training the model dataset, it has attached the final weight and model architecture file "models/cnnCat2.h5".

Presently, we can utilize this model to group in case an individual's eye is open or shut.

Chapter 3: Functionality/Working of Project

3.1 The Model Architecture

The model we utilized is worked with Keras utilizing Convolutional Neural Networks (CNN). A convolutional neural network is an extraordinary kind of profound neural network which performs very well for image classification purposes. A CNN essentially comprises of an input layer, an output layer and a hidden layer which can have multiple layers. A convolution activity is performed on these layers utilizing a channel that performs 2D matrix multiplication on the layer and filter.

The CNN model engineering comprises of the accompanying layers:

Convolutional layer; 32 nodes, kernel size 3

Convolutional layer; 32 nodes, kernel size 3

Convolutional layer; 64 nodes, kernel size 3

Completely associated layer; 128 nodes

The last layer is likewise a completely associated layer with 2 nodes. A Relu (Rectified Linear Unit) initiation work is utilized in every one of the layers aside from the yield layer in which we utilized SoftMax.

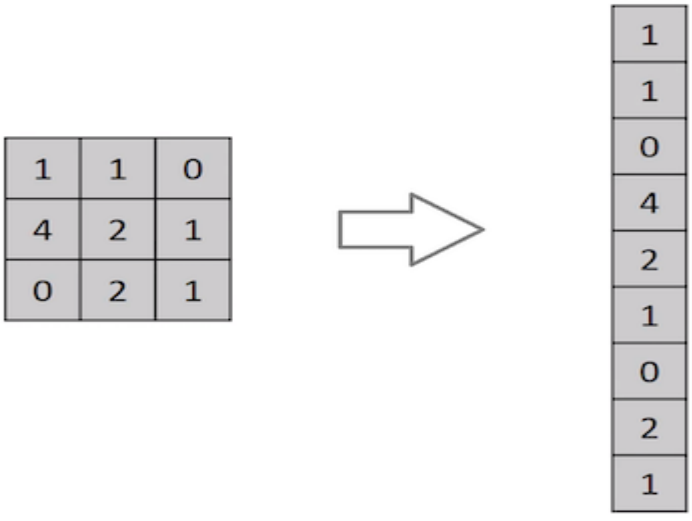

Figure 6: Conversion into 2D matrix

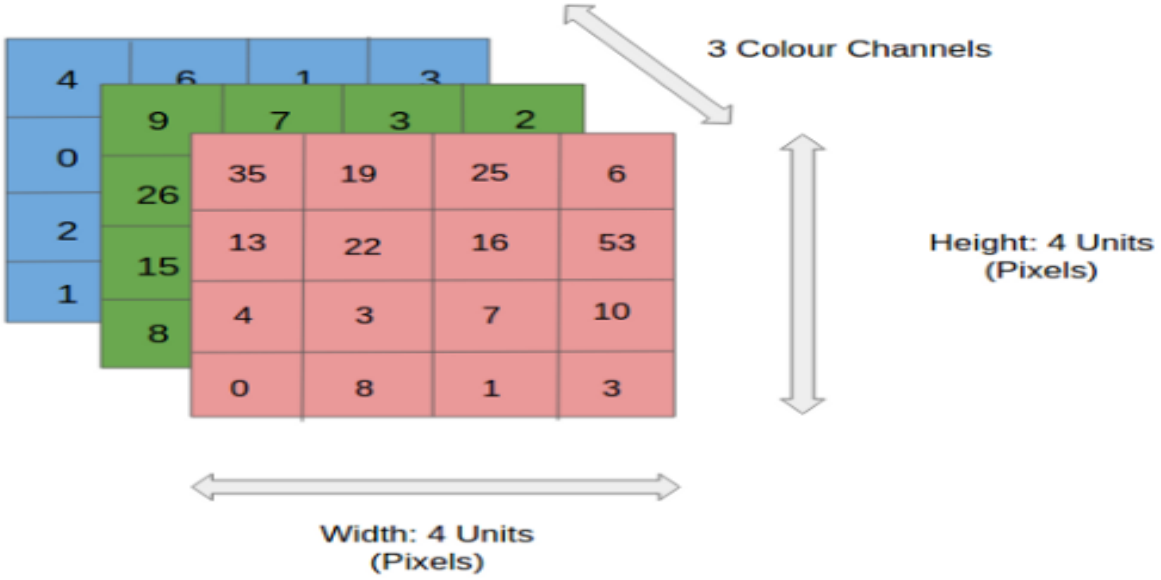

Figure 7: Input layer

3.2 Project Prerequisites

The necessity for this Python project is a webcam through which we will capture pictures. We need to have Python (3.6 version recommended) installed on our network, then, at that point, using pip, you can introduce the fundamental bundles.

OpenCV – pip install OpenCV-python (face and eye recognition).

TensorFlow – pip install TensorFlow (keras utilizes TensorFlow as backend).

Keras – pip install keras (to construct our classification model).

Pygame – pip install pygame (to play alarm sound).

