IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

DETECTING DRIVER DROWSINESS

USING MACHINE LEARNING

Suraj.R , Manish.M , T.Bindu Priya , Sathwika.V, Madhuri.P

Student/Scholar, Student/Scholar, Student/Scholar, Student/Scholar, Associate Professor
Computer Science and Engineering
(Cyber Security)
Hyderabad Institute of Technology and Management (HITAM), Gowdavelly(V), Medchal(M), Medchal-Malkajgiri Dist. 501401, Telangana, India

Abstract :

In today's quickly changing world, drowsy driving is a global hazard to road safety. This resulted in several accidents and fatalities. To address this issue, researchers developed a tiredness detection system that uses machine learning to evaluate driver eye behavior. This unique device uses computer vision techniques to continuously monitor the driver's eyes for indicators of tiredness, such as prolonged eye closure and abnormal eye movements. The technique, which consists of three major components—eye recognition, feature extraction, and sleepiness classification—uses advanced computer vision approaches such as Haar cascades and convolutional neural networks for exact eye localization. Rigorous real-world tests and a broad dataset were used to assess the system's performance, taking into account measures like as accuracy, precision, recall, and score. To summarize, this system presents a promising solution.

To evaluate the performance of the proposed system, extensive experiments were conducted with real-world driving scenarios and diverse driver demographics. To train and evaluate the sleepiness detection model, a huge dataset of driving conditions and driver demographics was compiled. The assessment criteria used to determine the system's overall efficacy in correctly and quickly detecting drowsiness are accuracy, precision, recall, and score.

Finally, this abstract offers a machine learning-based sleepiness detection system that uses eye behavior as a valid indicator of driver drowsiness. The system can successfully contribute to road safety by offering early warnings and reducing the occurrence of drowsy driving events by correctly identifying and classifying drowsy states.

1.INTRODUCTION

In today's fast-paced world, road safety is one of the most pressing concerns, and driver weariness remains a significant threat. Drunk driving has serious consequences, including the likelihood of accidents, injuries, and even death. The use of machine learning technologies to identify and decrease the risks associated with driver fatigue is becoming increasingly advantageous as technology advances. The difficulty of recognizing drowsy driving behavior is the primary emphasis of this study, which employs machine learning approaches. The goal is to develop an intelligent system that can monitor driver conduct and issue timely alerts when tiredness is identified. Fatigue-related accidents are a big issue all over the world, accounting for a sizable proportion of traffic events.

Driving is a common and vital action that significantly improves our freedom and mobility on a daily basis. However, there is a major risk 2 to road safety as the incidence of crashes increases, many of which are caused by fatigued drivers. Drowsy driving increases the likelihood of accidents because it impairs a driver's concentration, reaction time, and decision-making abilities. The Driver Drowsiness Detection project attempts to create a system that can detect drowsiness in drivers using machine learning methods. Drowsiness detection is critical for guaranteeing road safety because drowsy driving can result in accidents and fatalities. The technology uses facial landmarks and an eye aspect ratio (EAR) computation to assess whether the driver is drowsy based on their eye movements.

The Driver Drowsiness Detection project attempts to create a system that can detect drowsiness in drivers using machine learning methods. Drowsiness detection is critical for guaranteeing road safety because drowsy driving can result in accidents and fatalities. The technology uses facial landmarks and an eye aspect ratio (EAR) computation to assess whether the driver is drowsy based on their eye movements. The threat of sleepy driving presents a significant problem,

endangering the lives of drivers, passengers, and pedestrians. Every year, many accidents occur around the world as a result of drivers succumbing to weariness, resulting in sad outcomes and irreversible losses. Recognizing the crucial need to address this widespread issue, the "Driver Drowsiness Detection using Machine Learning Algorithms" project emerges as a light of hope, hoping to harness the power of technology to mitigate the risks associated with drowsy driving.

At its core, this project aims to use developments in machine learning, computer vision, and sensor technologies to create an intelligent system capable of detecting tiredness in drivers in real time. By evaluating facial landmarks and monitoring eye movements, the system can detect mild signs of weariness, alerting drivers to take appropriate measures or interventions to avoid accidents. 3 In response to this important issue, technological advancements, particularly in the field of machine learning, have paved the way for novel approaches to identifying and mitigating the hazards of driver fatigue.

