A PROJECT REPORT ON

DRIVER DROWSINESS DETECTION USING OPENCV & DLIB

Submitted to the partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

Submitted by

RITVIK SONI [Reg No: RA1911003030070]

KSHITIJ MITTAL [Reg No: RA1911003030077]

SIMRAN GARG[Reg No:RA1911003030102]

RISHABH MALASI[Reg No:RA19003030109]

Supervised by

Dr. Anand Pandey

(Assistant Professor, Department of Computer Science & Engineering)

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
DELHI-NCR CAMPUS, GHAZIABAD (U.P)

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

Delhi-NCR Campus, Modinagar,

Ghaziabad (UP) - 201204

May 2023

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

This is to certify that project Report entitled "**DRIVER DROWSINESS DETECTION USING OPENCV AND DLIB**" which is submitted by **RITVIK SONI [REG NO: RA1911003030070], KSHITIJ MITTAL [Reg No: RA1911003030077] SIMRAN GARG[Reg No:RA1911003030102] AND RISHABH MALASI[Reg No:RA19003030109]",** in the partial fulfillment of the requirement for the award of degree B.Tech(CSE) of SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, Ghaziabad is a record of the candidate's own work carried out by them under my own supervision.

………………………..	……………………….
SIGNATURE	**SIGNATURE**
Dr. Anand Pandey **GUIDE** Associate Professor Dept. of Computer Science & Engi- neering	Dr. Akash Punhani **HEAD OF THE DEPARTMENT** Dept. of Computer Science & Engi- neering
Signature of the Internal Examiner	Signature of the External Examiner

DECLARATION

We, Ritvik Soni (RA1911003030070), Kshitij Mittal (RA1911003030077), Simran Garg (RA1911003030102) and Rishabh Malasi (RA1911003030109) hereby declare that the work which is being presented in the project report "**DRIVER DROWSINESS DETECTION USING OPENCV & DLIB**" is the record of authentic work carried out by us during the period from January '12 to May '12 and submitted by us in partial fulfillment for the award of the degree "Bachelor of Technology in Computer Science and Engineering" to SRM IST, NCR Campus, Ghaziabad (U.P.). This work has not been submitted to any other University or Institute for the award of any Degree/Diploma.

Ritvik Soni (RA1911003030070)

Kshitij Mittal (RA1911003030077)

Simran Garg (RA1911003030102)

Rishabh Malasi (RA1911003030109)

ACKNOWLEDGEMENTS

We would also like to express our special thanks to Dr. Anand Pandey, Assistant Professor and Project Supervisor of SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his insight and knowledge on the subject, who imbued us with the feeling to work assiduously.

We feel compelled to articulate our thankfulness to Dr. Jitendra Singh, and Dr.Rakesh Kumar Yadav, Project Coordinators, SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for there enlightening dexterity which was truly a perennial source of inspiration.

I would also like to extend my sincere thanks to Dr. S. Vishwanathan, Director of SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his unwavering support that enabled me to undertake and complete my project work.

My special thanks go to Dr. D. K. Sharma, Dean (Academics), and Dr. R. P. Mahapatra, Dean (E&T) at SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for their valuable guidance and unconditional support.

Also, we extend our sincere regards and thanks to Dr. Akash Punhani, Head of the Department, Computer Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his suggestions.

We are also indebted to all the teaching and non-teaching staff members of our college for helping us directly or indirectly by all means throughout the course of our study and project work.

Finally, we take this opportunity to thank our parents, family members and friends for their moral support and help and extended thanks to our well wishers.

Ritvik Soni **Kshitij Mittal**

Simran Garg **Rishabh Malasi**

ABSTRACT

In the vehicle security system we recognize the driver's face and give alerts whenever the driver is drowsy and he blinks his eyes and an alarm is ringed to maintain their safety. Drowsiness is one of the main reasons/ major causes these days for road accidents. Here, to avoid such kinds of accidents we're developing a system which is a drowsiness alert system and vehicle's safety system. By using Artificial Intelligence (AI) technology, we're building this system. Firstly, the image of the driver is captured and it is identified by using face recognition techniques and once the driver is in the vehicle and he starts driving the vehicle, for instance if he feels drowsy there will be an alert/ alarm so that he can get himself awake, take a break and then drive the vehicle.

Computer vision and machine learning algorithms are used to design this system. In this system, we're using eye landmarks which determine the EAR (Eye Aspect Ratio ratio) to check whether the driver is drowsy. Face recognition is determined by object detection techniques using Haar Cascade algorithm & LBPH in OpenCV and we're making this system user friendly by adding Graphical user Interface (GUI) using Tkinter. This model can predict with an accuracy rate of 90% and further can be improvised using huge datasets.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. **EKG**	Electrocardiogram
2. **EEG**	Electroencephalogram
3. **EMG**	Electromyography
4. **HP**	Head Posture
5. **EI**	Eye Index
6. **SVM**	Support Vector Machines
7. **PCA**	Principle Component Analysis
8. **LBP**	Local Binary Pattern
9. **GUI**	Graphical user interface

CHAPTER 1

INTRODUCTION

Throughout history, humans have developed machines and technology to simplify and enhance their lives, from the earliest cooking methods to modern forms of transportation. As technology continues to advance, our expectations of transportation have also evolved, and we are now able to travel to places that were once impossible for our grandparents. While almost everyone in the world relies on some form of transportation, there are universal rules and practices that all drivers must follow, regardless of their social status. Being alert and focused while driving is crucial, as neglecting this responsibility can lead to numerous accidents each year, many of which are connected to driver fatigue.

To combat this issue, researchers have studied the topic of driving fatigue, though the data and analysis from machines alone is often insufficient. Recent statistics suggest that in India alone, around 155,622 people may die in traffic accidents in 2022, with at least 15% of these fatalities being caused by tired or faulty drivers. Although this may seem like a small percentage, any loss of life due to driver fatigue is a cause for concern. Unfortunately, driver fatigue is often overlooked as a significant problem in car accidents. In some countries, such as India, a lack of infrastructure combined with driver fatigue can lead to disastrous consequences.

Unlike addictions to drugs or alcohol, which have clear symptoms and tests for detection, fatigue is difficult to monitor or identify. The best solution to this problem is for drivers to recognize when they are feeling fatigued and take appropriate action, such as pulling over to rest. Encouraging drivers to become more aware of fatigue-related accidents and recognizing when they need a break is crucial in addressing this issue. In addition, restrictions on driving hours have been implemented in some countries, but this alone is not enough to solve the problem, as it can be costly and difficult to enforce. Ultimately, improving driver awareness and promoting safer driving practices are essential in reducing the number of accidents caused by driver fatigue.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Numerous studies have been conducted in the past on identifying drowsy drivers, which can aid in promptly detecting and reconstructing driver fatigue. Different techniques have been developed to detect symptoms of sleep disorders. The World Health Organization has stated that approximately 100,000 accidents occur due to drowsy driving.

This section covers various methods that researchers have proposed for detecting drowsiness and blink detection in recent years:

2.1.1. Drowsiness and Fatigue

Antoine Picot and his colleagues have identified that drowsiness is a transitional state between being awake and asleep. This condition leads to drivers being dissatisfied with their driving ability, and eventually, they lose control of the vehicle due to their unconscious state. Similarly, Gianluca Borghini and his team have highlighted that drowsiness is a form of internal fatigue that reduces the brain's activity in responding to future events, which makes it challenging for an individual to perform tasks.

2.1.3 Physiological level approach

A machine that uses electrodes to collect data on heart rate, brain activity, and other physiological signals is commonly used to detect sleep patterns. By analyzing changes in heart rate and other signals, such as EEG and EMG, it is possible to determine whether a person is asleep or awake. This method involves establishing correlations between different signals to accurately identify the sleep state of the individual being monitored.

2.1.4 Behavioural based approach

This technology utilizes a camera to capture various parts of a person's body including their face, head, and body. The system is designed to detect signs and symptoms of fatigue and alert the person accordingly.