3.3 Working of Project

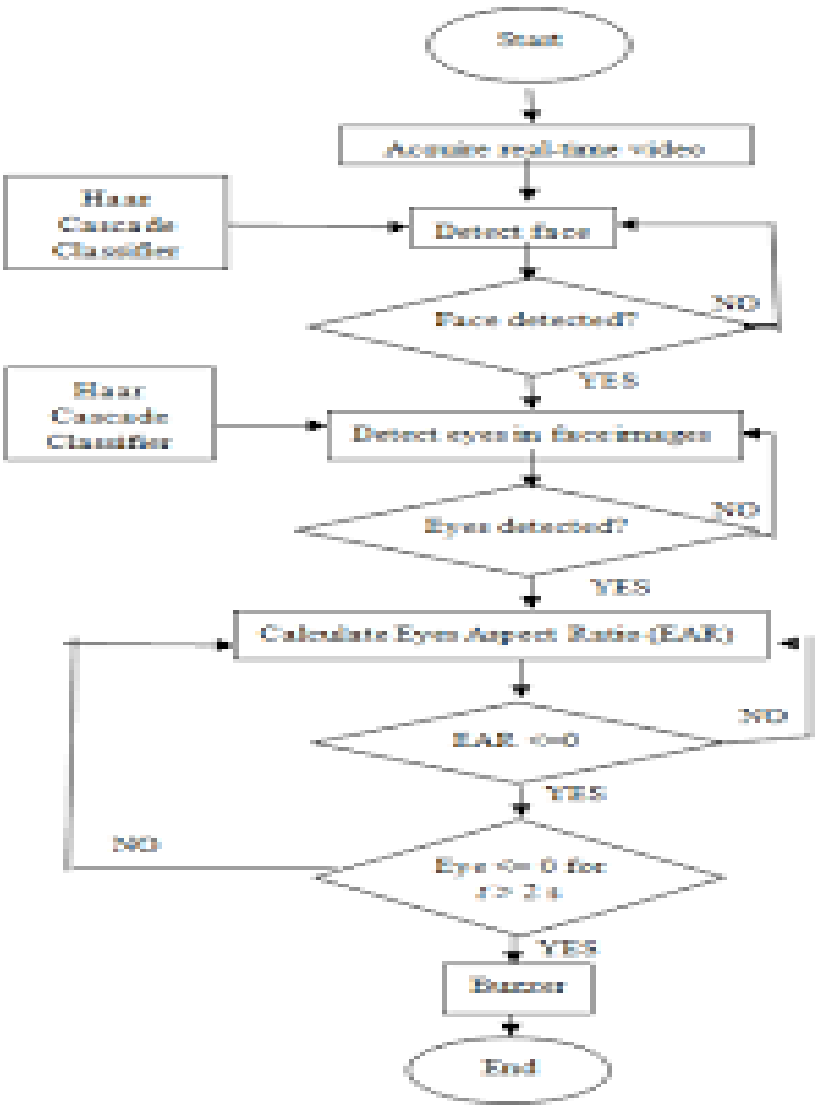

Figure 8: Flow Chart of drowsiness detection

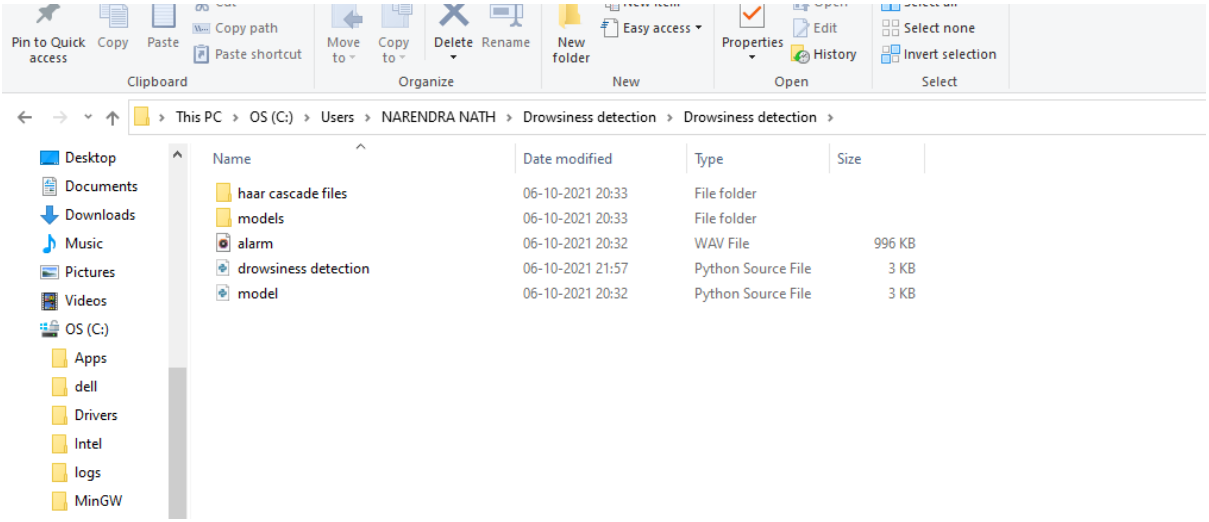

Figure 9: Folder/Files required for the project

The "haar cascade files" folder comprises of the xml records that are required to identify objects from the picture. For our situation, we are identifying the face and eyes of the individual.

The models folder contains our model file "cnnCat2.h5" which was trained on convolutional neural organizations.

We have a brief snippet "alarm.wav" which is played when the individual is feeling sleepy.

"Model.py" document contains the program through which we made our classification model via preparing on our dataset. You could see the execution of convolutional neural organization in this document.

"Drowsiness detection.py" is the primary record of our project. To begin the discovery method, we need to run this file.

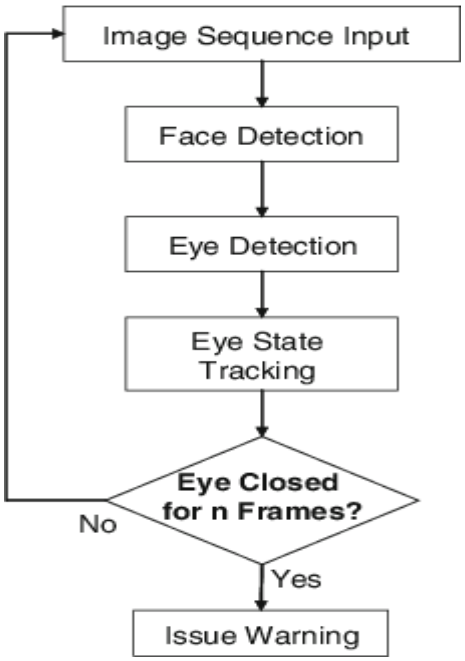

Figure 10: Flowchart of the project

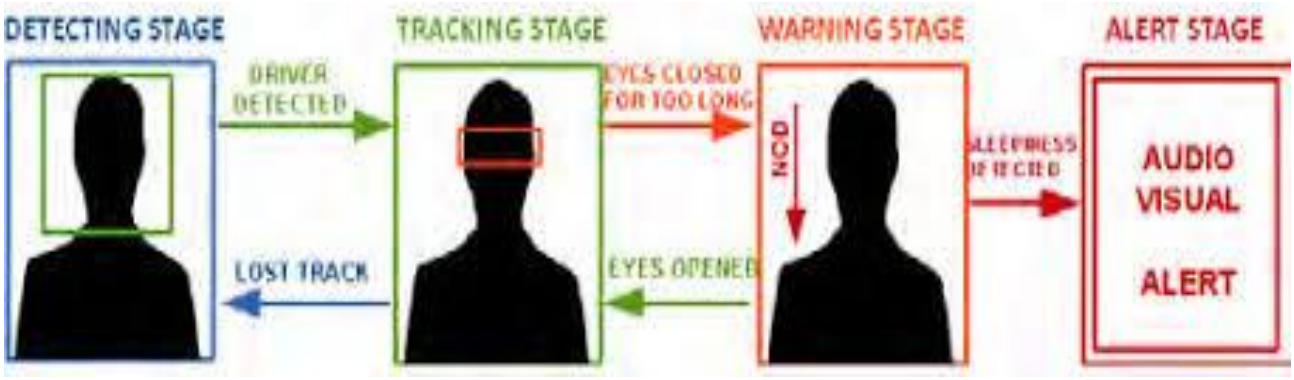

Figure 11: Working of the project

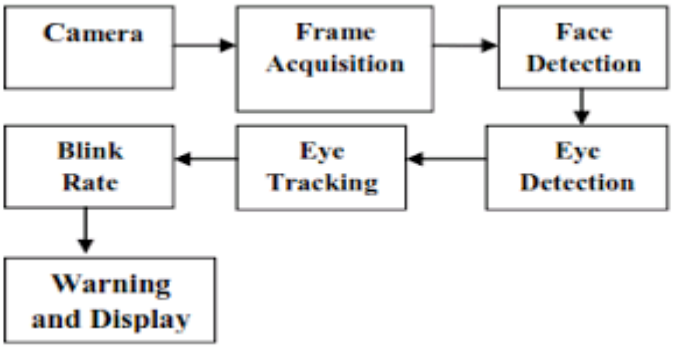

Figure 12: Flowchart

step 1 – Take Image as Input from a Camera

With a webcam, we will accept pictures as input. So, to get to the webcam, we made an endless circle that will catch each edge. We use the technique given by OpenCV, cv2.VideoCapture(0) to get to the camera and set the capture object (cap). cap.read() will read each frame and we store the picture in a frame variable.