2.LITERATURE SURVEY

Detecting driver drowsiness requires recognizing significant cues that indicate a decrease in alertness, which is critical for avoiding crashes. This section provides an in-depth review of the behavioral and physiological signs associated with tiredness. Physiological signs in eye movements include changes in head and body position, extended eyelid closure, and slow blink rates. Behavioral markers include variations in driving habits such as jerky steering and uneven speed control. Understanding these indicators is necessary for designing a comprehensive and efficient sleepiness detection system. The inclusion of machine learning, which allows for more advanced analysis and interpretation of complex data patterns, represents a significant advancement in drowsiness detection. The application of machine learning algorithms, including ensemble techniques and deep learning,in sleepiness detection is examined in this subsection.These methods improve classification and prediction accuracy by allowing significant traits to be extracted from a wide range of datasets. Machine learning research provides innovative ways for training models using tagged 10-datasets of alert and sleepy phases. A study of the literature demonstrates how machine learning algorithms have evolved to tackle the complexities of driver sleepiness detection, setting the framework for the development of an intelligent and adaptive system.

3.Future Scope

The future development of the website for the SHASTR project will involve several main areas of enhancement and extension. Advanced security features, the integration of additional tools, user training and awareness, code optimization, scalability, and performance optimization. The driver drowsiness detection project's future scope is large and promising, with opportunities for enhancements and development across multiple dimensions. As technology advances, sleepiness detection systems' effectiveness, efficiency, and integration can be greatly improved. Future versions of sleepiness detection systems can use more advanced machine learning methods, such as deep learning and neural networks, to improve accuracy and responsiveness.

These algorithms can scan larger datasets to better comprehend patterns of drowsiness, resulting in more precise and early identification of fatigue 54. Integration with other in-vehicle safety systems is another promising research area. Drowsiness detection systems can be further combined with adaptive cruise control, lane-keeping assistance, and automatic emergency braking systems. This integration has the potential to produce a comprehensive safety net that not only notifies the driver but also takes corrective action to prevent accidents, ultimately improving overall vehicle safety.

4. METHODOLOGY

STEP 1: Frames are continuously taken via a camera.

Step 2: The face is recognized and cropped from the frame.

STEP 3: Facial markers, such as the corners of the eyes, are identified inside the facial region. These landmarks can be identified using algorithms such as dlib or deep learning-based approaches.

STEP-4: The Eye Aspect Ratio (EAR) is calculated using the observed face landmarks. EAR is a metric that determines if the eyes are open or closed.Machine learning methods compare the calculated EAR values over time to detect whether the driver is drowsy.

STEP 5: If drowsiness is detected, an alarm will sound to inform the driver.To further notify the driver, an alarm message may appear on the screen or dashboard.

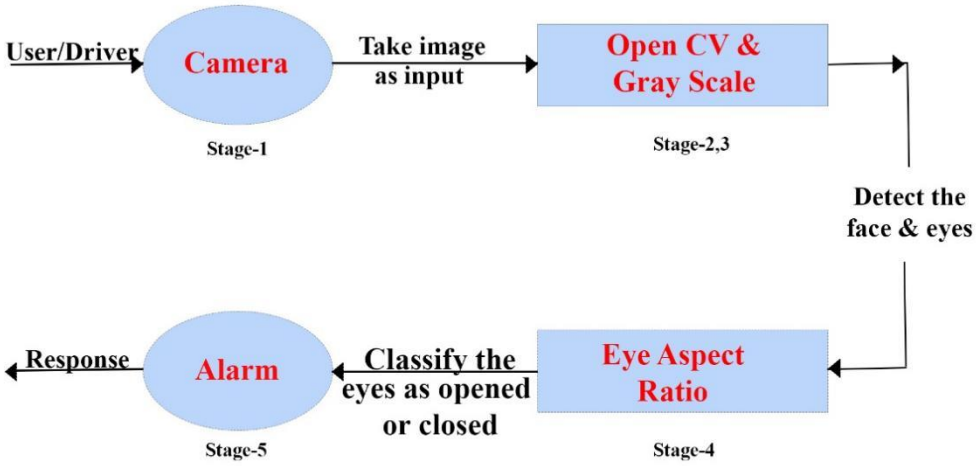

FIG:- Block Diagram

5.REQUIREMENTS AND INSTALLATION:

Hardware Requirements

Web Server

- Processor : Intel Core i5 or equivalent, multicore
- Hard Disk : SSD with at least 100 GB storage capacity
- RAM : 8 GB (16 GB recommended)

Database Server

- Processor : Intel Core i5 or equivalent, multicore
- Hard Disk : SSD with at least 100 GB storage capacity
- RAM : 8 GB (16 GB recommended)

Networking

- Network Bandwidth : Gigabit Ethernet interface

.Software Requirements

- Operating System : Windows 10,Linux,Mac OS
- Programming Languages PYTHON

Technologies used

- OpenCV
- Scipy
- Playsound

6.IMPLEMENTATION

import cv2
import dlib

These lines import the OpenCV (cv2) library, a prominent library for computer vision applications, as well as the dlib library, a machine learning and computer vision toolkit.

