# A
# Project Report
# On
# " Driver Drowsiness Detection System "

**Prepared by**


| Shubham Mendapara (18DCS046) | Krish Pabani (18DCS054) | Yash Paneliya (18DCS058) |
|---|---|---|


**Under the Supervision of**

Prof. Vidhi Pandya

Assistant Professor cum Research Fellow

A Report Submitted to

Charotar University of Science and Technology

for Partial Fulfillment of the Requirements for the

5th Semester Software Group Project-III (CS348)

**Submitted at**



**CSE**

**DEPSTAR**

**At: Changa, Dist: Anand – 388421**


**MAY, 2020**

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

# CERTIFICATE

This is to certify that the report entitled "**Driver Drowsiness Detection System**" is a bonafied work carried out by **Mr. Shubham Mendapara (18DCS046), Mr. Krish Pabani (18DCS054), Mr. Yash S. Paneliya (18DCS058)** under the guidance and supervision of **Prof. Vidhi Pandya** for the subject CS348-**Software Group Project-III** (CSE) of 5th Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Vidhi Pandya
Assistant Professor cum Research Fellow
Department of Computer Science & Engineering
DEPSTAR, CHARUSAT, Changa, Gujarat.

Dr. Amit Ganatra
Principal, DEPSTAR
Dean, FTE
CHARUSAT, Changa, Gujarat.

# Abstract

Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.

The objective of this intermediate Python project is to build a drowsiness detection system that will detect that a person's eyes are closed for a few seconds. This system will alert the driver when drowsiness is detected.

# Acknowledgement

We are privileged to have this opportunity to express our gratitude and acknowledge to everyone's never ending support and valuable contributions for our project.

Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. I hereby avail this opportunity to express my gratitude to number of people who extended their valuable time, full support and cooperation in developing the project.

We would like to express my sincere gratitude to my advisor Mrs. Vidhi Ma'am for the continuous support to our project and project related research, for her patience, motivation, and immense knowledge.

Our sincere thanks also go to Dean Sir Dr. Amit Ganatra who provided us an opportunity to work on a project and to be able to present the same.

At last, I thank my parents and friends for their constant encouragement without which this project would not have been possible.

# Table of Contents

Abstract.....i
Acknowledgement.....ii
Table of Contents.....iii
List of Figures.....iv
**Chapter 1: PROJECT DEFINITION**.....5
1.1 Project Summary.....5
1.2 Purpose.....5
1.3 Scope.....5
1.4 Objective.....5
**Chapter 2: PROJECT DESCRIPTON**.....6
2.1 Project Planning.....6
2.1.1 Project Effort and Time.....6
2.1.2 Roles and Responsibilities.....6
2.2 The Dataset.....7
2.3 The Model Architecture.....7
**Chapter 3: SYSTEMS REQUIREMENTS**.....8
3.1 User Characteristics.....8
3.2 Hardware Requirements.....8
3.3 Software Requirements.....8
**Chapter 4: MAJOR FUNCTIONALITY**.....9
**Chapter 5: SYSTEM ANALYSIS**.....10
**Chapter 6: SCREENSHOTS OF THE PROJECT**.....12
**Chapter 7: LIMITATIONS**.....14
**Chapter 8: FUTURE ENHANCEMENT**.....15
**Chapter 9: CONCLUSION AND DISCUSSION**.....16
9.1 Self-Analysis of Project Viabilities.....16
9.2 Problem Encountered.....16
9.3 Summary of Project Work.....16
**Chapter 10: REFERENCES**.....17

# List of Figures

Fig 5.1 Flowchart.....9
Fig 6.1 Open Eyes.....10
Fig 6.2 Closed Eyes.....11

# Chapter 1: PROJECT DEFINITION

## 1.1 Project Summary

Our project is on Driver Drowsiness Detection System.
This Project has been made for user protection. This ML model is well organized with proper dataset and accuracy.
For developing a ML modal, we use PyCharm. It is an open source software. For image we have used OpenCV

- You'll never get Sleepy at driving time!
- Sound start if person found Drowsy.
- Help to avoid accident.

## 1.2 Purpose

The main purpose for making this modal was, as we see now a day about 30% of accident are happening due to sleepily or drowsiness of driver while driving. By this we are think that we can help to reduce the people's life.

