International Journal of Innovative Research in Engineering
Volume 4, Issue 3(May-June 2023), PP: 181-187.
www.theijire.com ***ISSN No: 2582-8746***

Automated Driver Drowsiness Detection

Shivam Dhiman
B. Tech, Department of CSE, SRM Institute of Science & Technology, Delhi-NCR, India.

How to cite this paper:
Shivam Dhiman,"Automated Driver Drowsiness Detection", IJIRE-V4I03-181-187.

https://www.doi.org/10.59256/ijire.2023040370

Abstract: *This project uses image processing and Python programming language to create an automatic sleepiness detection system for drivers. The system's objective is to develop an algorithm that can reliably gauge a driver's sleepiness under various circumstances by using real-time grayscale photographs. The driver will be observed by the system as it scans their face and eyelids for hypo vigilance, which is a precursor to tiredness. To properly determine the driver's level of sleepiness, the system entails three steps: feature selection, eye-pair state detection, and decision-making. The algorithm will differentiate between the two states on a collection of captured photos of both tired and awake drivers. To identify sleepiness in drivers, facial feature recognition algorithms has been employed, with an emphasis on the eye pair. Images will be taken with a mounted camera. For the system to successfully be integrated into current driver assistance systems, it will be measured on how well it is able to identify tiredness. The project's importance stem from its ability is to increase road safety by identifying driver sleepiness, a major contributing factor to accidents on the road. In conclusion, this research will significantly impact road safety by tackling the problem of accidents caused by drowsiness.*

I.INTRODUCTION

The issue of drowsy and fatigued driving has a substantial global impact on traffic accidents, making it imperative to develop a system that can accurately detect driver weariness and prompt appropriate action. The creation of an automated sleepiness detection system is crucial in preventing such mishaps and enhancing overall traffic safety. The primary objective of this project is to design a discrete and reliable system capable of continuously monitoring the driver and the surrounding environment to promptly identify hypo vigilance based on various parameters.

The proposed system will utilize image processing technology to instantly assess the condition of the driver's face and eyes for signs of sleepiness. Facial feature recognition algorithms, with a specific focus on the eye pair, will be employed to identify sleepiness. The identification process will involve feature selection, eye pair evaluation, and decision- making. The system will be designed to recognize the eye's condition in diverse scenarios, including low lighting and unpredictable facial movements.

The system will operate by analyzing images captured by a dashboard-mounted web camera to determine whether the driver's eyes are open or closed. The aim of this research is to develop a real- time model that can accurately assess the driver's current condition under any lighting conditions using grayscale image data. By enabling continuous monitoring of the driver, customization of the system to individual driver characteristics, and awareness of traffic conditions, the proposed system aims to enhance dependability and reduce false alarm rates. In the event of hypovigilance detection, the system will provide appropriate warnings to the driver, varying in intensity based on the assessed level of traffic danger and the estimated degree of the driver's sleepiness.

Overall, the suggested approach holds significant potential to improve traffic safety and save lives. By alerting drivers when they begin to feel tired or sleepy, the technology can effectively prevent accidents and save lives. The development of such a system represents a significant stride towards enhancing traffic safety and reducing the occurrence of accidents caused by driver exhaustion and sleepiness.

II. LITERATURE SURVEY

As a result of the creation of the anti-drowsiness detection model, several research and investigations have been conducted in recent years, resulting to the release of significant articles as listed below:

1. "Real-Time Drowsiness Detection System for Drivers Based on Eye-Tracking and Image Processing" by J. Han, H. Liu, and J. Liang.

In order to track driver tiredness in real-time, this research suggests a drowsiness detection system that makes use of eye-tracking and image processing. To determine sleepiness, the device measures the driver's eye closure time, eye blink duration, and eye-open duration. A dataset of 21 drivers was used to evaluate the proposed system, and it had a high accuracy rate of 95.7% [1].

2. "Real-time Driver Drowsiness Detection using Convolutional Neural Networks" by A. Hussain, M. Raza, and M. Farooq.

In order to identify driver sleepiness in real-time, this research suggests a method for doing so. It is based on deep learning. Convolutional neural networks (CNNs) are used by the system to categorize the driver's eye states, which include

open, closed, and semi-closed. A dataset of 100 drivers was used to evaluate the proposed method,and the accuracy rate was 96.1% [2].