predictor=dlib.shape_predictor("/Users/sathwika/Downloads/shape_ predictor_68_face_landmarks.dat")

Here, we load a pre-trained face landmark prediction model. This model was trained with a gradient boosting-based form prediction technique. It identifies and forecasts the coordinates of 68 facial landmarks on identified faces, including the corners of the eyes, nose, and mouth.

from scipy.spatial import distance as dist
from playsound import playsound

These lines import particular functions from the scipy.spatial (distance) and playsound modules. The distance function calculates the Euclidean distance between two places, while playsound plays alarm sounds when drowsiness is detected.

def calculate_eye_aspect_ratio(eye):
A = dist.euclidean(eye[1], eye[5])
B = dist.euclidean(eye[2], eye[4])
C = dist.euclidean(eye[0], eye[3])
ear = (A + B) / (2.0 * C) return ear

This function, calculate_eye_aspect_ratio, computes the Eye Aspect Ratio (EAR). EAR is a metric that determines if a person's eyes are open or closed. It is calculated using the Euclidean distance between particular spots around the eyes.

def sound_alarm(path):
playsound(path)

When this function, sound_alarm, is called, it plays the alarm sound supplied in the route parameter. It employs the playsound library to play the sound.

detector = dlib.get_frontal_face_detector()

This line initializes the face detector with the dlib library's get_frontal_face_detector function. The detector detects faces in pictures or video frames.

predictor=dlib.shape_predictor("/Users/sathwika/Downloads/shape_* predictor_68_face_landmarks.dat")

We just discussed this line above. It initializes the facial landmark predictor with the pre-trained model file.

EYE_AR_THRESH = 0.25
EYE_AR_CONSEC_FRAMES = 48

These lines describe the constants used to determine whether a person's eyes are closed. EYE_AR_THRESH provides the threshold value for the eye aspect ratio below which the eyes are regarded closed, and EYE_AR_CONSEC_FRAMES is the number of consecutive frames the eyes have to be closed before activating the alert.

The remainder of the code contains the main loop, which gathers video frames from the webcam, detects faces, calculates eye aspect ratios, and sounds an alarm when drowsiness is detected. This code detects drowsiness by calculating the eye aspect ratio, which is the ratio of eye width to eye height. When a person's eyes are closed for an extended period of time, indicating drowsiness, an alert is triggered. The dlib package is utilized for facial landmark detection and face recognition, while OpenCV is used for video capture and processing.

7.CONCLUSION:

In this early stage of SHASTR's conceptualization, we investigated the complexities of existing penetration testing systems, such as operating systems designed specifically for penetration testing and independent online tools for vulnerability analysis, and identified inherent challenges such as ease of access, reliance on high-end computer systems, and limited availability for a single task. Our proposed solution, SHASTR, emerges not just as a response to these difficulties, but also as an informing platform with the potential to revolutionize the cybersecurity landscape.

The driver drowsiness detection project is an important application of machine learning and computer vision technology for improving road safety and preventing accidents caused by driver weariness. Using facial landmark identification, eye tracking, and machine learning algorithms, the system can reliably measure the driver's state of attentiveness in real time and deliver early warnings to avoid future accidents. Several major insights and outcomes were gained during the project's development and implementation.

To begin, the system illustrates the efficacy of using face landmarks and eye aspect ratios as markers of drowsiness, giving a dependable approach for identifying driver fatigue. The use of modern technologies such as OpenCV, dlib, and machine learning frameworks has resulted in the creation of a reliable and effective drowsiness detection system. 53 The driver sleepiness detection initiative offers a big step forward in using technology to address road safety issues. By recognizing and minimizing the dangers associated with driver weariness, the system helps to reduce accidents, save lives, and promote safe driving practices. Moving forward, ongoing research, development, and integration of new technologies will improve the effectiveness and usability of sleepiness detection systems, ultimately making roadways safer for all.

8.References

[1] g0tmi1k, kali linux documentation, kali.org .

[2] Parrot OS, parrot os documentation, parrotsec.org, .

[3] N Hamzah, A Ariffin, H Hamid, Web-Based Learning Environment Based on Students' Needs,

[4] Basu, S. (2022, June 29). 7 penetration testing phases for web applications: A detailed account. Astra.

[5] Agio. (2022, June 8). Vulnerability scanning vs. penetration testing.

[6] UML Diagrams, geeksforgeeks.org,

[7] Nmap Reference Guide, nmap.org, 60

[8] Nikto Documentation, cirt.net/Nikto2,

[9] SQLmap, sqlmap.org,

[10] Searchsploit Manual, exploit-db.com,

[11] 7 Pentesting Tools you must know about, hackerone.com.