## 1.3 Scope

The idea of this project can be a great idea for startup for us. We will share this project to as many people as we can.

## 1.4 Objective

The main objective of this project is to reduce to accident to save people's life and to make world more digital and risk free.

# Chapter 2: PROJECT DESCRIPTION

## 2.1 Project Planning

### 2.1.1 Project Effort and Time

The programmers have given about 1-2 hours per day to make this project.

### 2.1.2 Roles and Responsibilities

Team Members have worked effectively and have undertaken the following tasks:

- Researching in the related field of work
- Creating a step to follow
- Connecting to OpenCV
- Detected the driver face
- Detected the eye on face
- Run modal to classify the score whether open or close
- Finally calculates percentage of Drowsiness

## 2.2 The Dataset

The dataset used for this model is created by us. To create the dataset, we wrote a script that captures eyes from a camera and stores in our local disk. We separated them into their respective labels ‘Open’ or ‘Closed’. The data was manually cleaned by removing the unwanted images which were not necessary for building the model. The data comprises around 7000 images of people’s eyes under different lighting conditions. After training the model on our dataset, we have attached the final weights and model architecture file “models/cnnCat2.h5”.

Now, you can use this model to classify if a person’s eye is open or closed.

## 2.3 The Model Architecture

The model we used is built with Keras using Convolutional Neural Networks (CNN). A convolutional neural network is a special type of deep neural network which performs extremely well for image classification purposes. A CNN basically consists of an input layer, an output layer and a hidden layer which can have multiple numbers of layers. A convolution operation is performed on these layers using a filter that performs 2D matrix multiplication on the layer and filter.

The CNN model architecture consists of the following layers:

Convolutional layer; 32 nodes, kernel size 3
Convolutional layer; 32 nodes, kernel size 3
Convolutional layer; 64 nodes, kernel size 3
Fully connected layer; 128 nodes
The final layer is also a fully connected layer with 2 nodes. In all the layers, a Relu activation function is used except the output layer in which we used Softmax.

# Chapter 3: SYSTEMS REQUIREMENTS

## 3.1 User Characteristics

This Model has been made for user protection. ML model is well organized with proper dataset and accuracy.

## 3.2 Hardware Requirements

- Processor: i3/i5/i7 8gen or above
- Processor Speed: 2.16GHZ or above
- RAM: 3 GB RAM or above
- Storage: 20 GB hard disk or above

## 3.3 Software Requirements

- Pycharm
- OpenCV
- TensorFlow

# Chapter 4: MAJOR FUNCTIONALITY

For developing a ML modal, we use PyCharm. It is an open source software. For image we have used OpenCV

- **You'll never get Sleepy at driving time!**

  This will to avoid Drowsiness related Accident by this driver didn't try to sleep will driving.

- **Sound start if person found Drowsy**

  As soon as this device find that drive is in sleep position this will start alarm, which help driver to wake.
  .

# Chapter 5: SYSTEM ANALYSIS

When the program starts, we'll start capturing the video of the user using OpenCV. We'll capture all the frames of video and detect whether the eyes are close or open. Let's go in a bit detail.

## Step 1 – Take Image as Input from a Camera

With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the image in a frame variable.

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

To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier face = cv2.CascadeClassifier(' path to our haar cascade xml file'). Then we perform the detection using faces = face.detectMultiScale(gray). It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face.