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

To distinguish the face in the picture, we need to initially change over the picture into grayscale as the OpenCV calculation for object recognition takes dark pictures in the information. We needn't bother with shading data to recognize the articles. We will utilize haar cascade classifier to identify faces. This line is utilized to set our classifier face = cv2.CascadeClassifier(' way to our haar course xml record'). Then, at that point, we play out the location utilizing faces = face.detectMultiScale(gray). It returns an array of direction with x, y coordinates, and stature, the width of the limit box of the object. Presently we can repeat over the countenances and draw limit boxes for each face.

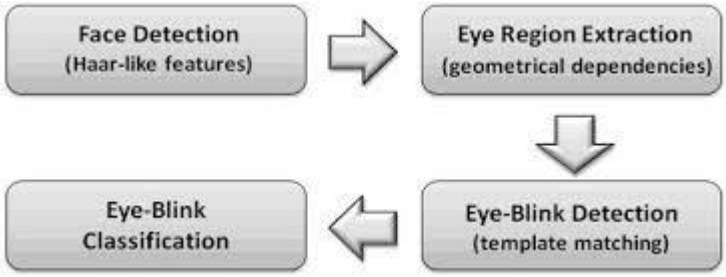

Figure 13: Block Diagram of drowsiness detection

Step 3 – Detect the eyes from ROI and feed it to the classifier

The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in leye and reye respectively then detect the eyes using left_eye = leye.detectMultiScale(gray). Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code.

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

l_eye only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into r_eye.

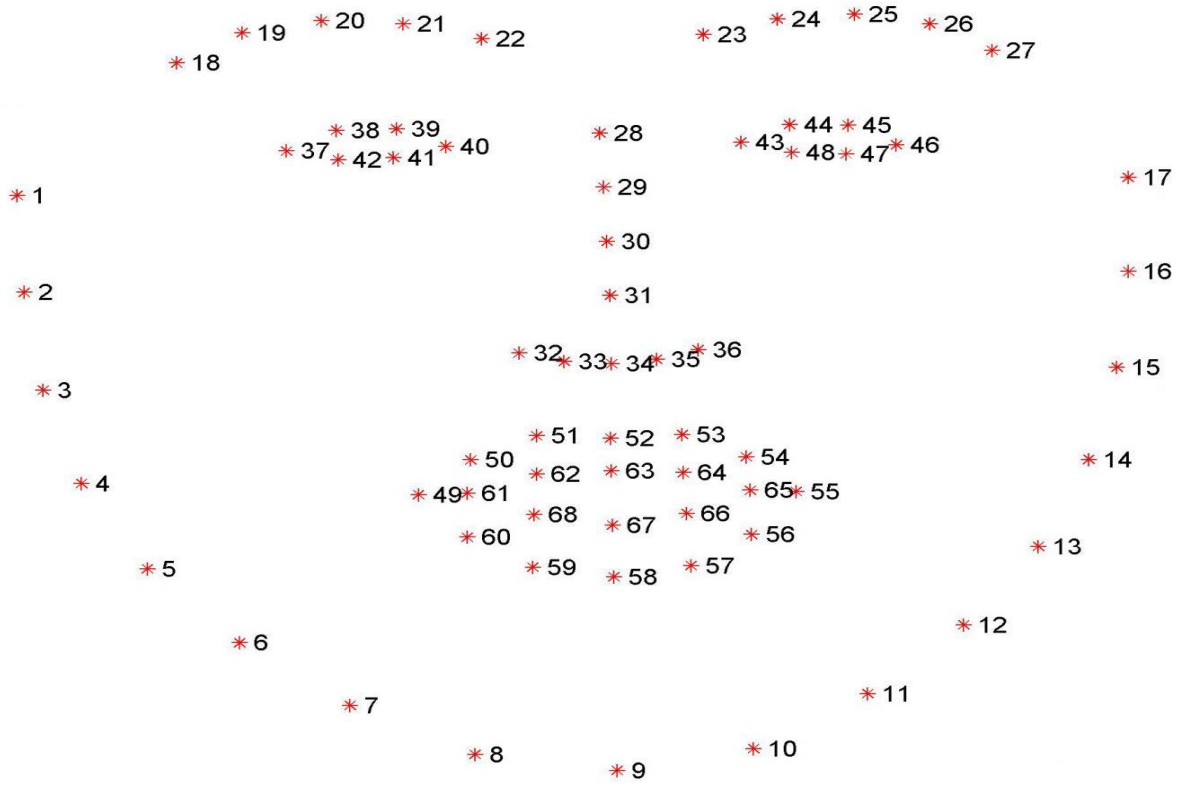

Figure 14: Visualizing the 68 facial landmark coordinates from the iBUG 300-W dataset (larger resolution).

Step 4 – Classifier will Categorize whether Eyes are Open or Closed

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the color image into grayscale using r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY). Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images cv2.resize(r_eye, (24,24)). We normalize our data for better convergence r_eye = r_eye/255 (All values will be between 0-1). Expand the dimensions to feed into our classifier. We loaded our model using model = load_model('models/cnnCat2.h5') . Now we predict each eye with our model

lpred = model.predict_classes(l_eye). If the value of lpred[0] = 1, it states that eyes are open, if value of lpred[0] = 0 then, it states that eyes are closed.

Step 5 – Calculate Score to Check whether Person is Drowsy

The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using cv2.putText() function which will display real time status of the person.

```
cv2.putText(frame, “Open”, (10, height-20), font, 1, (255,255,255), 1, cv2.LINE_AA )
```

A threshold is defined for example if score becomes greater than 15 that means the person’s eyes are closed for a long period of time. This is when we beep the alarm using sound.play()

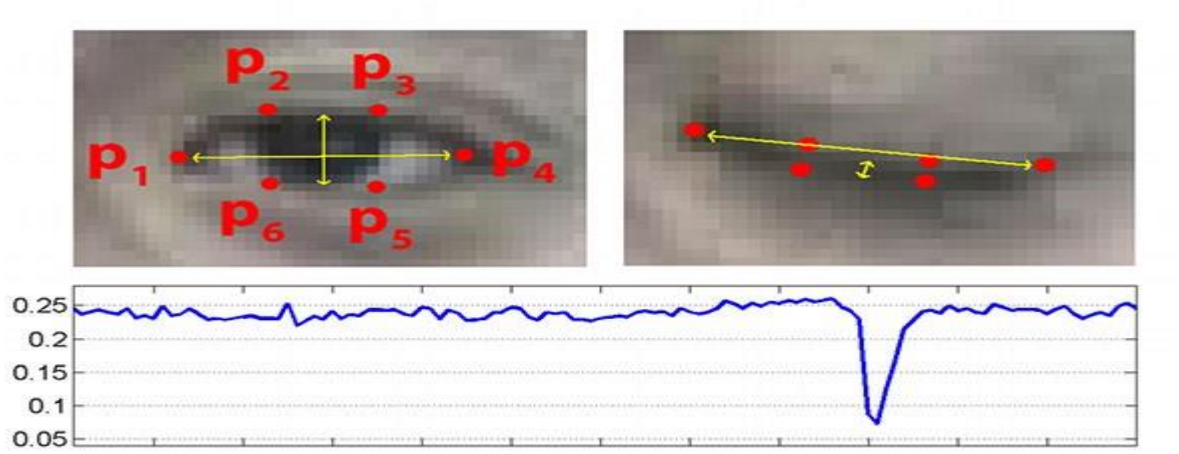

Figure 15: Top-left: A visualization of eye landmarks when then the eye is open. *Top-right:* Eye landmarks when the eye is closed. Bottom: Plotting the eye aspect ratio over time. The dip in the eye aspect ratio indicates a blink

