

e-ISSN:

www.ijprems.com editor@ijprems.com

DRIVER'S DROWSINESS DETECTION USING MACHINE LEARNING

Mr. D.Krishna¹, Palley Chetana Reddy², Saleem Pasha³, Perla Rakesh⁴

¹Assistant.Professor CSE Dept ACE Engineering College Hyderabad, India. ^{2,3,4}Student CSE ACE Engineering College Hyderabad, India.

ABSTRACT

This project involves detecting and alerting the driver/ co-passengers if the driver is drowsy. 41% of road accidents in India are caused because the driver feels drowsy when he/she is driving the vehicle, to avoid this problem our system takes the driver's image or video (frame by frame) as input and using the appropriate model classifies the driver's eye position as sleepy or not sleepy, if the system detects that the driver is drowsy an alarm is set off notifying the driver/ the co-passengers. The system records the videos and detects the driver's face in every frame by employing image processing techniques. The system can detect facial landmarks, calculates Eye Aspect Ratio (EAR) and Eye Closure Ratio (ECR) to detect driver's drowsiness based on adaptive thresholding. The proposed approach focuses on building a drowsiness detection mechanism to alert the driver to avoid the catastrophe. In this work, the detection system can identify whether the driver's eyes were closed or open even in low light or dim light and how much time the eyes were in closed state. Based on the time the system will generate an alert.

Keywords: early drowsiness detection, real time monitoring, Reduce accidents, face and land mark detection, eye aspect ratio.

1. INTRODUCTION

In India, traffic accidents caused by fatigued drivers who fall asleep at the wheel account for 40% of all collisions. The research shows how Indian highway drivers disregard the value of getting enough rest and end up putting lives at danger. This number may even be somewhat lower given that, among the many factors that may contribute to an accident, weariness is frequently overlooked. In developing nations like India, a terrible combination of infrastructure and fatigue can lead to calamity. Contrary to alcohol and drugs, which have obvious key signs and tests that are simple to obtain, fatigue is generally quite difficult to measure or detect. The best ways to address this issue are probably to raise awareness of accidents caused by drowsiness and to encourage drivers to report their own. As a job's demand rises, so do its wages, which encourages an increasing number of individuals to take it on. This is true while using a vehicle for transportation at night. Drivers are motivated by money, which also denotes drowsiness, fatigue, or the inability to maintain eye contact. We can offer more sophisticated solutions to improve our standard of living thanks to technological advancement.

According to NHTSA data, nearly 100,000 crashes involving drowsy driving are reported each year. The precise number would be far higher. Deep insights into a variety of physiological aspects of the body can be gained from facial expressions. Face detection is the first step in the process and there are a plethora of algorithms and strategies available for it. Humans who feel drowsy exhibit certain highly unique facial expressions and movements, such as starting to close their eyes. Tracking eyes to identify fatigue and categories a driver as drowsy is one solution to this major issue.

2. OBJECTIVES

Enhance Safety: The primary objective is to improve road safety by preventing accidents and incidents caused by drowsy or fatigued drivers.

Early Drowsiness Detection: Detect the early signs of drowsiness in drivers, allowing for timely intervention and alerting to prevent accidents.

Real-Time Monitoring: Continuously monitor the driver's state of alertness in real time, providing immediate feedback when drowsiness is detected.

Reduce Accidents: Decrease the number of accidents and collisions attributed to driver fatigue by proactively warning drivers and/or triggering safety measures

Cost-Effectiveness: Consider the cost-effectiveness of the solution, taking into account the benefits of reducing accidents and improving safety

3. PROBLEM STATEMENT

Develop an efficient drowsiness detection system using machine learning for long-haul truck drivers. The system must use in-cab cameras and physiological data to detect drowsiness and provide timely alerts, enhancing safety by mitigating the risk of accidents due to driver fatigue.

@International Journal Of Progressive Research In Engineering Management And Science



www.ijprems.com

editor@ijprems.com

e-ISSN: INTERNATIONAL JOURNAL OF PROGRESSIVE 2583-1062 **RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)** Impact

Vol. 04, Issue 05, May 2024, pp: 193-199

Factor: 5.725

Due to the use of EEG (Electroencephalography) and ECG(Electrocardiography) in the existing systems, it makes the driver feel uncomfortable while driving. Some of the existing systems use complex algorithms which may delay the system in detecting the state of the driver if he/she is drowsy.