## 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 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.

## 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. 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().

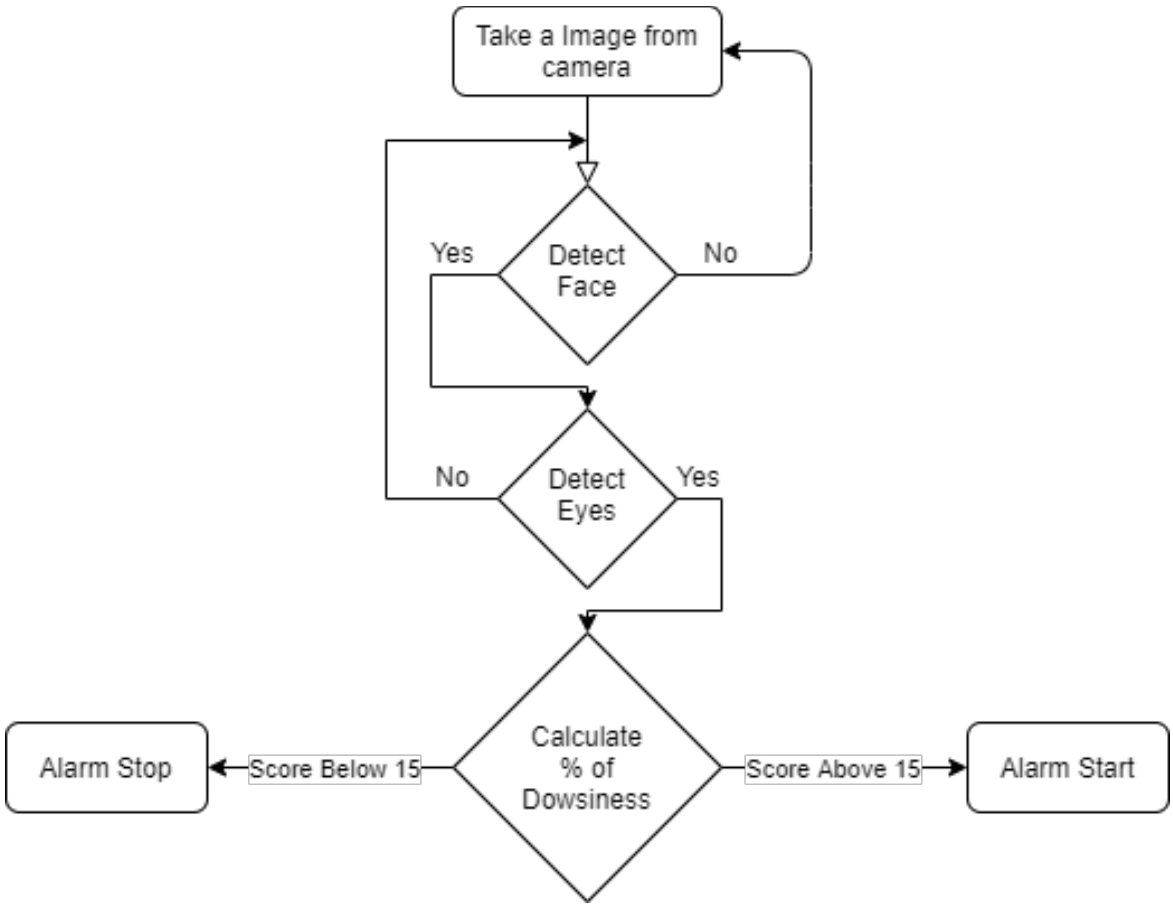


Fig 5.1: Flow Chart

# Chapter 6: SCREENSHOTS OF THE PROJECT

Driver Drowsiness Detection System will detect when the eyes are open there will be NO red line and alarm sound, it shows driver is awake.