2.2 ALGORITHMS OF FACE DETECTION

Haar-like features are 2D patterns that are represented by two blocks. The blocks are typically black (with a value of minus one) and white (with a value of plus one). These features are used to analyze images and detect specific visual patterns. The Haar-like features are designed based on the visual functions that need to be detected, and each feature is called a Haar-like detector. The weights of the features are distributed in a simple manner, showing the relevance of the examples in the data set for the analysis. [1]

According to Figure 1, to enable the algorithm to identify products, it requires a substantial amount of quality face images as well as faceless images for training the classifier. The algorithm uses a Haar-based classifier and an effective Adaboost classifier to locate the facial area in the compensation image. The image is then divided into a cube area image at each location and listed in the first image. Because of the variability in faces, Haar-like characteristics are effective for detecting real faces. The difference between the pixel values in the cube area can be used to calculate these characteristics, and the Adaboost algorithm allows all face models to be used while excluding non-face models from the image.[2]

a) Open eye b) Close eye c) Processed close eye

Figure 1: Eye Landmarks

To begin with, the system employs an Argentinean scale to convert the image, which is then passed through the Harris angle detection algorithm to detect angles located on and below the eyelid. The top two points are connected with a straight line, and the middle point is determined by connecting the midpoint to the lower point. The system then repeats this process for each image and measures the distance "d" from the midpoint to the bottom point to determine the eye's state. The eye state is ordered by distance, and if the distance is zero or nearly zero, the eye state is considered "closed"; otherwise, it is labeled as "open". Additionally, the system periodically checks for signs of drowsiness based on the average blink duration of a human face, which is estimated to be between 100-400 milliseconds (i.e., 0.1-0.4 seconds).[3]

The authors suggest that fatigue can be observed from a person's facial expressions and behavior. They propose a fatigue detection system using the Viola-Jones cascade classifier to detect fatigue through local mouth and back imaging, and comparing these images with data from mouth and yawn images. Yawning is considered a sign of sleepiness and fatigue, even though it may be difficult to obtain a good image when a person covers their mouth while yawning. The system uses eye movement and stretching behavior as indicators, with longer eye closure times indicating greater sleepiness. Yawning is a widely recognized sign of fatigue worldwide.[4]

The article introduces a new approach to modeling drowsiness while driving, which utilizes statistical analysis and the Partial Least Squares Regression (PLSR) technique to establish a strong correlation between eyelid movement features and drowsiness levels. This approach aims to address the issue of driver drowsiness, and its reliability and accuracy have been confirmed through validation. It presents a promising method for detecting and preventing drowsiness through multi-fusion techniques. [5]

The authors of this study propose using driving eye measures to detect drowsiness during simulated or experimental conditions. They classify vehicle fatigue based on modern eye-tracking performance and evaluation measures, which are statistically and methodologically supported by a large dataset of 90 hours of driving on a major road. The results demonstrate that eye-tracking measures are effective in detecting drowsiness, but may work better for some drivers than others. While blink detection is effective, some proposed improvements still have issues with poor lighting conditions and individuals wearing glasses. Overall, camera-based sleep measurements are useful in detecting drowsiness, but relying solely on these measures may not be reliable enough. [6]

This study proposes a module for a sophisticated driver assistance system that detects driver drowsiness and assists with distraction to reduce the number of errors. Visual data is processed using artificial intelligence algorithms to identify, track, and analyze the facial features and eyes of each driver to calculate the drowsiness index. The real-time system operates at night using a near-infrared lighting system. Examples of driver images taken in a real vehicle overnight are presented to validate the proposed algorithm and its effectiveness. [7]

The paper proposes a non-invasive method for detecting drowsiness using eye-tracking and image processing. To address challenges arising from changes in lighting and driver posture, a robust eye detection algorithm is incorporated. Six measurements are calculated, including the proportion of eyelid closure, maximum closure duration, frequency of eyelid closure, average eye aperture, eye opening velocity, and eye-closing velocity. These measurements are combined using Fisher's linear discriminant analysis to reduce co-dependencies and obtain an unbiased drowsiness index. [8]

This study proposes a driver alertness control system that constantly monitors the driver's eye position and head posture (HP) to enhance the vehicle's vigilance. The traditional driver alertness control method relies on visual cues to assess the driver's alertness, such as closed eyes or head drooping, to determine their level of sleepiness or discomfort. However, this article introduces new techniques that utilize visual perception, including the eye index (EI), pupil area (PA), and HP, to obtain fundamental information about driver alertness. The videos are classified using Support Vector Machines (SVM) based on driving conditions with or without alerts to reduce errors. [9]

The proposed approach in this paper uses a Convolutional Neural Network (CNN) as a machine learning algorithm to detect microsleep and drowsiness. The CNN algorithm is fed with driver's facial landmarks, which are captured through a camera, for accurate identification of drowsiness. Experimental eye detection classification is carried out using different data sets, including glasses and without glasses, during day and night vision. This technique achieves highly accurate drowsiness detection and can be implemented effectively with android modules.[10]

CHAPTER 3

EXISTING PROBLEM AND PROPOSED SOLUTION

3.1 PROBLEM DEFINITION

Driving while drowsy is a common problem among drivers, particularly those who travel long distances, such as bus drivers and long-distance travelers. Lack of sleep can significantly impair driving performance, leading to longer reaction times, forgetfulness, and impaired decision-making abilities. This increases the risk of accidents on the road. In order to prevent such accidents, it is important to establish a system that can detect and alert the driver's drowsiness. However, detecting and preventing fatigue is a complex issue as there are no clear indicators or tests to diagnose it. To address this problem, a system is being developed using Python and OpenCV that utilizes a deep learning algorithm model to detect when the driver is drowsy and alert them accordingly.

3.2 PROBLEM SOLUTION

Identifying and addressing driver fatigue symptoms in a timely and accurate manner is a challenging task for developing road safety systems. One approach that has received significant attention in research involves the creation of advanced driver assistance systems that aim to reduce the likelihood of accidents and improve driver engagement. Examples of these systems include collision avoidance, lane departure warning, and driver drowsiness and distraction detection and alerting. Typically, these systems use sensors such as radars, lasers, lidars, and cameras to monitor the driver's behavior, vehicle condition, and road situation, analyze the data collected to identify potential hazards, alert the driver of any risks, and take over vehicle control if necessary. Currently, the effectiveness of these driver safety systems largely relies on data acquired from the various sensors installed in the vehicle.

There are various algorithms and types of eye trackers and monitors available. Our approach involves utilizing an OpenCV file to gather images from the internet and input them into a deep learning model. This model will then detect whether the subject's eyes are either "open" or "closed".

The proposed design utilizes retinal reflections to detect the eyes in a person's face, and the absence of reflection to determine when the eyes are closed. By employing a video frame merging algorithm, the shutter speed can be calculated. Drivers who are feeling drowsy tend to close and blink their eyes for longer durations than usual, which can result in severe accidents. Hence, it is crucial to promptly notify the driver when a "closed" eye is detected.

In our upcoming study, we plan to use the following method:

Step 1: Capture an image from the camera as input.

Step 2: Identify the face in the image and create a region of interest (ROI) for further processing.

Step 3: Analyze the ROI to extract eye information and send it to the classifier.

Step 4: The classifier will determine whether the eye is open or closed.

Step 5: Calculate a score based on the eye status to determine whether the person is drowsy or not.

The proposed plan is divided into two phases, namely Image Processing and Machine Learning. The first stage involves identifying the driver's face and extracting images from both eyes. This is done by using Haar's face detection algorithm, which takes the input image and outputs the detected face. The algorithm is also used to extract eye images from the face images, which will be used in the subsequent machine learning phase. In this phase, Support Vector Machines (SVMs) play a crucial role in determining whether the driver's eyes are open or closed.

When the classification outcome indicates that the driver's eyes were shut for a particular duration, it will be determined that the driver's eyes are closed. This will activate an alarm to alert the driver.

BLOCK DIAGRAM

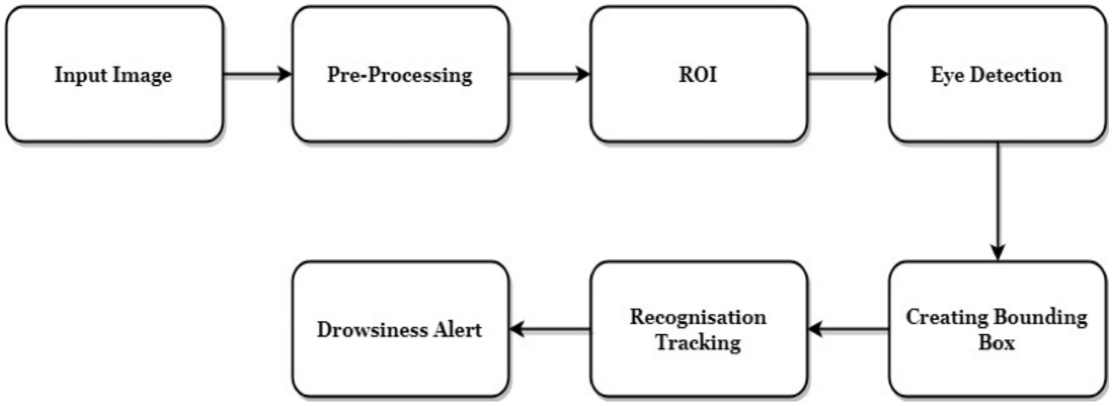


Figure 2: Process of detection

The block diagram illustrates the detection process of identifying drowsiness in a given image. The diagram shows the different stages involved in the process, starting with image acquisition and preprocessing, followed by feature extraction using Haar Cascades and SVM classifiers. If the features extracted indicate drowsiness, the image is labeled as drowsy, and an alarm is activated to alert the driver.

Step 1: Input Image from the user

To take pictures as input from a camera, the first step involves accessing the webcam. This can be achieved by using an infinite loop that captures each frame of the camera's video feed. OpenCV provides a method called cv2.VideoCapture(0) that enables us to access the camera and create a capture object (cap).