4. PROPOSED SYSYTEM

Almost everyone has experienced this drowsiness problem while in driving. The people worst affected by drowsy driving are the teenagers, professional truck drivers who have to drive on long routes for a long period of the time without breaks, cab drivers who also drive for a long period of time, sometimes to complete their targets to get bonus profit and shift workers who work at night or for late hours also get affected due to tiredness.

Drowsy driving can be happened due to driving on highways for long time without any breaks particularly at night. To overcome the above problem, the drowsiness detection and prevention system is implemented.

This paper describes the solution to this drowsiness problem. Our system uses the camera to monitors the driver. It is placed in front to the driver to analyze the behavior of the driver i.e. whether the driver's eyes are closed or not. The front camera constantly captures the image of the driver and with the help of this checks whether the eyes of the driver are closed or not.

Advantages: This system provides Drowsiness detection and accident prevention. It detects the Driver Continuously.

5. HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

- Processor - Pentium IV
- RAM 4 GB (min)
- Hard Disk 20 GB
- Key Board Standard Windows Keyboard
- Mouse Two or Three Button Mouse
- Monitor SVGA

SOFTWARE REQUIREMENTS:

- Operating system Windows 7, 8, 10, 11, mac os •
- Coding Language Python
- Back-End Python

6. TECHNOLOGY DESCRIPTION

Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library

7. PACKAGES USED

VLC: VLC (VideoLAN Client) is a free and open-source multimedia player developed by the VideoLAN project. It supports various audio and video codecs, streaming protocols, and formats. VLC is widely used for playing media files on various platforms including Windows, macOS, Linux, iOS, and Android.

Train: The term "train" might refer to various things depending on the context. In programming, there might not be a specific package named "train."

However, it could refer to libraries or modules related to machine learning, such as TensorFlow, Py Torch, or scikitlearn, which are commonly used for training machine learning models.

sys: The sys module in Python provides functions and variables that are used to manipulate different parts of the Python runtime environment. It allows interaction with the interpreter and provides access to system-specific parameters and functions.



www.ijprems.com

editor@ijprems.com

e-ISSN: INTERNATIONAL JOURNAL OF PROGRESSIVE 2583-1062 **RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)** Impact

Vol. 04, Issue 05, May 2024, pp: 193-199

Web Browser: A web browser is a software application used to access information on the World Wide Web. Examples of popular web browsers include Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and Opera. Web browsers allow users to view web pages, navigate between pages, and interact with web-based content.

8. ALGORITHM

Initialization: Initialize necessary parameters and resources like video capture, facial detector, shape predictor, and alert sound.

Loop through Frames: Continuously capture frames from the camera.

Face and Landmark Detection: Detect faces in the frame using a face detector. If a face is detected, extract facial landmarks (eye, mouth).

Calculate Eye Aspect Ratio (EAR): Calculate the EAR for both eyes. Compute the average EAR.

Check for Drowsiness:

If the average EAR falls below a certain threshold (close thresh), increment a drowsiness counter (flag).

Check if a yawn is detected (yawn_countdown).

Check the face direction for signs of drowsiness (getFaceDirection).

If the drowsiness counter exceeds certain thresholds (frame thresh 1, frame thresh 2, frame thresh 3), trigger an alert and update a map counter (map_counter).

Alert and Actions:

If drowsiness is detected, play an alert sound (alert.play()).

If the map counter reaches a threshold (map_counter >= 3), play a "take a break" sound and open a web browser to find nearby hotels or motels.

Resetting Flags:

If the EAR rises above the threshold, reset the drowsiness counter and stop the alert.

Display and Control:

Display the frame with any annotations.

Listen for the 'ESC' key to exit the loop.

Sample Code:

import cv2 import math import numpy as np import dlib import imutils from imutils import face utils from matplotlib import pyplot as plt import vlc import train as train import sys, webbrowser, datetime def yawn(mouth): return((euclideanDist(mouth[2], mouth[10])+euclideanDist(mouth[4], mouth[8]))/(2*euclideanDist(mouth[0], mouth[6]))) def getFaceDirection(shape, size): image_points = np.array([shape[33], shape[8], shape[45], shape[36], shape[54], shape[48]], dtype="double") model_points = np.array([(0.0, 0.0, 0.0),