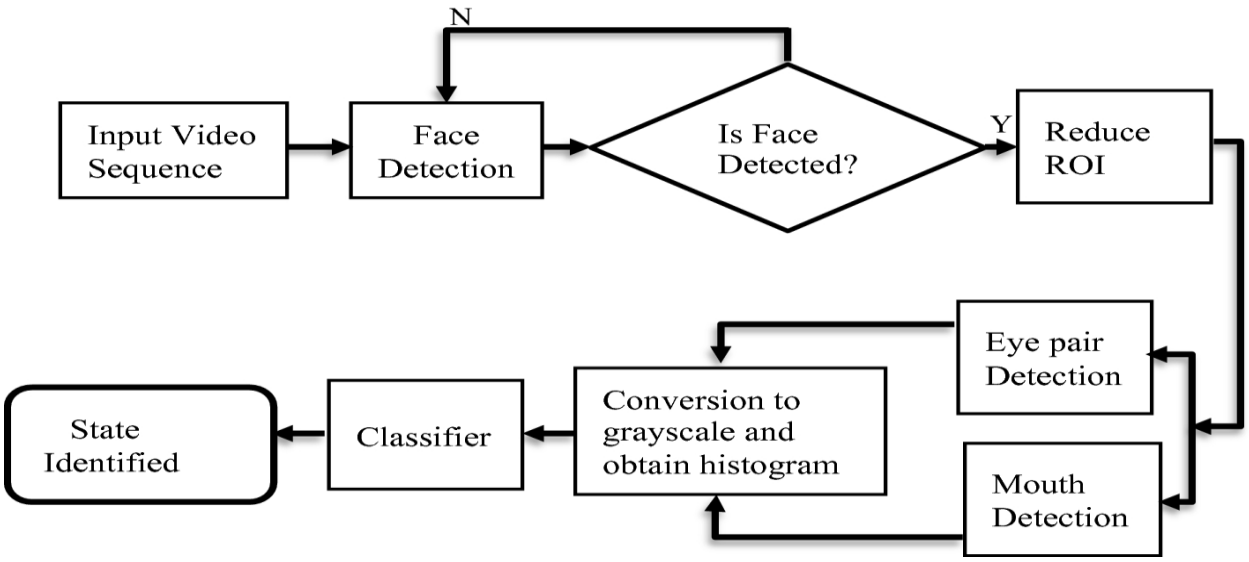

Figure 16: Block Diagram of drowsiness detection

3.4 Overview on OpenCV

CV: Computer Vision

Computer vision is a method for understanding how photos and movies are stored, as well as manipulating and retrieving data from them. Artificial Intelligence relies on or is mostly based on computer vision. Self-driving vehicles, robots, and picture editing applications all rely heavily on computer vision.

3.4.1 OpenCV

OpenCV is a large open-source library for computer vision, machine learning, and image processing, and it currently plays a critical part in real-time operations, which are critical in today's systems. It may be used to detect items, people, and even human handwriting in photos and movies. Python can process the OpenCV array structure for analysis when it is combined with other modules such as NumPy. We employ vector space and execute mathematical operations on these characteristics to identify visual patterns and their different features.

Look at the following image:

Figure 17: Example of OpenCV

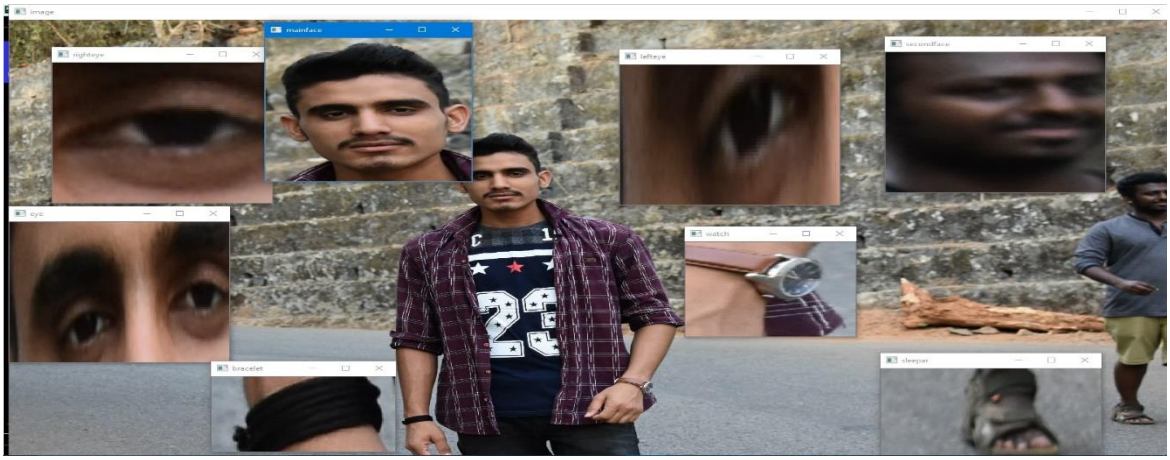

Figure 18: Example of OpenCV

Many bits of information that are contained in the original image may be acquired from the aforementioned original image. Because there are two faces accessible in the above image, and the person(I) in the image is wearing a bracelet, watch, and other jewellery , we can extract all of this information from the original image using OpenCV.

Applications of OpenCV: There are lots of applications which are solved using OpenCV, some of them are listed below

- face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anamoly (defect) detection in the manufacturing process (the odd defective products)

3.4.2 Image-Processing

Image processing is a technique for performing operations on a picture in order to improve it or extract relevant information from it.

"Picture processing is the study and modification of a digital image, notably in order to increase its quality," says the fundamental definition of image processing.

3.4.3 Digital- Image

An image is a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is termed the picture's intensity or grey level at that location.

In other words, a picture is nothing more than a two-dimensional matrix (3-D in the case of coloured images) described by the mathematical function f(x, y) at every location in the image, the pixel value at that place describing how bright that pixel is and what colour it should be. Image processing is essentially signal processing in which the input is an image and the output is an image or a set of characteristics based on the image's requirements.

Image processing basically includes the following three steps:

1. Importing the image
2. Analysing and manipulating the image
3. Output in which result can be altered image or report that is based on image analysis

How Does A Computer Read An Image?
Consider the below image:

Figure 19: Example of OpenCV

Because we are humans, we can quickly discern that this is the picture of a person who is me. However, if we ask the computer, "Is that my photo?" The computer can't say anything since it isn't figuring it all out on its own.

Any image is interpreted by the computer as a series of numbers ranging from 0 to 255. There are three basic channels in any colour image: red, green, and blue.