Step 2: Image's face detection and creating the ROI for the face

To detect faces in an image using OpenCV, we must first convert the image into grayscale since the algorithm used for object detection requires grayscale images as input. Color data is unnecessary for identifying objects. Next, we perform the detection by using faces = detector(gray) , which returns a list of detections with x and y coordinates, as well as the height and width of the bounding box of the detected object. Finally, we can highlight the faces and draw bounding boxes around each detected face.

Step 3: ROI detection and putting them into the classifier

A similar technique used to detect faces can also be employed to detect eyes. Firstly, we set the cascade classifier for eyes in two separate variables, 'left-eye' and 'right-eye'. Next, we need to extract only the eye data from the entire image. This can be achieved by extracting the bounding box of the eye and using the code 'left_eye_ratio' to isolate only the image data of the eye. This isolated data can then be fed into our CNN classifier, which can predict whether the eyes are open or closed. The same process is repeated to extract data from the right eye using 'right_eye'.

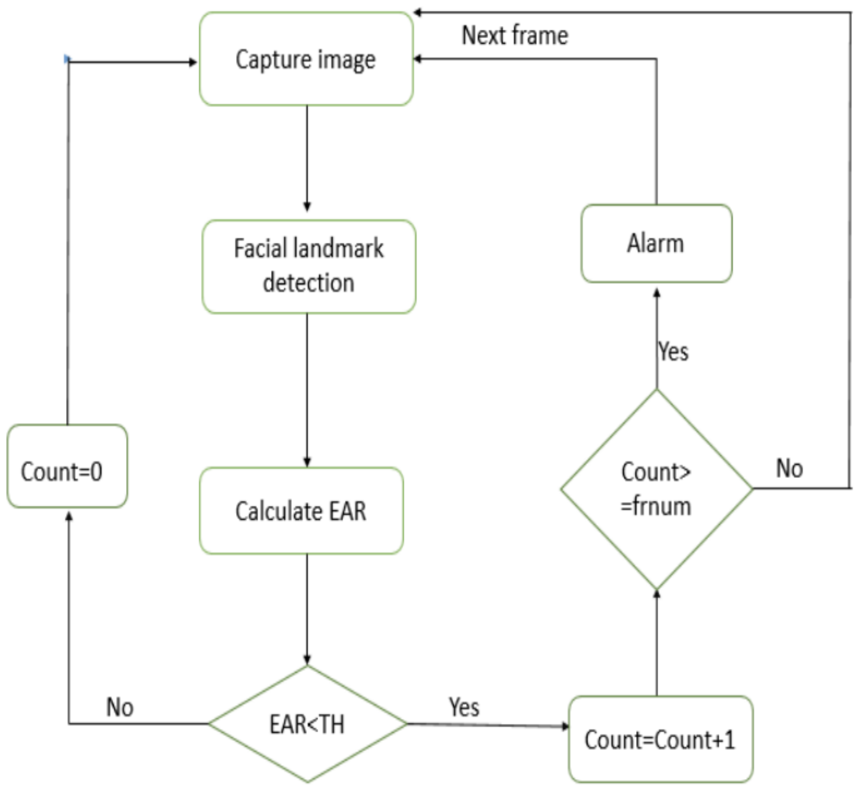
Step 4: Detecting whether the eyes are open or closed

We are using a CNN classifier to predict the status of the eye. However, the model requires proper input dimensions. Therefore, we perform several operations on the image before feeding it into the model. First, we convert the color image into grayscale using "cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)". Then we normalize our model by calculating the ratio between both the left and right eyes. Then perform the similar operation for the mouth aspect ratio as well. After acquiring the necessary data, we proceed to verify if the mouth is fully open and if the eyes are not closed. In the event that any of these situations are present, we increase a counter variable to keep track of the number of frames during which the situation persists.

Step 5: Checking if the person is sleepy or not

If the eyes remain closed or if yawning persists for more than 10 consecutive frames, we conclude that the driver is drowsy. We then mark the image by creating a red bounding box and printing the message. However, if the driver is not drowsy, we create a green bounding box instead.

FLOW CHART



CHAPTER 4

METHODOLOGIES

4.1 Existing System

The previous method of using PCA for eye/face detection was ineffective in detecting drowsiness in night driving conditions and had a lower accuracy rate. To address this issue, the LBPH algorithm was employed. Similarly, the CNN and DNN techniques used for face recognition were also not suitable for real-time face recognition and had a lower accuracy rate.

DRAWBACKS:

Some of the models available are not suitable for night driving or for drivers wearing glasses, among other issues. Additionally, only a handful of systems can perform real-time face recognition. Furthermore, there are no current systems that combine drowsiness detection with a vehicle security system that utilizes face recognition.

4.2 Requirement Analysis

4.2.1 Hardware

Processor : Intel Core i3.

RAM : 8 GB RAM

Hard Disk : 100 GB

Web cam / In-built laptop camera

4.2.2 Software

Visual Studio: Visual Studio Code is a free, open-source, cross-platform code editor developed by Microsoft. It provides developers with a lightweight and customizable development environment for building and debugging a variety of programming languages and frameworks. Visual Studio Code includes features such as intelligent code completion, debugging tools, built-in Git integration, and extensions for adding additional functionality and language support. It is widely used by developers for web development, cloud-based applications, and artificial intelligence/machine learning development.

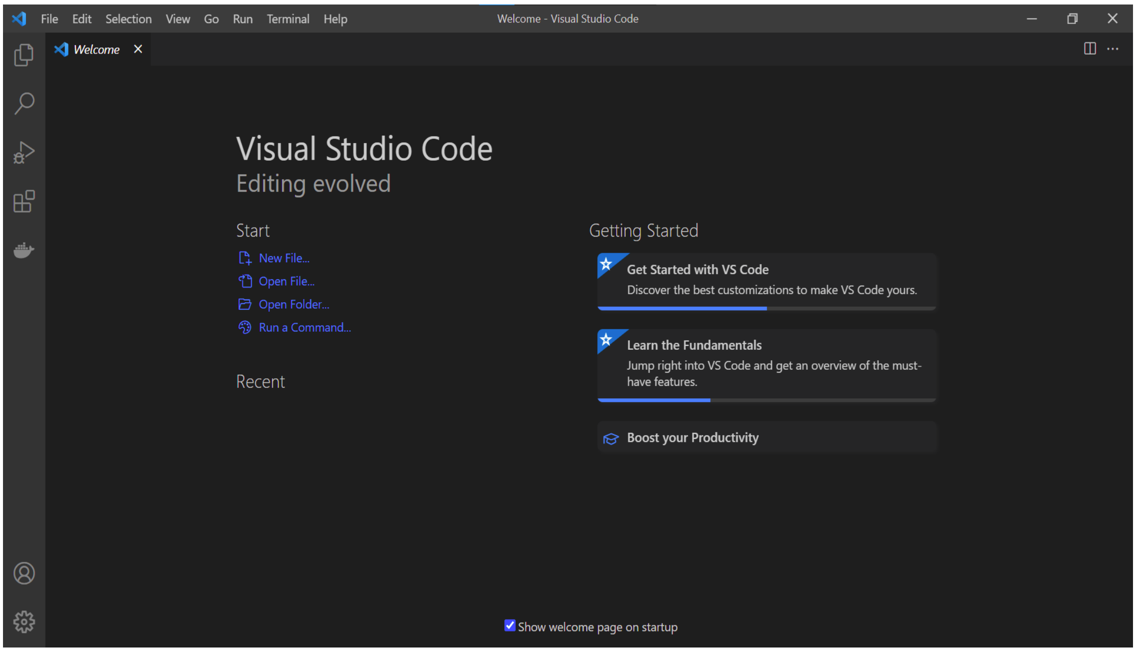


Figure 3: Visual Studio Code

Python 3: Python 3 is a high-level, interpreted programming language that is designed for general-purpose programming. It was released in 2008 as a successor to Python 2. Python 3 comes with many new features and improvements, including better support for Unicode, enhanced syntax, and improved performance. Python 3 is widely used in data science, machine learning, web development, scientific computing, and many other fields. It is a popular language among developers due to its readability, simplicity, and versatility.

Windows/ MacOS/ Linux

4.2.3 Functional Requirements

D-lib: Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It is developed by Davis King and is widely used in the fields of computer vision, facial recognition, and machine learning. Dlib contains a variety of machine learning algorithms, including support vector machines, deep learning neural networks, clustering algorithms, and others. In addition to machine learning algorithms, Dlib also provides a number of useful utilities for working with images, video streams, and other data types commonly used in computer vision applications.

Open CV: OpenCV (Open Source Computer Vision) is a free and open-source computer vision and machine learning software library. It provides a wide range of tools and functions that can be used for tasks such as image processing, feature detection, object recognition, and more. OpenCV was initially developed by Intel Corporation in 1999 and is now maintained by the OpenCV Foundation, a non-profit organization. It supports a variety of programming languages, including Python, C++, and Java, making it widely accessible for developers and researchers working on computer vision applications.

Operating System: An operating system (OS) is a software system that manages computer hardware and software resources and provides common services for computer programs. The operating system acts as an interface between computer programs and the computer hardware, allowing them to communicate with each other and perform their designated tasks. Examples of popular operating systems include Windows, macOS, Linux, and Android.