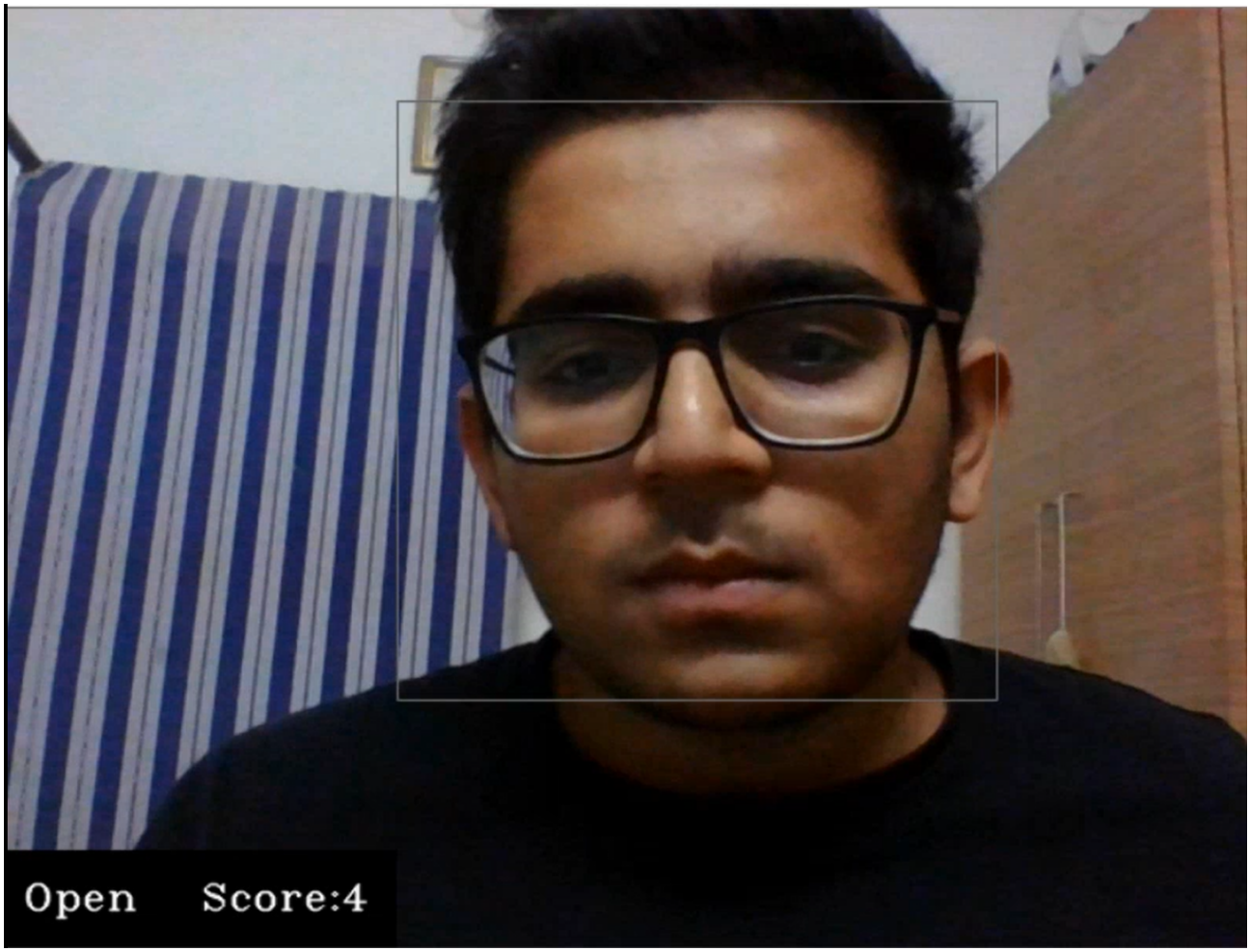


Fig 6.1: Open Eyes

Driver Drowsiness Detection System will detect when the eyes are closed there will be a red line and alarm will start beeping, it will awake driver before a mishap happens.