Chapter 4: Results and Discussion

4.1 Implementation

dataset_new: In this folder, we have our dataset that is downloaded by the link given above.
haar cascade files: This folder has files that are used to detect the face and eyes of a person, these files are xml files. The haar cascade files have many xml files that are required to detect objects in an image. You can download this also just by searching on google.
models: In this, we have our model 'custmodel.h5' file that we have created above.
alarm: This file is used to play the alert sound when a person closes its eyes for a few seconds.
model.py: It is the python file in which we have created our classification model which is trained on our dataset. You can see the implementation of how we have created the model and how we trained it according to our dataset.
my_drowsiness_detection.py: This file consists of full implementation of our project in which we have loaded the model(custmodel), and used it to alert the person whenever he/she will feel drowsy. So this is the main file, you have to run this file for detection procedure.

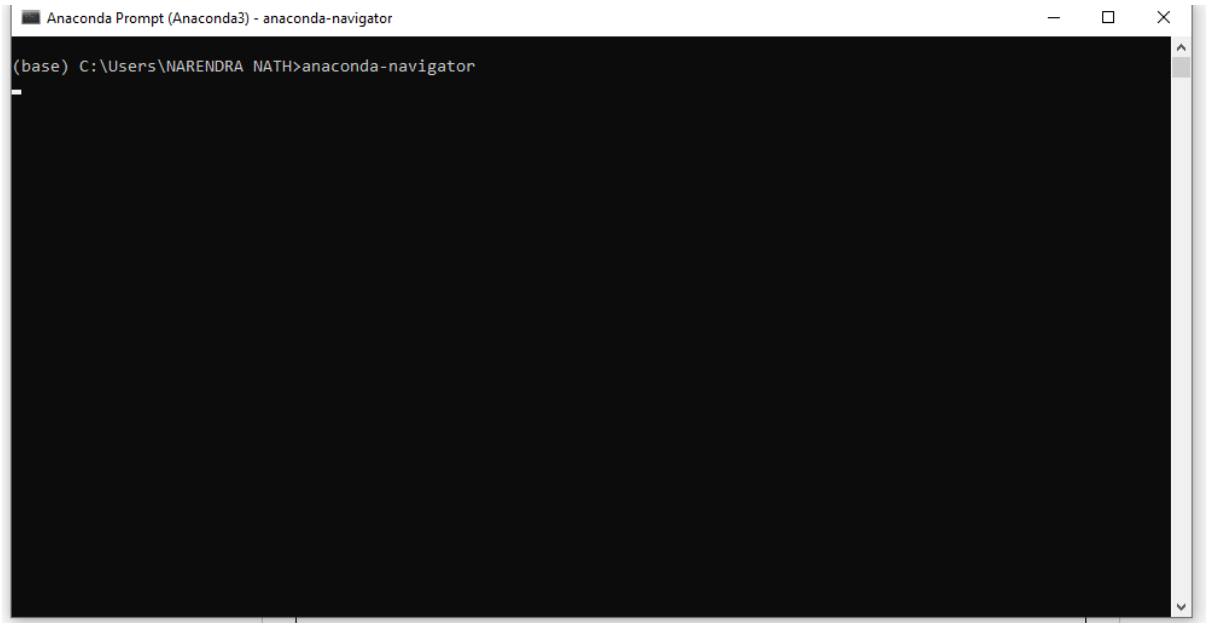

Figure 20: Running of Project

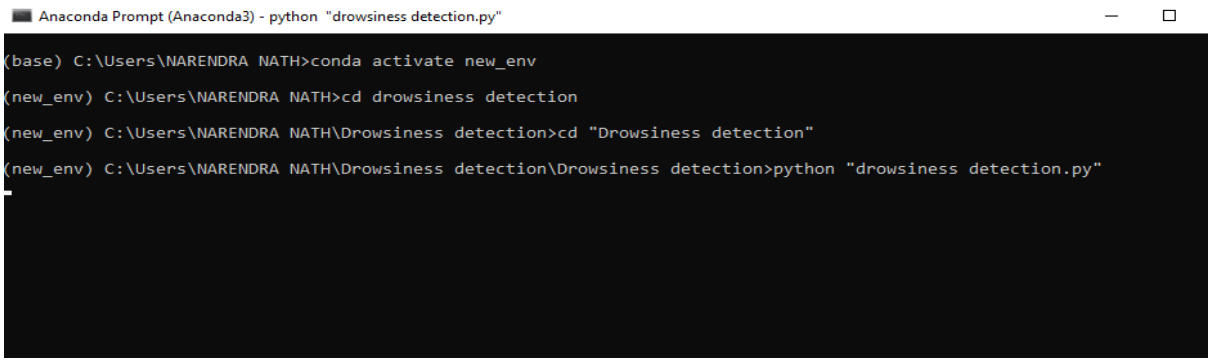

Figure 21: Running of Project

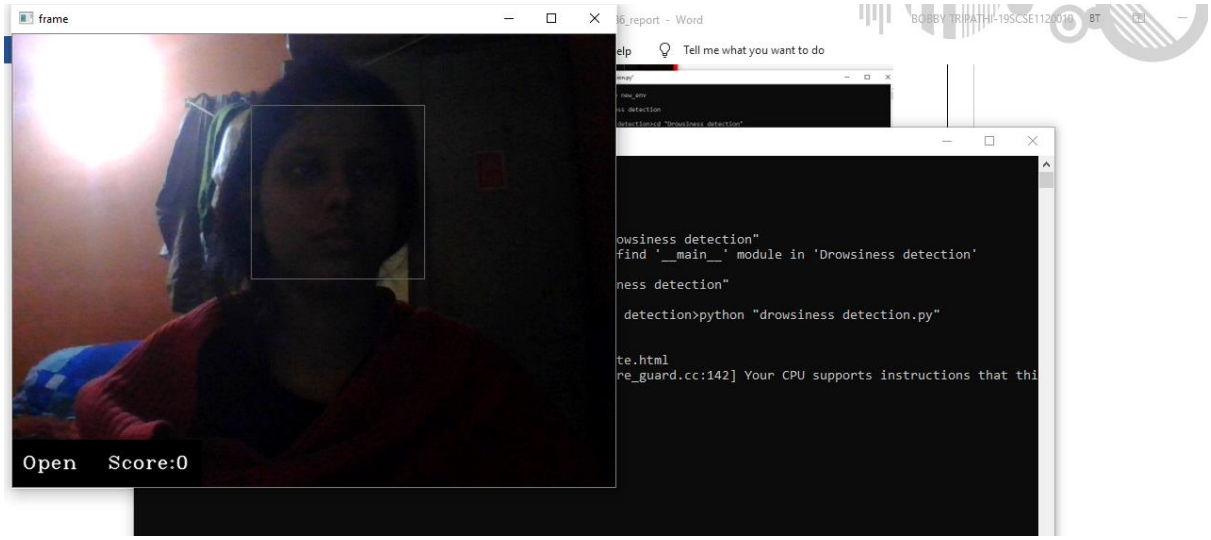

Figure 22: Running of Project

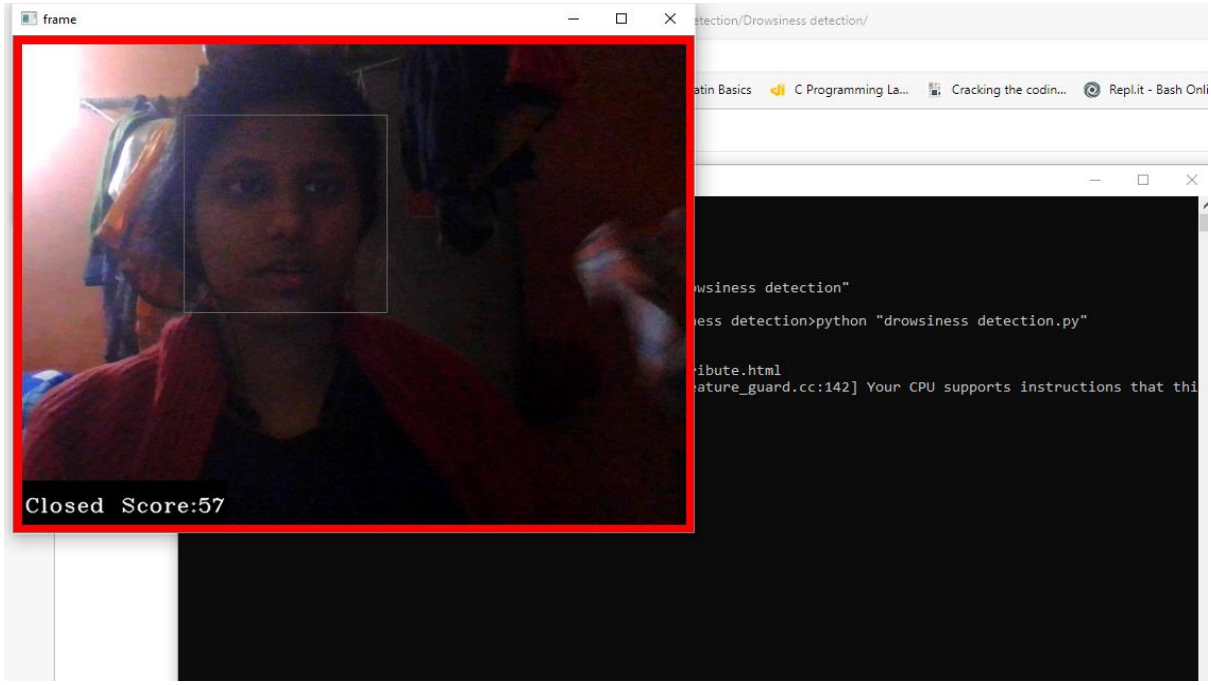

Figure 23: Running of Project

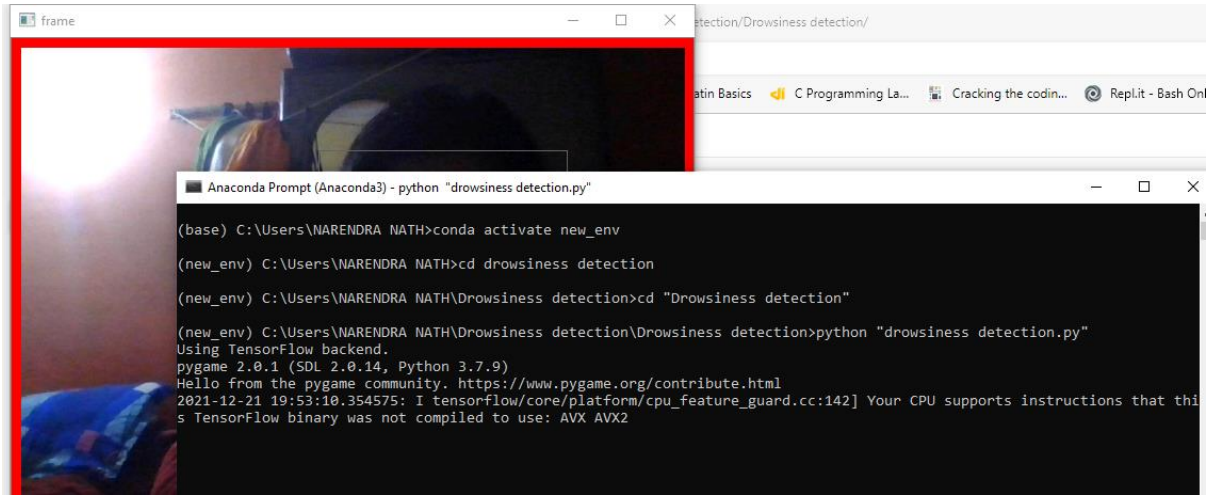