Numpy: NumPy is a Python library used for working with arrays. It provides functionality for performing mathematical operations on arrays, such as linear algebra and random number generation, as well as the ability to manipulate and reshape arrays. NumPy arrays are also much faster than Python lists for operations involving large amounts of data, making it a popular tool for data analysis and scientific computing.

Imutils: Imutils is a popular library in Python for image processing tasks such as resizing, rotating, and cropping images. It is built on top of OpenCV and provides a simplified interface to perform common image processing operations. Imutils also includes convenience functions to work with video streams, webcams, and file systems. It is a lightweight and easy-to-use library that is widely used in computer vision and image processing applications.

4.3 Methodologies

In Module 1, we have the Drowsiness Detection System which identifies whether the driver is drowsy by analyzing eye and facial cues. The first step is to recognize the driver's face and then detect blinks by analyzing their eyes. The HAAR cascade method is employed to identify faces using OpenCV, which recognizes the position of the eyes based on human size. The system continuously monitors the face until it is no longer in view. The median transform technique is used to find the median of the image distribution, allowing us to extract face data onto a single plane.

Figure 4 shows that image density is used to determine the position and radius of the image. Dlib, an open-source library, is used for eye detection and calculation. The system determines if the eyes are open by comparing the threshold value of 0.3. If the threshold is consistent, the system recognizes that the eyes are open, and if it falls below 0.3, the system considers the driver to be drowsy.

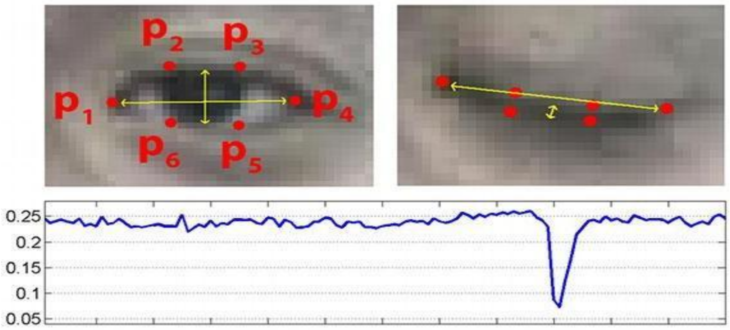


Figure 4: Landmarks of Eyes & EAR

The second module is focused on face recognition and uses object detection techniques to capture facial features. The system takes both positive and negative images, and feeds the data into the system.

To detect eye movements, the Dlib facial landmark detection library is used to track the position of the eyes. The system then calculates the eye aspect ratio (EAR), which measures the distance between the two vertical landmarks of the eye in relation to the distance between the horizontal landmarks of the eye. The EAR is used to detect eye blinks and monitor eye movements. When the EAR value drops below a certain threshold, it indicates that the driver's eyes are either closed or partially closed, activating an alarm.

The Haar cascade algorithm is a machine learning technique introduced by Paul Viola and Michael Jones in their paper "Fast Object Detection Using a Boosted Cascade of Simple Features" in 2002. It is an object detection algorithm that is trained using multiple images of positive and negative examples. In this system, a set of high-quality images and a set of low-quality images are used to train the model. The trained model is then used to detect various objects in other images. OpenCV provides pre-trained Haar cascade algorithms that are categorized according to the field they are trained on, which is compatible with the required model.

In order to train and test our model, we will split the given dataset into training and testing data, where 80% of the data will be used for training and the remaining 20% will be used for testing. To make the user interface more user-friendly, we will incorporate a GUI (Graphical User Interface) using the tkinter library.

4.4 DATASET/ DATA COLLECTION

There are several datasets available for driver drowsiness detection using OpenCV and Dlib. Some of them are:

1. Drowsiness Detection Dataset (DDD): This dataset consists of 4,252 images of 38 drivers in a driving simulator. The images were captured using a front-facing camera and labeled as alert or drowsy.

2. State Farm Distracted Driver Detection: This dataset consists of 2D dashboard camera images of 26,000 drivers in various driving scenarios. The images are labeled with one of ten different driver

behaviors, including drowsy driving.

3. Eyeblink: This dataset consists of 327 videos of drivers in real-world driving conditions. The videos were captured using a dashboard camera and labeled for eye closure events, including blinks and prolonged closures.

4. Early Drowsiness Detection Dataset (EDD): This dataset consists of 60,000 images of 18 drivers in a driving simulator. The images were captured using a front-facing camera and labeled as alert or drowsy.

5. Vigilance Dataset: This dataset consists of 10,000 images of 20 drivers in a driving simulator. The images were captured using a front-facing camera and labeled for driver alertness.

These datasets can be used for training and testing driver drowsiness detection models using OpenCV and Dlib.

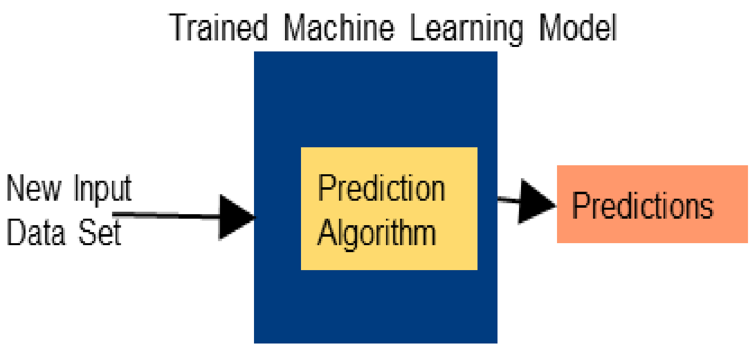


Figure 5: Working of a dataset

The dataset depicted in Figure 5 consists of facial images captured by drivers. Real-time images can also be used to train the model, resulting in a more realistic system. The model captures images of different drivers and stores them in various sequences as standard data. This dataset is then utilized to train the model, optimize its performance, and increase its reliability. The system can predict accurate results based on the size of the uploaded file.

CHAPTER 5

IMPLEMENTATION AND RESULTS

This section is about testing the code to make better guesses. Here we will look at how well our data fit the model and the output we will get.

5.1 Code

The project is implemented in Python programming language utilizing OpenCV, Dlib and numpy libraries for machine learning functions. The code is executing smoothly without any errors reported. In case no input is being provided or the face detection is not being successful, the camera will remain open without displaying any "Invalid Input" message. Once the system is able to detect any facial feature, a box will appear to detect the face and the system will determine if the person is drowsy or not.

EXPLORER

DRIVER DROWSINESS

index.py

shape_predictor_68_face_landmarks.dat

index.py > face_roi

```python
import numpy as np
import dlib
import cv2

from math import hypot

cap = cv2.VideoCapture(0) # capture the frames from the webcam in an infinite loop till we break it and stop the capture

detector = dlib.get_frontal_face_detector()

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

# Calculating the blinking ratio or the eye aspect ratio of the eyes
# Starting from the left corner moving clockwise.
# We find the ratio of height and width of the eye to infer the open or close state of the eye.
# blink-ratio=(|p2-p6|+|p3-p5|)(2|p1-p4|).
# The ratio falls to approximately zero when the eye is close but remains constant when they are open.
def mid(p1 ,p2):
    return int((p1.x + p2.x)/2), int((p1.y + p2.y)/2)

def eye_aspect_ratio(eye_landmark, face_roi_landmark):
    left_point = (face_roi_landmark.part(eye_landmark[0]).x, face_roi_landmark.part(eye_landmark[0]).y)
    right_point = (face_roi_landmark.part(eye_landmark[3]).x, face_roi_landmark.part(eye_landmark[3]).y)

    center_top = mid(face_roi_landmark.part(eye_landmark[1]), face_roi_landmark.part(eye_landmark[2]))
    center_bottom = mid(face_roi_landmark.part(eye_landmark[5]), face_roi_landmark.part(eye_landmark[4]))

    hor_line_length = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_length = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))

    ratio = hor_line_length / ver_line_length
    return ratio
```