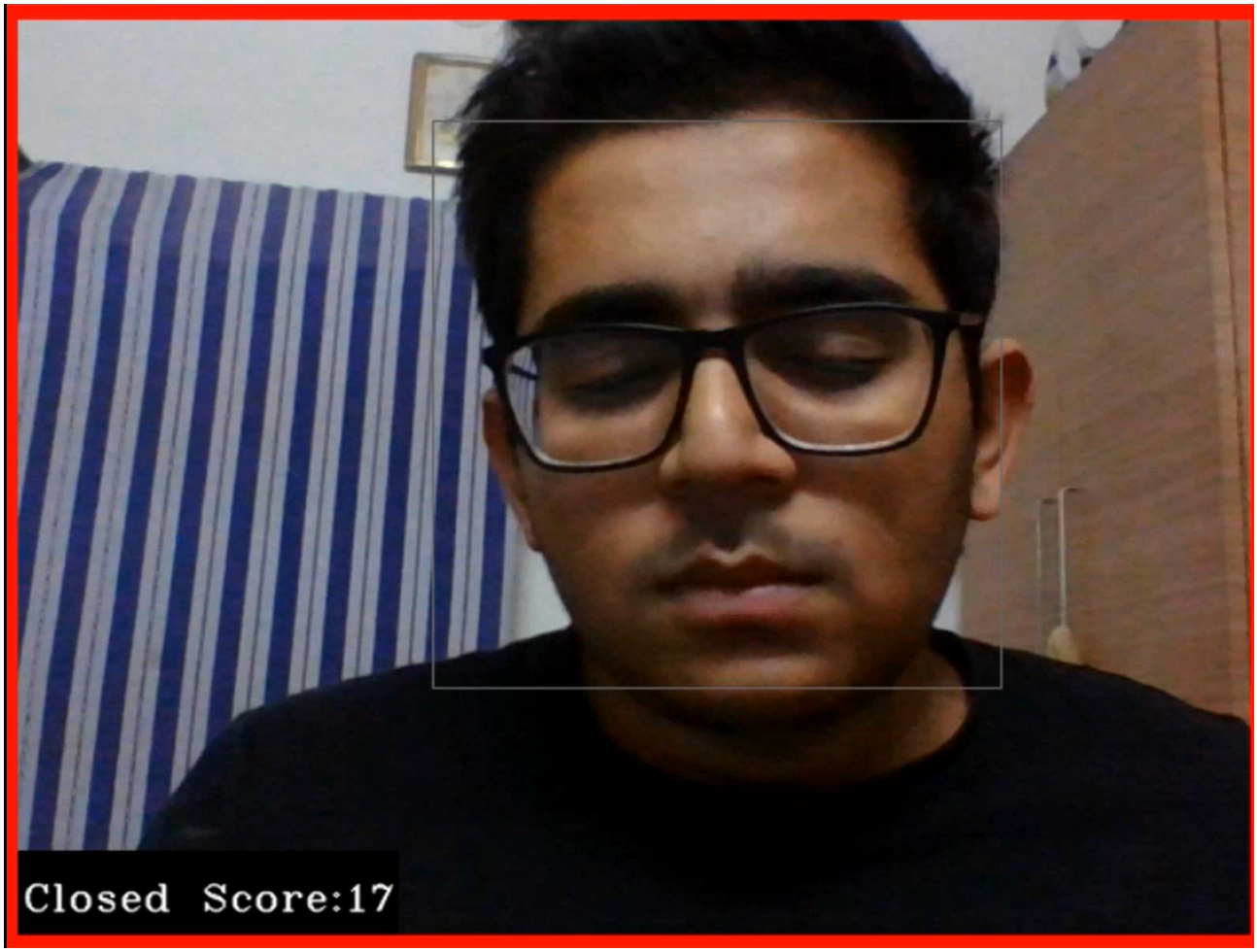


Fig 6.2: Closed Eyes

# Chapter 7: LIMITATIONS

Following are the currently persisting limitations in the application:

- Learning python was bit challenging.
- Moving from Development to ML
- Managing Large Dataset with ML model

# Chapter 8: FUTURE ENHANCEMENT

Following are some aspirations specifying our future enhancements:

- Testing ML model in real life Situations.
- What to improve dataset so more our modal user friendly.
- Increasing Accuracy so that our modal can be more Adaptable.

# Chapter 9: CONCLUSION AND DISCUSSION

## Self-Analysis of Project Viabilities

The project was a good experience for us. We think that from this project we learnt a lot about how a dataset works. We learnt how to develop an ML model with large data set.

## Problem Encountered

- Learning python was bit challenging.
- Moving from Development to ML.
- Managing Large Dataset with ML model

## Summary of Project Work

The project was a great experience of working in a team. The importance of time bound and coordinated execution of work was realized. It gave us an experience to develop ML model for detection of sleepy person at time of driving.
The application is a user friendly and can be run by any person with the help of PyCharm in desktop. The preparation of this project has helped a lot to learn the much unknown features of OpenCV, Keras, TensorFlow.

# Chapter 10: REFERENCES


- https://www.infoq.com/presentations/understanding-deep-learning/
- https://www.youtube.com/playlist?list=PL3GCZkoyKK4ehcDlwaVD8fdiS-w2U6nU
- https://www.youtube.com/playlist?list=PL9ooVrP1hQOFJ8UZl86fYfmB1_P5yGzBT
- https://www.youtube.com/playlist?list=PLS1QulWo1RIa7D1O6skqDQ-JZ1GGHKK-K