Figure 24: Running of Project

4.2 PROOF OF RESEARCH PAPER (COMMUNICATED)

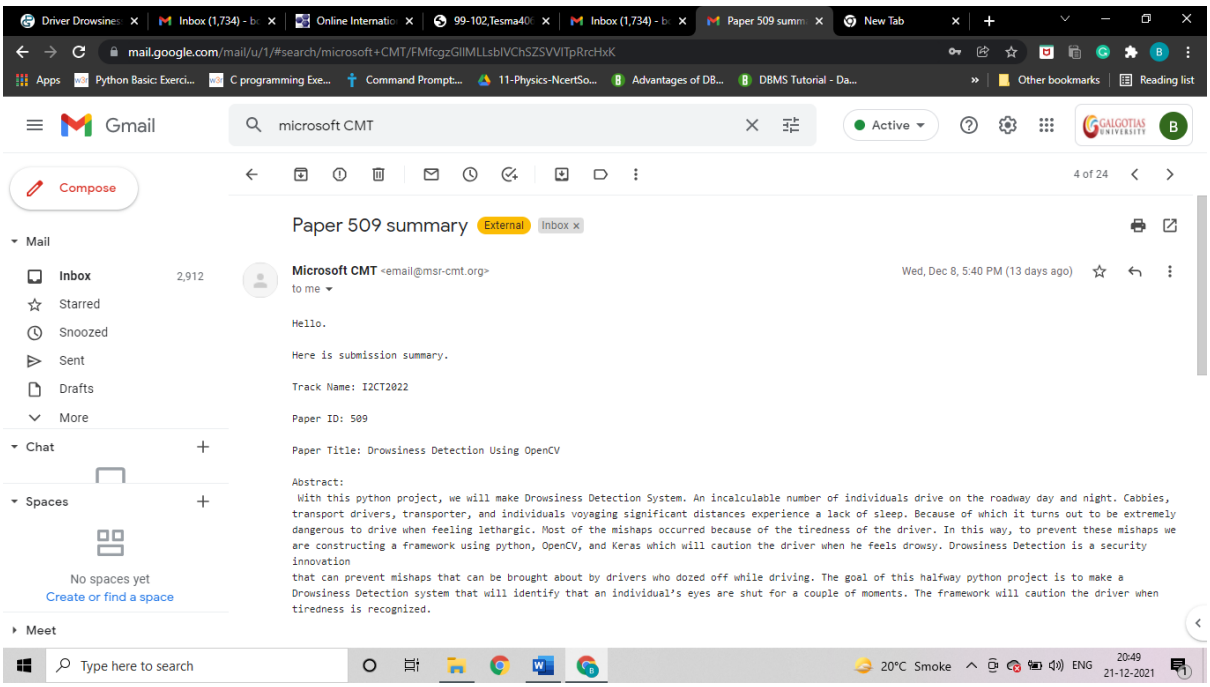

Figure 25: Submission of Research paper

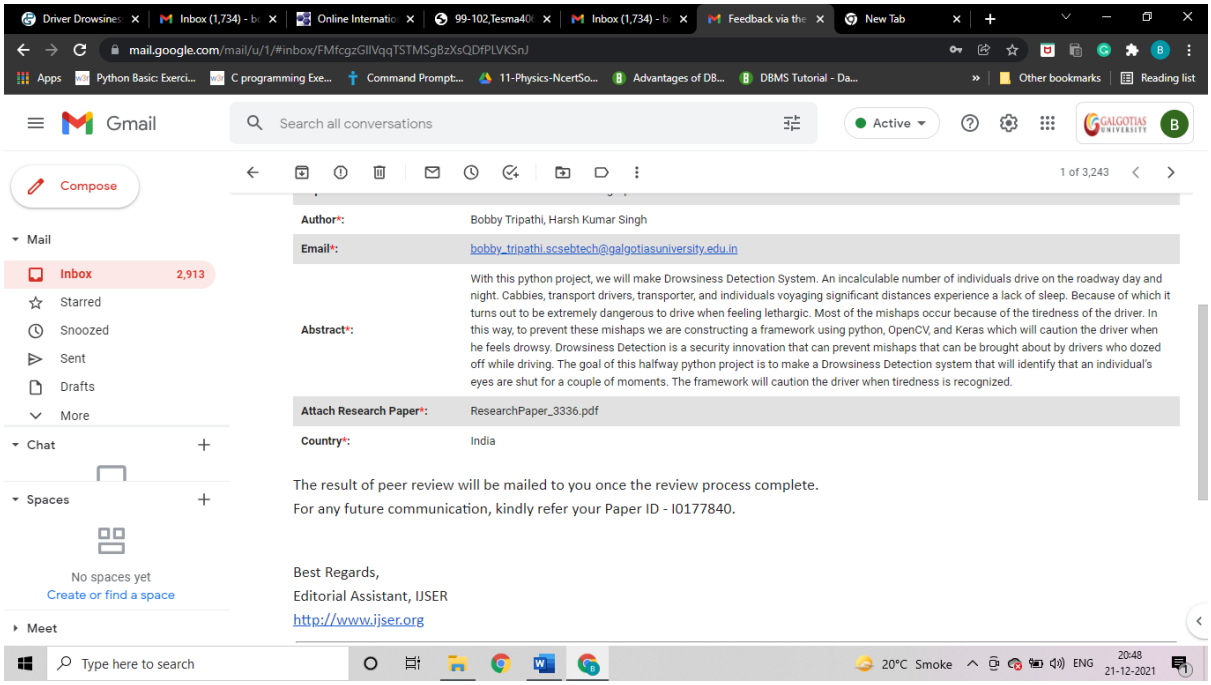

Figure 26: Submission of Research paper

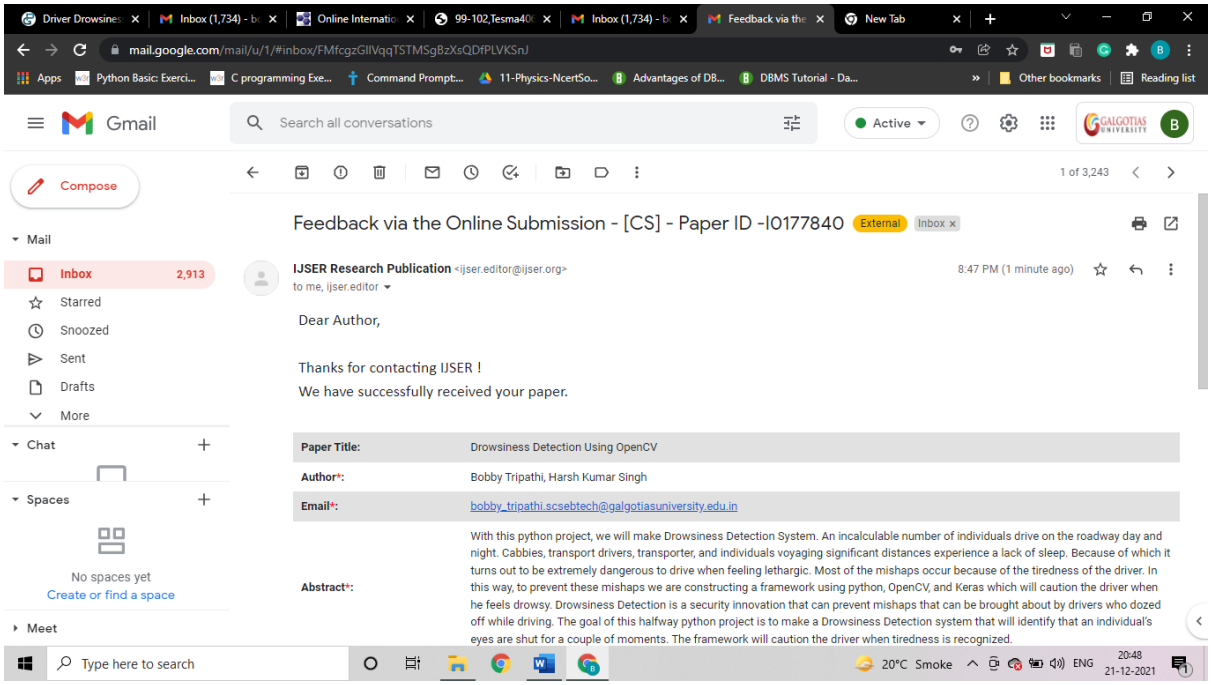

Figure 27: Submission of Research paper

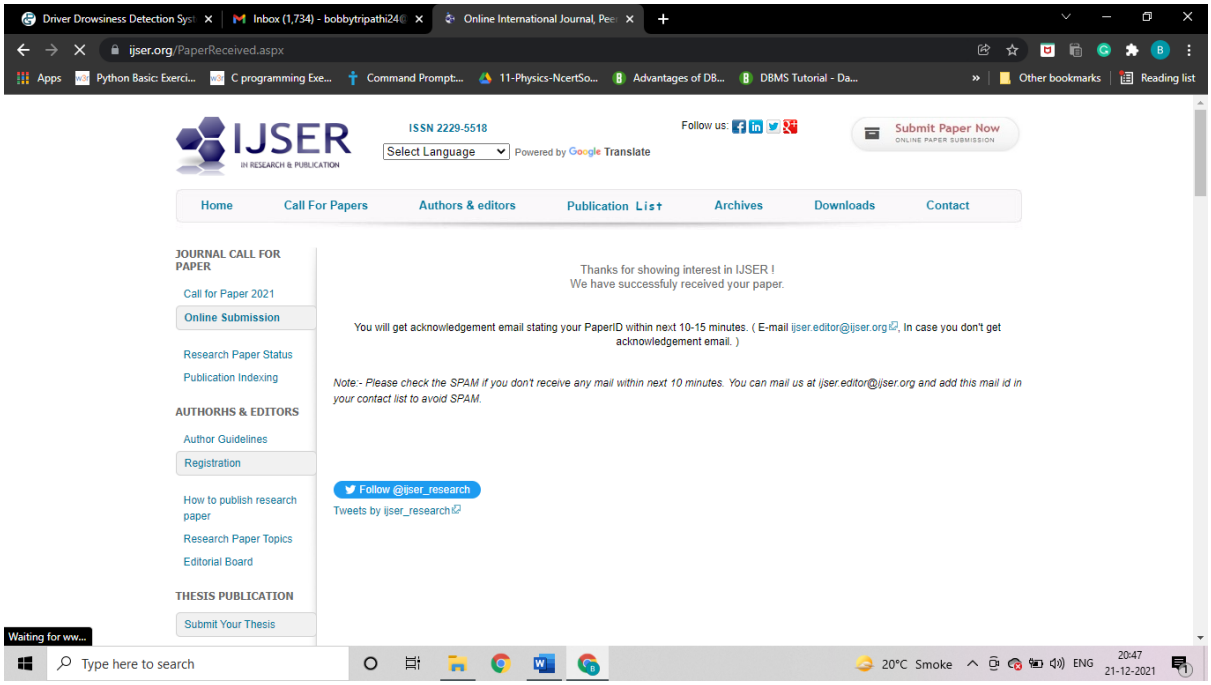

Figure 28: Submission of Research paper

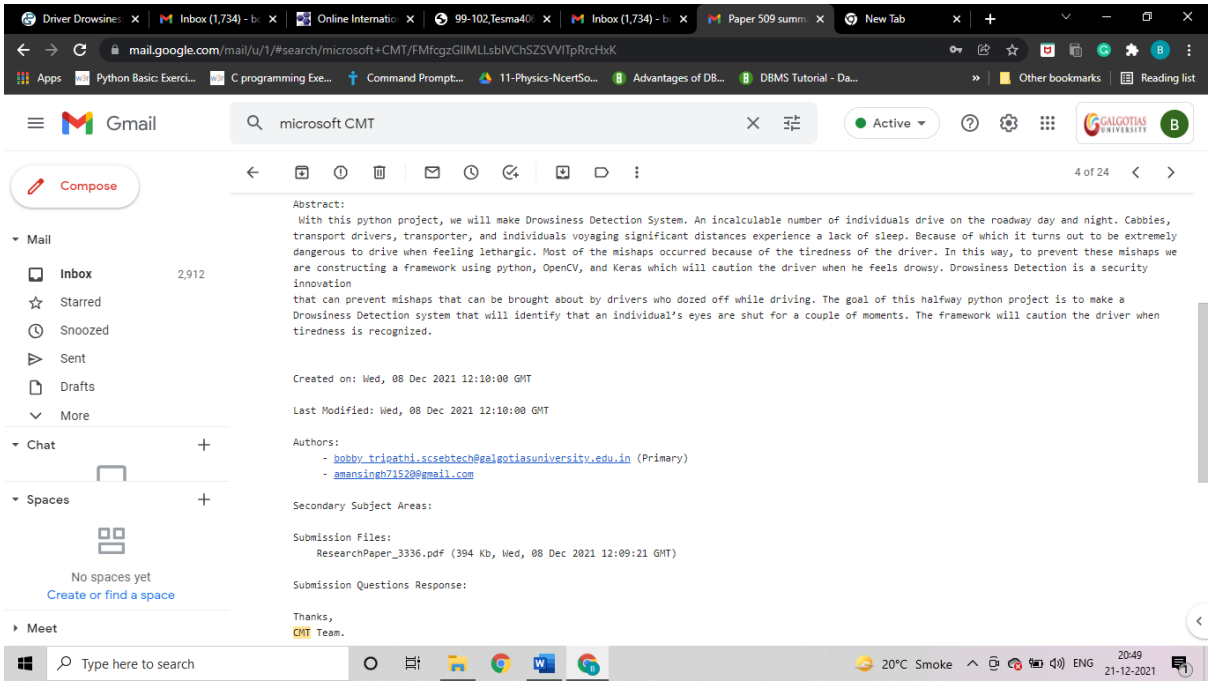

Figure 29: Submission of Research paper

Chapter 5: Conclusion and Future Scope

5.1 Conclusion

This research offered a technique for supporting drivers to avoid severe accidents caused by driver tiredness and alcohol use by assisting his/her state. Algorithms connected to image processing are used to determine the driver's state. A buzzer alerts the motorist if he or she is sleepy. The location of the driver's head is established using the center of gravity, and the driver's present condition is recognized as a result. The movement of the head is caught using a high-resolution camera. A system adds the capability of detecting yawning. If a motorist yawns too frequently, an alarm is triggered. A sensor is used to determine whether or not the motorist is intoxicated. For accuracy, there should be a sufficient distance between the sensor and the driver. To motivate the driver to arrive at their destination safely, an alert is created, which might take the form of audio or vibration. Despite the need for more study, the suggested system may efficiently identify the driver's state and significantly reduce the incidence of road accidents.