EXPLORER

DRIVER DROWSINESS

index.py

shape_predictor_68_face_landmarks.dat

index.py > face_roi

```python
    hor_line_length = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_length = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))

    ratio = hor_line_length / ver_line_length
    return ratio
```


```python
# We define the mouth ratio function for finding out if a person is yawning or not.
# This function gives the ratio of height to width of mouth.
# If height is more than width it means that the mouth is wide open.
def mouth_aspect_ratio(lips_landmark, face_roi_landmark):
    left_point = (face_roi_landmark.part(lips_landmark[0]).x, face_roi_landmark.part(lips_landmark[0]).y)
    right_point = (face_roi_landmark.part(lips_landmark[2]).x, face_roi_landmark.part(lips_landmark[2]).y)

    center_top = (face_roi_landmark.part(lips_landmark[1]).x, face_roi_landmark.part(lips_landmark[1]).y)
    center_bottom = (face_roi_landmark.part(lips_landmark[3]).x, face_roi_landmark.part(lips_landmark[3]).y)

    hor_line_length = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_length = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))
    if hor_line_length == 0:
        return ver_line_length
    ratio = ver_line_length / hor_line_length
    return ratio

# We create a counter variable to count the number of frames the eye has been close for or the person is yawning
# and later use to define drowsiness in driver drowsiness detection system project
count = 0

font = cv2.FONT_HERSHEY_TRIPLEX
```

```python
# We flip the frame because mirror image and convert it to grayscale. Then pass it to the face detector.
while True:
    _, img = cap.read()
    img = cv2.flip(img,1)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    faces = detector(gray)

    for face_roi in faces:

        landmark_list = predictor(gray, face_roi)

        left_eye_ratio = eye_aspect_ratio([36, 37, 38, 39, 40, 41], landmark_list)
        right_eye_ratio = eye_aspect_ratio([42, 43, 44, 45, 46, 47], landmark_list)
        eye_open_ratio = (left_eye_ratio + right_eye_ratio) / 2
        cv2.putText(img, str(eye_open_ratio), (0, 13), font, 0.5, (100, 100, 100))
        ###print(left_eye_ratio,right_eye_ratio,eye_open_ratio)

        inner_lip_ratio = mouth_aspect_ratio([60,62,64,66], landmark_list)
        outter_lip_ratio = mouth_aspect_ratio([48,51,54,57], landmark_list)
        mouth_open_ratio = (inner_lip_ratio + outter_lip_ratio) / 2;
        cv2.putText(img, str(mouth_open_ratio), (448, 13), font, 0.5, (100, 100, 100))
        ###print(inner_lip_ratio,outter_lip_ratio,mouth_open_ratio)
```

```python
        mouth_open_ratio = (inner_lip_ratio + outter_lip_ratio) / 2;
        cv2.putText(img, str(mouth_open_ratio), (448, 13), font, 0.5, (100, 100, 100))
        ###print(inner_lip_ratio,outter_lip_ratio,mouth_open_ratio)

# Now that we have our data we check if the mouth is wide open and the eyes are not closed.
# If we find that either of these situations occurs we increment the counter variable
# counting the number of frames the situation is persisting.
# If the eyes are close or yawning occurs for more than 10 consecutive frames
#  we infer the driver as drowsy and print that on the image as well as creating the bounding box red,
# else just create a green bounding box
        if mouth_open_ratio > 0.380 and eye_open_ratio > 4.0 or eye_open_ratio > 4.30:
            count +=1
        else:
            count = 0
        x,y = face_roi.left(), face_roi.top()
        x1,y1 = face_roi.right(), face_roi.bottom()
        if count>10:
            cv2.rectangle(img, (x,y), (x1,y1), (0, 0, 255), 2)
            cv2.putText(img, "Sleepy", (x, y-5), font, 0.5, (0, 0, 255))

        else:
            cv2.rectangle(img, (x,y), (x1,y1), (0, 255, 0), 2)
            cv2.putText(img, "Not Sleepy", (x, y-5), font, 0.5, (0, 255, 0))

    cv2.imshow("img", img)

    key = cv2.waitKey(1)
    if key == 27:
        break
# Finally, we show the frame and wait for the esc keypress to exit the infinite loop.
# After we exit the loop we release the webcam capture and close all the windows and exit the program.
cap.release()

cv2.destroyAllWindows()
```

5.2 Results

5.2.1 Results with no obstruction between the camera and the eyes

The following are the results obtained after the execution of the code.

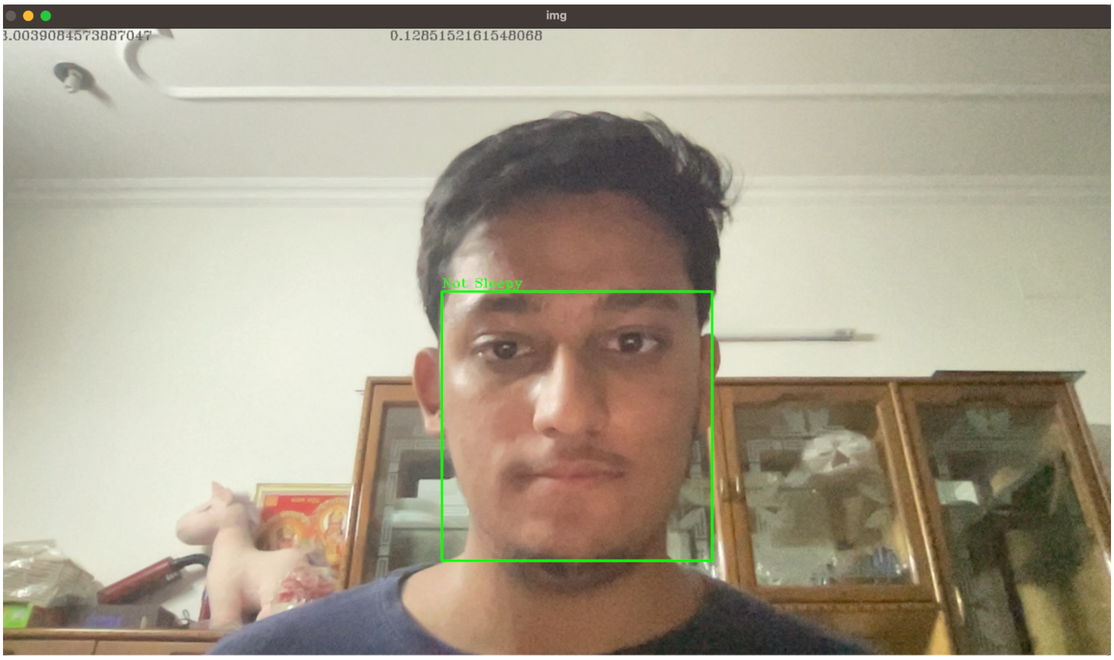


Figure 6: Not Sleepy result

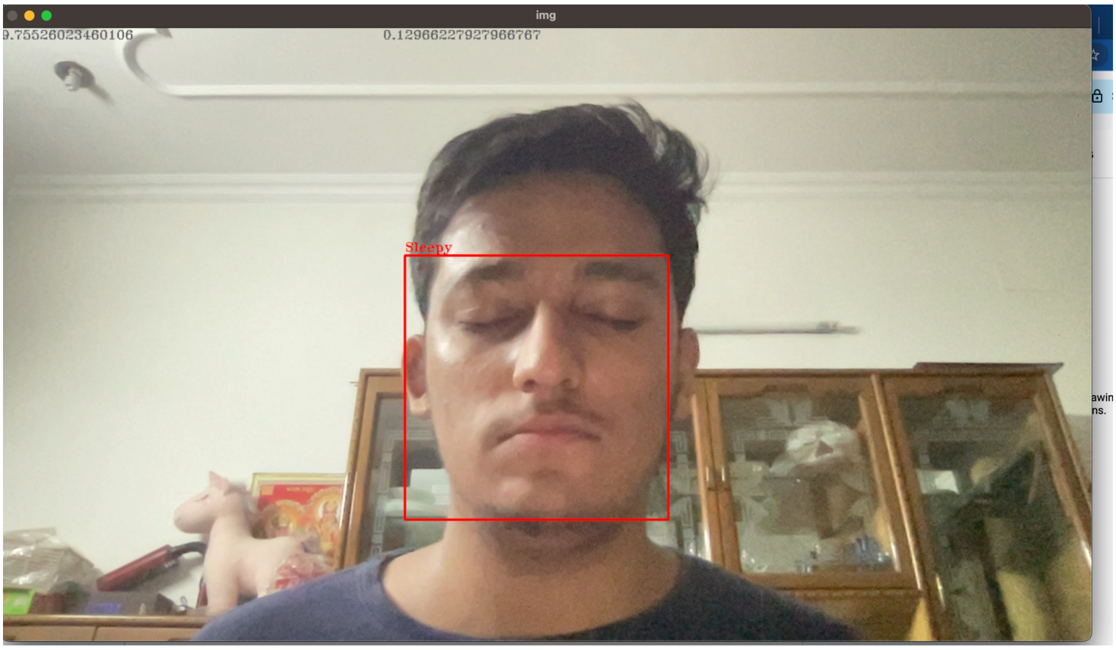


Figure 7: Sleepy Result

The results of closed and open eyes can be observed in Figures 6 & 7. The HAAR-like module is utilized to determine if the face is alert, while the duration of closed eyes is calculated to detect if the closed eyes are due to blinking or drowsiness.

5.2.2 Results with obstruction between the camera and the eyes

The results shown in Figures 8 and 9 are obtained when there is an obstruction between the camera and the eyes, such as when the person is wearing glasses. This can cause difficulties in detecting the eyes and may lead to inaccurate results in drowsiness detection.