JPREMS	
~~~	

www.ijprems.com Vol. 04, Issue 05, May 2024, pp: 193-199 5.725 editor@ijprems.com (0.0, -330.0, -65.0),(-225.0, 170.0, -135.0), (225.0, 170.0, -135.0),(-150.0, -150.0, -125.0), (150.0, -150.0, -125.0)]) $focal_length = size[1]$ center = (size[1]/2, size[0]/2)camera_matrix = np.array( [[focal_length, 0, center[0]], [0, focal_length, center[1]], [0, 0, 1]], dtype = "double") $dist_coeffs = np.zeros((4,1))$ (success, rotation_vector, translation_vector) = cv2.solvePnP(model_points, image_points, camera_matrix, dist coeffs, flags=cv2.SOLVEPNP ITERATIVE) return(translation_vector[1][0]) def euclideanDist(a, b): return (math.sqrt(math.pow(a[0]-b[0], 2)+math.pow(a[1]-b[1], 2))) def ear(eye): return ((euclideanDist(eye[1], eye[5])+euclideanDist(eye[2], eye[4]))/(2*euclideanDist(eye[0], eye[3]))) def writeEyes(a, b, img): y1 = max(a[1][1], a[2][1]) $y_2 = min(a[4][1], a[5][1])$ x1 = a[0][0] $x^2 = a[3][0]$ cv2.imwrite('left-eye.jpg', img[y1:y2, x1:x2]) y1 = max(b[1][1], b[2][1]) $y_2 = min(b[4][1], b[5][1])$ x1 = b[0][0] $x^2 = b[3][0]$ cv2.imwrite('right-eye.jpg', img[y1:y2, x1:x2]) # open_avg = train.getAvg() # close_avg = train.getAvg() alert = vlc.MediaPlayer('focus.mp3')  $frame_thresh_1 = 15$  $frame_thresh_2 = 10$  $frame_thresh_3 = 5$  $close_thresh = 0.3 # (close_avg+open_avg)/2.0$ flag = 0 $yawn_countdown = 0$  $map_counter = 0$  $map_flag = 1$ capture = cv2.VideoCapture(0)avgEAR = 0detector = dlib.get_frontal_face_detector() predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat') (leStart, leEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"] (reStart, reEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]



www.ijprems.com Vol. 04, Issue 05, May 2024, pp: 193-199 5.725 editor@ijprems.com (mStart, mEnd) = face utils.FACIAL LANDMARKS IDXS["mouth"] while(True): ret, frame = capture.read() size = frame.shape # gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) gray = framerects = detector(gray, 0)if(len(rects)): shape = face_utils.shape_to_np(predictor(gray, rects[0])) leftEye = shape[leStart:leEnd] rightEye = shape[reStart:reEnd] leftEyeHull = cv2.convexHull(leftEye) rightEyeHull = cv2.convexHull(rightEye) # print("Mouth Open Ratio", yawn(shape[mStart:mEnd])) leftEAR = ear(leftEye) #Get the left eye aspect ratio rightEAR = ear(rightEye) #Get the right eye aspect ratio avgEAR = (leftEAR + rightEAR)/2.0eyeContourColor = (255, 255, 255) if(yawn(shape[mStart:mEnd])>0.6): cv2.putText(gray, "Yawn Detected", (50,50), cv2.FONT_HERSHEY_COMPLEX, 1,(0,255,127),2) yawn_countdown=1 if(avgEAR<close_thresh): flag+=1 eyeContourColor = (0,255,255)print(flag) if(yawn_countdown and flag>=frame_thresh_3): eyeContourColor = (147, 20, 255) cv2.putText(gray, "Drowsy after yawn", (50,50), cv2.FONT_HERSHEY_COMPLEX, 1,(0,255,127),2) alert.play() if(map_flag):  $map_flag = 0$ map_counter+=1 elif(flag>=frame_thresh_2 and getFaceDirection(shape, size)<0): eyeContourColor = (255, 0, 0)cv2.putText(gray, "Drowsy (Body Posture)", (50,50), cv2.FONT_HERSHEY_COMPLEX, 1,(0,255,127),2) alert.play() if(map_flag):  $map_flag = 0$ map_counter+=1 elif(flag>=frame_thresh_1): eyeContourColor = (0, 0, 255)cv2.putText(gray, "Drowsy (Normal)", (50,50), cv2.FONT_HERSHEY_COMPLEX, 1,(0,255,127),2) alert.play() if(map_flag):  $map_flag = 0$ map_counter+=1 elif(avgEAR>close_thresh and flag):



www.ijprems.com editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 193-199

**Factor:** 5.725

print("Flag reseted to 0") alert.stop() yawn_countdown=0 map_flag=1 flag=0 if(map_counter>=3): map_flag=1 map_counter=0 vlc.MediaPlayer('take_a_break.mp3').play() webbrowser.open("https://www.google.com/maps/search/hotels+or+motels+near+me") cv2.drawContours(gray, [leftEyeHull], -1, eyeContourColor, 2) cv2.drawContours(gray, [rightEyeHull], -1, eyeContourColor, 2) writeEyes(leftEye, rightEye, frame) if(avgEAR>close_thresh): alert.stop() cv2.imshow('Driver', gray) if(cv2.waitKey(1)==27): break capture.release() cv2.destroyAllWindows()