5.2 Future Scope

In this study, we created a drowsy driver caution architecture that may be executed in a variety of ways. We used OpenCV to detect faces and eyes using a haar cascade classifier, and then we used a CNN show to predict the outcome. This system will be enhanced further and will contain other safety features, such as limiting the number of persons who may access or drive the automobile. If an attempt is made to steal the vehicle, the engine will not start or a warning tone will be heard. In the case of an alleged robbery, a photograph of the robber is transmitted to the vehicle's owner, who will file a lawsuit against the thief.

Reference

https://arxiv.org/abs/1904.07312

https://pypi.org/project/opencv-python/

https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e

https://neurohive.io/en/popular-networks/vgg16/

[1] “Development of drowsiness detection system” VehicleResearch Laboratory, Nissan Center NISSAN MOTOR Co., Ltd. Hiroshi Ueno Masayuki Kaneda Masataka Tsukino

[2] “Face and eye tracking algorithm based on digital image processing “Department of Electrical Engineering, Universidad de Chile AV. Tupper 2007. Claudio A. Perez, Alvaro Palma, Carlos A. Holzmann and Christian Pera

[3] “Active Accident Avoidance Case Study: Integrating Drowsiness Monitoring System with Lateral Control and Speed Regulation in Passenger Vehicles” Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety Columbus, OH, USA. September 22- 24, 2008. Pınar Boyra, John H. L. Hansen.

[4] M. Hemamalini, P. Muhilan“Accident preventionusing eye blink sensor”, vol 1, Issue L11, 2017. RamalathaMarimuthu, A. Suresh, M. Alamelu and S.Kanagaraj “Driver fatigue detection using image processing and accident

prevention", International journal of pure and applied mathematics, vol. 116, 2017.

[5] TejaswiniJagdale, PradnyaJadhav, PrajaktaTotre,MayuraZadane, ShrilekhaMankhai "Driver drowsiness detection, alcohol detection and accident prevention", IJET, vol3 issue1, jan 2017.

[6] Drowsy Driving NHTSA reports. (2018, January 08). Retrieved from https://www.nhtsa.gov/risky-driving/drowsy-driving.

[7] Cornez T, Cornez R. Android Programming Concepts. Jones & Bartlett Publishers; 2015.

[8] K. Fagerberg.Vehicle-based detection of inattentive driving for integration in an adaptive lane departure warning system Drowsiness detection, M.S. thesis, KTH Signals Sensors and Systems, Stockholm, Sweden, 2004.

[9] Krajewski J, Sommer D, Trutschel U, Edwards D, Golz M. Steering wheel behavior based estimation of fatigue. The fifth international driving symposium on human factors in driver assessment, training and vehicle design 2009;118-124.