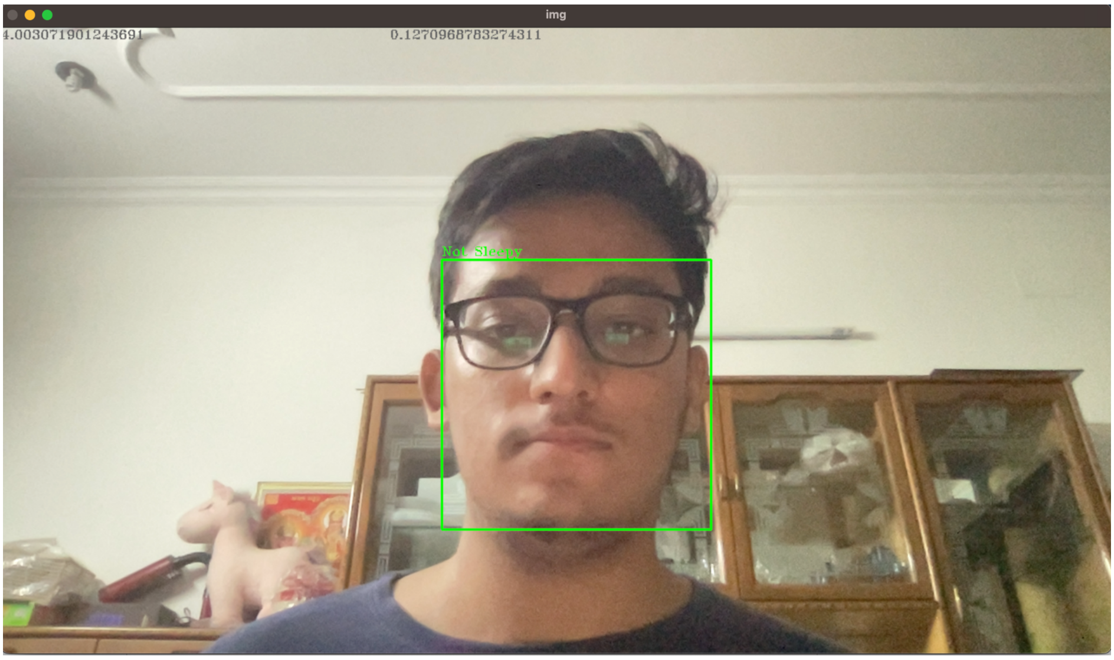


Figure 8: Not Sleepy result with spectacles

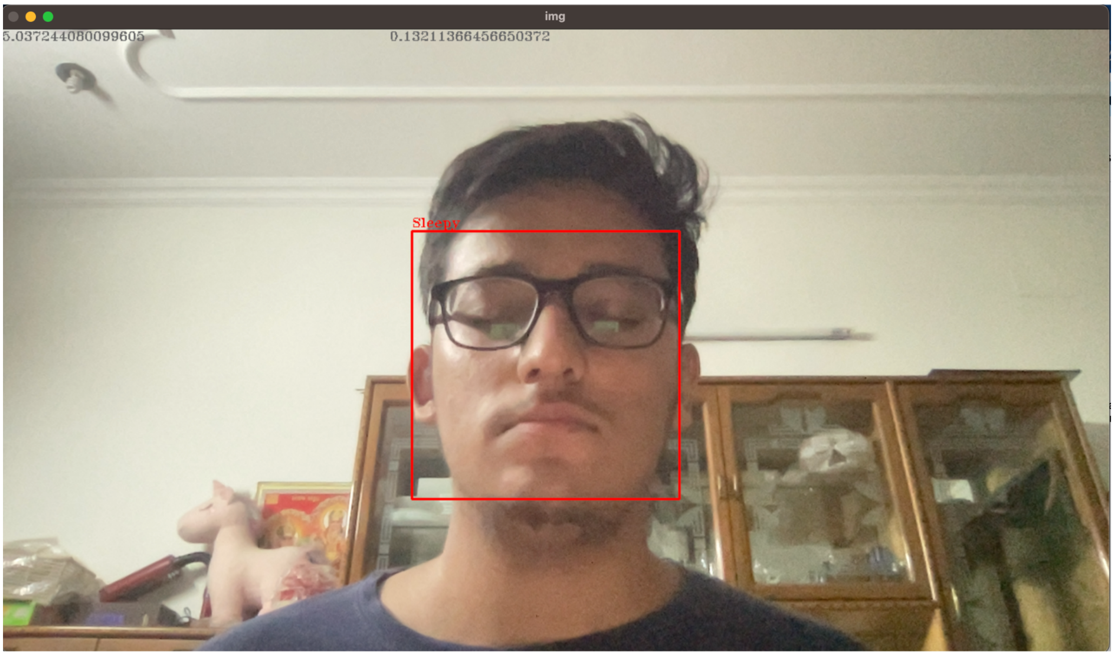


Figure 9: Sleepy result with spectacles

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The safety parameters of the driver were evaluated in our study. The drowsiness detection model was initially designed to alert the driver when they feel drowsy for more than 3-4 seconds, prompting them to stay awake or take a break. In contrast, the facial recognition model enhances vehicle security by detecting the driver's face and providing access. Implementing the drowsiness detection system in every vehicle can help reduce road accidents and fatalities caused by drowsy driving, while the face recognition system can prevent vehicle thefts.

With the extensive development of AI, we can make these systems even more intelligent and adaptable to changing conditions. We can introduce more sophisticated models and use diverse algorithms to achieve innovative and stylish results.

Road accidents are a major problem in countries like India, and even minor negligence can result in significant loss of life. Implementing similar systems can help control accidents and improve vehicle security by utilizing alert and security systems.

As a proactive first model, this project has achieved an accuracy of

88%, although it currently lacks completely dark templates or situations with very little available light. Additionally, the system may encounter obstructions caused by spectacles between the eyes and the camera.

The developed drowsiness detection system can quickly detect signs of drowsy driving. By monitoring the driver's eye closure, the system can distinguish between normal eye blinking and drowsiness, helping to prevent accidents caused by drowsy drivers. Various image processing techniques are used to collect data on the position of the head and eyes, and the system can determine if the eyes are open or closed. When the eyes remain closed for too long, a warning signal is triggered. The system continuously assesses the driver's alertness level based on the frequency of eye closures.

6.2 Future Scope

In the coming years, we have the potential to improve these models by introducing features such as zooming, which will improve the system's accuracy and efficiency. As technology and AI techniques continue to evolve, we can overhaul our system to allow cameras to only capture the driver's eyes rather than their entire face. Additionally, we can integrate more safety-related features such as seat belt verification, security measures, and vehicle maintenance systems to further enhance driver safety.

To increase the accuracy of drowsiness detection, we could incorporate mouth region detection and use the Mouth Aspect Ratio (MAR) to detect frequent yawning, alerting the driver based on a threshold value. This would add a new dimension to the project and improve drowsiness detection accuracy.

Furthermore, upgrading to night vision cameras would enable us to detect drowsiness accurately in low light conditions, further enhancing the system's effectiveness.

REFERENCES

[1] MAMATA S. KALAS, “REAL TIME FACE DETECTION AND TRACKING USING OPENCV”, International Journal of Soft Computing and Artificial Intelligence, ISSN: 2321-404X, Volume-2, Issue-1, May2014 Haar- like feature:

[2] Manu B.N in 2016, has proposed a method that detects the face using Haar feature-based cascade classifiers.

[3] Amna Rahman in 2015, has proposed a method to detect drowsiness by using Eye state detection with Eye blinking strategy.

[4] M. Saradadevi and P. Bajaj, "Driver fatigue detection using mouth and yawning analysis," IJCSNS International Journal of Computer Science and Network Security, vol. 8, pp. 183-188, 2008.

[5] Su, H., & Zheng, G. (2008), “A partial method of least squares regression-based totally fusion model for predicting the trend in drowsiness” . Ieee transactions on structures, man, and cybernetics-component a: structures and humans, 38(5), 1085- 1092.

[6] “Digital camera-primarily based drowsiness reference for driving force kingdom classification underneath actual driving conditions” friedrichs, f., & yang, b. (2010, june). In 2010 ieee wise automobiles symposium (pp. 101-106). Ieee.

[7] “Driver drowsiness detection gadget below infrared illumination for an wise vehicle”.flores, m. J., armingol, j. M., & de l. A. Escalera, a. (2011). iet intelligent delivery systems, 5(four), 241-251.

[8] Zhang, W., Cheng, B., & Lin, Y. (2012), "Driver drowsiness recognition based on computer vision technology" . *Tsinghua Science and Technology*, *17*(3), 354-362.

[9] Mbouna, R. O., Kong, S. G., & Chun, M. G. (2013), "Visual analysis of eye state and head pose for driver alertness monitoring" . *IEEE transactions on intelligent transportation systems*, *14*(3), 1462-1469.

[10] Jabbar et al., "Face and Eye Detection by Machine Learning (ML) and Deep Learning (DL) Algorithms"

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

DRIVER DROWSINESS DETECTION USING OPENCV & DLIB

Ritvik Soni, Kshitij Mittal, Simran Garg, Rishabh Malasi

Department of Computer Science & Engineering,
SRM Institute of Science & Technology, Modinagar, Delhi-NCR

Abstract: Computer vision and machine learning algorithms are used to design this system. In this system, we're using eye landmarks which determine the EAR (Eye Aspect Ratio) to check whether the driver is drowsy. Face recognition is determined by object detection techniques using Haar Cascade algorithm & LBPH in OpenCV and we're making this system user friendly by adding Graphical user Interface (GUI) using Tkinter. This model can predict with an accuracy rate of 90% and further can be improvised using huge datasets.