# 9. TESTING

The testing of the "DRIVER'S DROWSINESS " system involves various methodologies to ensure its reliability, accuracy, and effectiveness. Here are key testing methodologies employed during different stages of development:

## **TYPES OF TESTS:**

### Unit Testing:

Test individual modules and components in isolation to ensure they function as intended. Verify that each unit of code, such as algorithms, preprocessing functions, and user interface components, produces the expected output.

### **Integration Testing:**

Evaluate the interaction between different modules and components to ensure seamless integration. Verify that the system components work together as a cohesive unit, detecting and resolving any issues related to data flow and communication.

Functional Testing: Validate that the system meets the specified functional requirements. Test core functionalities such as data analysis, predictive modeling, user interface interactions, and cross-disciplinary adaptability.

### **Performance Testing:**

Evaluate the system's responsiveness, scalability, and resource usage under different conditions. Test the application's ability to handle large datasets and complex computations without compromising performance.

### **Regression Testing:**

Verify that new updates or modifications do not adversely impact existing functionalities. Re-run previously conducted tests to ensure that any changes have not introduced new errors or broken existing features.

#### Security Testing:

Assess the system's resistance to unauthorized access, data breaches, and other security vulnerabilities. Implement measures to safeguard sensitive data processed by the application.

### Adaptive Learning and Continuous Improvement Testing:

Assess the system's ability to adapt and improve over time. Validate that the adaptive learning mechanisms are effectively enhancing the system's predictive capabilities based on new data and scenarios.

### User Acceptance Testing (UAT):

Involve end-users, researchers, and scientists in the testing process to gather feedback on the system's usability and effectiveness. Ensure that the system aligns with user expectations and fulfils their requirements.



e-ISSN:

www.ijprems.com editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 193-199

# **10. CONCLUSION**

In this project I was able to classify the images of the driver based on the eye positions as sleepy or not sleepy, I was able to alert the driver and his/her co-passengers when the driver was feeling drowsy. 41% of road accidents in India are caused because the driver feels drowsy when he/she is driving the vehicle, I believe that this model will help in reducing the % of road accidents caused due to drowsiness. For future implementation or work that I would be doing on this project will be the accurate detection of this model in night and low light conditions, so that it will bring about a change in the aspect of night travelling and will make people to feel secured even if they are driving alone.

## **11. REFERENCES**

- [1] Dipesh Patel, Kunal Bendale(2022). Drowsiness Detecting from Face Images (eyes and mouth) Using Tensorflow & Open-cv with Real time Video. 7(4), 190-193.
- [2] K. Sree Rama Murthy, Bhavana Siddineni(2020). An Efficient Drowsiness Detection Scheme using Video Analysis.11(1),574-581.
- [3] S. Kranthi Reddy, Cheruku Sai Ganesh (2020). DRIVER DROWSINESS DETECTION SYSTEM.7(4) 2007-2011.
- [4] Mahek Jain, Bhavya Bhagerathi, Sowmyarani C N (2021). Real-Time Driver Drowsiness Detection using Computer Vision.2(1) 109-113.
- [5] Rohith Chinthalachervu, Immaneni Teja(2022). Driver Drowsiness Detection and Monitoring System using Machine Learning. 1-12.
- [6] Vasundhara Iyer, Atharv Vanjari (2021). Driver Drowsiness Detection System. 10(12) 275-278.
- [7] AYMAN ALTAMEEM, ANKIT KUMAR, RAMESH CHANDRA POONIA (2021). Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning. 162805-162819.
- [8] Md. Tanvir Ahammed Dipu, Syeda Sumbul Hossain (2021). Real-time Driver Drowsiness Detection using Deep Learning. 12(7) 844-850.
- [9] Aishwarya Biju and Anitha Edison(2020). Drowsy Driver Detection Using Two Stage Convolutional Neural Networks. 7-12.