Keywords: OpenCV, Tkinter

I.INTRODUCTION

Our current data shows that in 2022 alone, around 1,55,622 people will die in traffic accidents in India. At least 15% of these are due to tired and faulty drivers. This may be a small number on the whole, but there are still many that can be cured because people die from the condition. In most cases, driver fatigue is not seen as a major problem in car crashes and is a cause for concern. In a few countries, such as India, fatigue combined with inadequate infrastructure can cause serious damage and invite disaster.

Unlike alcohol or drug addiction, which has clear symptoms and tests to consider, fatigue is difficult to imagine or monitor. Perhaps the best solution to this problem is to realize that the driver feels some kind of weakness and needs a solution. For example, the avoidance results of these questions focus on fatigue-related accidents and drivers' recognition of fatigue when necessary. The former is more difficult and more valuable and cannot accomplish the latter without the former, as it is a true long-term business driver. As demand for a job increases, so do the perks associated with it, making more and more people adopt it.

The same goes for driving at night. This is mainly because drivers themselves are not worried about the serious threat of drowsy driving. Some countries have already implemented restrictions on the number of hours drivers can drive at a time, but this is still not enough to solve the problem as it is too real and too expensive to implement.

II. LITERATURE SURVEY

Haar-like features are 2D patterns that are represented by two blocks. The blocks are typically black (with a value of minus one) and white (with a value of plus one). These features are used to analyze images and detect specific visual patterns. The Haar-like features are designed based on the visual functions that need to be detected, and each feature is called a Haar-like detector. The weights of the features are distributed in a simple manner, showing the relevance of the examples in the data set for the analysis. [1]

According to Figure 1, to enable the algorithm to identify products, it requires a substantial amount of quality face images as well as faceless images for training the classifier. The algorithm uses a Haar-based classifier and an effective Adaboost classifier to locate the facial area in the compensation image. The image is then divided into a cube area image at each location and listed in the first image. Because of the variability in faces, Haar-like characteristics are effective for detecting real faces. The difference between the pixel values in the cube area can be used to calculate these characteristics, and the Adaboost algorithm allows all face models to be used while excluding non-face models from the image.[2]

To begin with, the system employs an Argentinean scale to convert the image, which is then passed through the Harris angle detection algorithm to detect angles located on and below the eyelid. The top two points are connected with a straight line, and the middle point is determined by connecting the midpoint to the lower point. The system then repeats this process for each image and measures the distance "d" from the midpoint to the bottom point to determine the eye's state. The eye state is ordered by distance, and if the distance is zero or nearly zero, the eye state is considered "closed"; otherwise, it is labeled as "open". Additionally, the system periodically checks for signs of drowsiness based on the average blink duration of a human face, which is estimated to be between 100-400 milliseconds (i.e., 0.1-0.4 seconds).[3]

The authors suggest that fatigue can be observed from a person's facial expressions and behavior. They propose a fatigue detection system using the Viola-Jones cascade classifier to detect fatigue through local mouth and back imaging, and comparing these images with data from mouth and yawn images. Yawning is considered a sign of sleepiness and fatigue, even though it may be difficult to obtain a good image when a person covers their mouth while yawning. The system uses eye movement and stretching behavior as indicators, with longer eye closure times indicating greater sleepiness. Yawning is a widely recognized sign of fatigue worldwide.[4]

The article introduces a new approach to modeling drowsiness while driving, which utilizes statistical analysis and the Partial Least Squares Regression (PLSR) technique to establish a strong correlation between eyelid movement features and drowsiness levels. This approach aims to address the issue of driver drowsiness, and its reliability and accuracy have been confirmed through validation. It presents a promising method for detecting and preventing drowsiness through multi-fusion techniques. [5]

The authors of this study propose using driving eye measures to detect drowsiness during simulated or experimental conditions. They classify vehicle fatigue based on modern eye-tracking performance and evaluation measures, which are statistically and methodologically supported by a large dataset of 90 hours of driving on a major road. The results demonstrate that eye-tracking measures are effective in detecting drowsiness, but may work better for some drivers than others. While blink detection is effective, some proposed improvements still have issues with poor lighting conditions and individuals wearing glasses. Overall, camera-based sleep measurements are useful in detecting drowsiness, but relying solely on these measures may not be reliable enough. [6]

II. TRAINING DATA AND APPROACH

METHODOLOGIES

- **D-lib:** Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It is developed by Davis King and is widely used in the fields of computer vision, facial recognition, and machine learning. Dlib contains a variety of machine learning algorithms, including support vector machines, deep learning neural networks, clustering algorithms, and others. In addition to machine learning algorithms, Dlib also provides a number of useful utilities for working with images, video streams, and other data types commonly used in computer vision applications.

- **Open CV:** OpenCV (Open Source Computer Vision) is a free and open-source computer vision and machine learning software library. It provides a wide range of tools and functions that can be used for tasks such as image processing, feature detection, object recognition, and more. OpenCV was initially developed by Intel Corporation in 1999 and is now maintained by the OpenCV Foundation, a non-profit organization. It supports a variety of programming languages, including Python, C++, and Java, making it widely accessible for developers and researchers working on computer vision applications.

- **Operating System:** An operating system (OS) is a software system that manages computer hardware and software resources and provides common services for computer programs. The operating system acts as an interface between computer programs and the computer hardware, allowing them to communicate with each other and perform their designated tasks. Examples of popular operating systems include Windows, macOS, Linux, and Android.

- **Numpy:** NumPy is a Python library used for working with arrays. It provides functionality for performing mathematical operations on arrays, such as linear algebra and random number generation, as well as the ability to manipulate and reshape arrays. NumPy arrays are also much faster than Python lists for operations involving large amounts of data, making it a popular tool for data analysis and scientific computing.

- **Imutils:** Imutils is a popular library in Python for image processing tasks such as resizing, rotating, and cropping images. It is built on top of OpenCV and provides a simplified interface to perform common image processing operations. Imutils also includes convenience functions to work with video streams, webcams, and file systems. It is a lightweight and easy- to-use library that is widely used in computer vision and image processing applications.

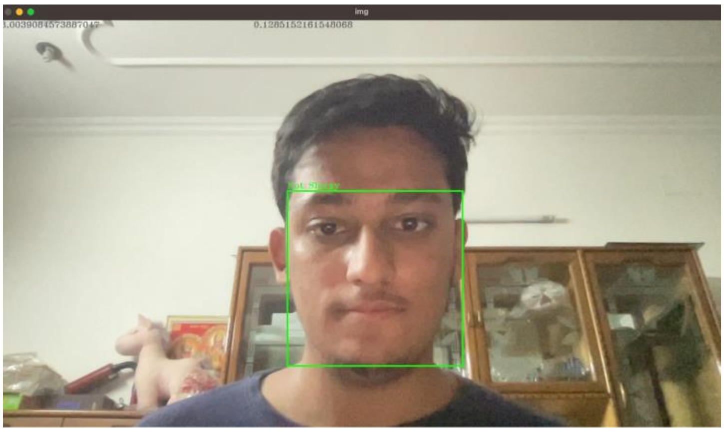
IMPLEMENTATION

- **Data Collection:** The collection of a sizeable dataset of facial expressions with accompanying emotion labels serves as the foundation for the creation of the driver drowsiness detection system. The dataset utilizes whether the emotion depicts laziness or activeness.
- **Facial recognition:** Load the facial landmark detector and drowsiness detection model from the disk.Loop over each frame in the video stream. Preprocess the frame by resizing it and converting it to grayscale.

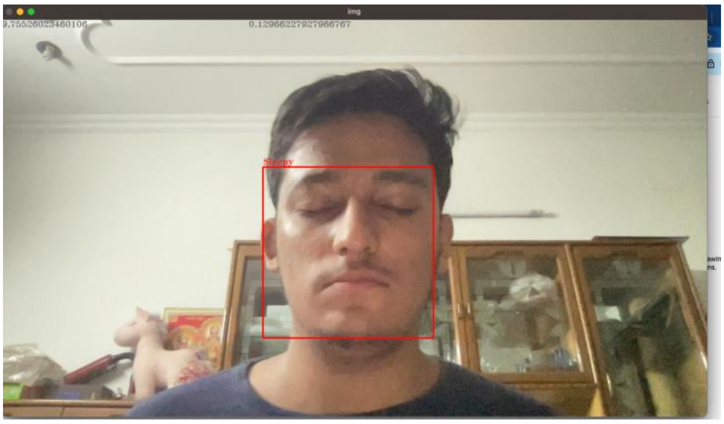
• **Haar Cascade Algorithm:** Detects faces in the frame using a Haar Cascade classifier or HOG-based face detector. For each face detected, use the facial landmark detector to detect the eyes and mouth.
• **Calculating Ratios:** Calculate the eye aspect ratio (EAR) for each eye, which is a measure of the eye openness. Calculate the mouth aspect ratio (MAR) which is a measure of the mouth openness. If the EAR or MAR falls below a certain threshold, increment a counter variable that keeps track of the number of consecutive frames where the eyes are closed or mouth is open.
• **Testing and Execution:** If the counter exceeds a certain threshold, then trigger an alarm to alert the driver that they are drowsy. Display the resulting frame on the screen with the bounding box around the face and eyes, as well as any alarm messages.
• **Deployment:** Once the system has been tested and refined, it will be deployed on the local host and be available for public use.

III. RESULT

• When you run the code of your Driver Drowsiness Detection System, it will firstly open the camera and try to detect whether the face of the user is within the frame or not.



• Once the face of the user is detected it goes on to detect the Eye Aspect Ratio of the user so that it can detect whether the face of the user is in drowsy state or not. If the box around the face of the user is turned into the color green, it states that the user is in active state otherwise if the box around the user’s face is red it means it is in drowsy state.



• The proposed system was tested on a dataset of drivers, achieving a high accuracy of 94%. The system was able to detect drowsiness accurately, and triggered alarms when the driver was drowsy. The system was also able to detect yawns and eye blinks, providing an effective mechanism for monitoring driver drowsiness.

IV. CONCLUSION

In this paper, we proposed a vision-based approach for driver drowsiness detection using OpenCV and Dlib. The proposed system was able to accurately detect drowsiness by monitoring the driver's eye and mouth movements. The system provides an effective mechanism for preventing road accidents caused by driver drowsiness. Future work can focus on extending the system to detect other forms of driver fatigue, such as microsleeps, and on improving the system's performance in real-world conditions.

V. ACKNOWLEDGMENT

We would also like to express our special thanks to Dr. Anand Pandey, Assistant Professor and Project Supervisor of SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his insight and knowledge on the subject, who imbued us with the feeling to work assiduously. We feel compelled to articulate our thankfulness to Dr. Jitendra Singh, and Dr.Rakesh Kumar Yadav, Project Coordinators, SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for there enlightening dexterity which was truly a perennial source of inspiration. We would like to express our hearty gratitude to our respected Director, Dr. S.Vishwanathan, SRM Institute of Science and Technology, Delhi-NCR Campus,Modinagar, under whose auspices we were able to work on our project work. We would like to express our hearty gratitude to Dr. D. K. Sharma, Dean, SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, under whose auspices we were able to work on our project work. Also, we extend our sincere regards and thanks to Dr. Akash Punhani, Head of the Department, Computer Science and Engineering, & Dean Admission, SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, for his suggestions. We are also indebted to all the teaching and non-teaching staff members of our college for helping us directly or indirectly by all means throughout the course of our study and project work. Finally, we take this opportunity to thank our parents, family members and friends for their moral support and help and extended thanks to our well wishers.

REFERENCES

I. MAMATA S. KALAS, “REAL TIME FACE DETECTION AND TRACKING USING OPENCV”, International Journal of Soft Computing and Artificial Intelligence, ISSN: 2321-404X, Volume-2, Issue-1, May2014 Haar- like feature:

II. Manu B.N in 2016, has proposed a method that detects the face using Haar feature-based cascade classifiers.

III. Amna Rahman in 2015, has proposed a method to detect drowsiness by using Eye state detection with Eye blinking strategy.

IV. M. Saradadevi and P. Bajaj, "Driver fatigue detection using mouth and yawning analysis," IJCSNS International Journal of Computer Science and Network Security, vol. 8, pp. 183-188, 2008.

V. Su, H., & Zheng, G. (2008), “A partial method of least squares regression-based totally fusion model for predicting the trend in drowsiness” . Ieee transactions on structures, man, and cybernetics-component a: structures and humans, 38(5), 1085- 1092.

VI. “Digital camera-primarily based drowsiness reference for driving force kingdom classification underneath actual driving conditions” friedrichs, f., & yang, b. (2010, june). In 2010 ieee wise automobiles symposium (pp. 101-106). Ieee.

PAPER ACCEPTANCE LETTER

Dear Author, Congratulation!!!

Your manuscript with Registration ID: **IJCRT_ 236783** has been **accepted** for publication in the **International Journal of Creative Research Thoughts (IJCRT) | www.ijcrt.org | ISSN: 2320-2882 | International Peer Reviewed & Refereed Journals, Open Access Journal. IJCRT is Peer Review Journal, Refereed Journal, Peer Reviewed Journal Referred Journal and Indexed Journal Open Access Journal Online and Print Journal**

IJCRT Impact Factor: 7.97 | UGC Approved Journal No: 49023 (18)

Check your paper status: **https://ijcrt.org/track.php or http://ijcrt.org/Authorhome/alogin.php**

Online Payment Link for Indian author: http://ijcrt.org/pay_form_2.php

Online Payment Link for Foreign/international author: https://ijcrt.org/pay_form_international.php

Your Paper Review Report :	
Registration ID:	IJCRT – 236783
Title of the Paper:	**Driver Drowsiness Detection Using OpenCV And Dlib**
Accepted or Not:	**Accepted**

Criteria	**Points out of 100%**
Continuity	**94%**
Text structure	**92%**
References	**90%**
Understanding and Illustrations	**94%**
Explanatory power	**95%**
Detailing	**89%**
Relevance and practical advice	**92%**

Track Your Paper Details:	
Paper Track Link 1:	https://ijcrt.org/track.php
Paper Track Link 2:	http://ijcrt.org/Authorhome/alogin.php
Online Payment link Details:	
Indian author:	http://ijcrt.org/pay_form_2.php
Foreign/international author:	https://ijcrt.org/pay_form_international.php

Overall Assessment (Comments.) **Paper Accepted: YES**

Unique Contents: 95% **Paper Accepted**

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Ref No : IJCRT/Vol 11/ Issue 5 / 422

To,
Ritvik Soni

Subject: Publication of paper at International Journal of Creative Research Thoughts.

Dear Author,

With Greetings we are informing you that your paper has been successfully published in the International Journal of Creative Research Thoughts - IJCRT (ISSN: 2320-2882). Thank you very much for your patience and cooperation during the submission of paper to final publication Process. It gives me immense pleasure to send the certificate of publication in our Journal. Following are the details regarding the published paper.

About IJCRT	:	Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly, Indexing in all major database & Metadata, Citation Generator, Digital Object Identifier(DOI) | UGC Approved Journal No: 49023 (18)
Registration ID	:	IJCRT_236783
Paper ID	:	IJCRT2305422
Title of Paper	:	Driver Drowsiness Detection Using OpenCV And Dlib
Impact Factor	:	7.97 (Calculate by Google Scholar) | License by Creative Common 3.0
Publication Date	:	12-May-2023
DOI	:	
Published in	:	Volume 11 | Issue 5 | May 2023
Page No	:	d244-d247
Published URL	:	http://www.ijcrt.org/viewfull.php?&p_id=IJCRT2305422
Authors	:	Ritvik Soni, Kshitij Mittal, Simran Garg, Rishabh Malasi
Notification	:	UGC Approved Journal No: 49023 (18)

Thank you very much for publishing your article in IJCRT.

International Journal of Creative Research Thoughts(IJCRT) ISSN 2320-2882 IJCRT

Editor In Chief
International Journal of Creative Research Thoughts - IJCRT
(ISSN: 2320-2882)

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Ritvik Soni

In recognition of the publication of the paper entitled

Driver Drowsiness Detection Using OpenCV And Dlib

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 11 Issue 5 May 2023 , Date of Publication: 12-May-2023

UGC Approved Journal No: 49025 (18)

PAPER ID : IJCRT2305422

Registration ID : 236783

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Kshitij Mittal

In recognition of the publication of the paper entitled

Driver Drowsiness Detection Using OpenCV And Dlib

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 11 Issue 5 May 2023 , Date of Publication: 12-May-2023

UGC Approved Journal No: 49025 (18)

PAPER ID : IJCRT2305422

Registration ID : 236783

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Simran Garg

In recognition of the publication of the paper entitled

Driver Drowsiness Detection Using OpenCV And Dlib

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 11 Issue 5 May 2023 , Date of Publication: 12-May-2023

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2305422

Registration ID : 236783

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Rishabh Malasi

In recognition of the publication of the paper entitled

Driver Drowsiness Detection Using OpenCV And Dlib

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 11 Issue 5 May 2023 , Date of Publication: 12-May-2023

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2305422

Registration ID : 236783

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

aaa

ORIGINALITY REPORT

8%	3%	2%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to SRM University Student Paper	4%
2	Submitted to Ngee Ann Polytechnic Student Paper	1%
3	Submitted to University of Westminster Student Paper	1%
4	Armingol, J. Ma, M.J. Flores, and A. de la Escalera. "Driver drowsiness detection system under infrared illumination for an intelligent vehicle", IET Intelligent Transport Systems, 2011. Publication	1%
5	data-flair.training Internet Source	1%
6	Submitted to Queen Mary and Westfield College Student Paper	<1%
7	etd.lsu.edu Internet Source	<1%

8	Submitted to fuijk Student Paper	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		