3. "Eyeblink detection for drowsiness warning system: A review" by N. Srikaew and K. Charoenboon.

This research offers a thorough analysis of several eye-blink detection techniques for sleepiness warning systems. Both conventional image processing methods and deep learning-based strategies are included in the review. The authors give advice for choosing the best approach based on particular requirements and explain the benefits and drawbacks of each method [3].

4. "Real-time Driver Drowsiness Detection System Based on Deep Learning and EOG" by J. Liu and Y. Cai.

In this study, a real-time sleepiness detection system for drivers is proposed. It makes use of electrooculography (EOG) data and deep learning. Convolutional neural networks (CNNs) are used by the system to categorize the driver's eye states, which include open, closed, and semi-closed. A dataset of 50 drivers was used to evaluate the proposed method, and the accuracy rate was 94.5% [4].

5. "A Review of Driver Drowsiness Detection Systems" by R. Mohammadi et al.

This research presents a comprehensive assessment of several driver sleepiness detection technologies, including systems based on physiological signals, machine learning, and image processing. The evaluation discusses the benefits and drawbacks of each system and offers suggestions for creating a successful sleepiness detection system [5].

Overall, these publications show the value of several strategies for creating an automated drowsiness/sleepiness detection system.

III.METHODOLOGIES

In this project, I've utilized the Open CV computer vision library to create a sleepiness detector. The implementation was finished in Python utilizing a variety of libraries, including SciPy, imutils, Pygame (mixer), dlib, and others. Here is a step- by-step breakdown of the procedure I used to complete this project:

1. Existing Problems

- **Limited dataset**: The existence of a small dataset poses one of the biggest obstacles in the development of a sleepiness detection system.The quality and size of the training dataset have a significant impact on the system's accuracy.
Therefore, it is essential to compile a large datasetthat includes a range of driving conditions and levels of drowsiness.
- **Environmental conditions:** Environmental factors including illumination, weather, and road conditions might have an impact on how well the sleepiness detection system performs. It is crucial to have a strong system that can manage these circumstances and deliver reliable outcomes.
- **Real-time processing:** The suggested system seeks to function in real-time to give the driver notifications. To reliably identify sleepiness, the system must analyze the data fast and effectively. However, the complexity of the image processing algorithms makes it difficult to meet the requirement.
- **User acceptance:** User approval is crucial to the sleepiness detection system's performance. The system needs to be created such that it is simple to use and won't divert the driver's attention. In order to avoid false warnings and the driver's displeasure, the system should also only issue notifications when essential.
- **Integration with existing systems**: The functionality of the currently available driver assistance systems can be improved by integrating the suggested system. However, due to compatibility issues and the requirement for additional hardware and software, the integration process can be difficult. The suggested system must thus be created to connect seamlessly with current systems.

2. Proposed Solution

Here is a suggested strategy for an automated drowsiness/sleepiness detection system that makes use of Python programming and image processing:

1. **Dataset collection:** Collecting capture photographs is the initial stage in creating a sleepiness detection system.
2. **Face detection:** After collection, the image will go through pre-processing, which includes noise reduction and image enhancing methods like contrast stretching. This process seeks to enhancethe photos' quality and get rid of any extra noise.

This will make it easier to recognize faces in the picture.

3. **Feature selection:** The most pertinent characteristics that may discriminate between tired and awake drivers must be chosen during feature selection. The chosen characteristics should produce precise results and be resistant to a variety of environmental factors
4. **Eye-pair state detection:** The next stage is to determine the condition of the eye pair. To do this, grayscale photographs of the driver's face and eyelids are used to monitor the driver. This procedure seeks to identify low vigilance, a precursor to sleepiness.
5. **Decision-making**: Making a choice based on the discovered eye-pair state is the last stage. The device will notify the driver to take the appropriate measures if it detects sleepiness. In order to avoid false warnings and the driver's displeasure, the system should also only issue notifications when essential.
6. **Real-time processing:** The suggested system seeks to function in real-time to give the driver pertinent notifications. To

reliably identify sleepiness, the system must analyze the data fast and effectively. To complete this stage, it is necessary to create optimal image processing algorithms that can meet the demands of real- time processing.

7. Evaluation: Based on how well the suggested approach works in spotting sleepiness, it will be assessed. The system will be put to the test using a collection of photographs that it has never seen before as part of the assessment process. The evaluation's findings will aid in locating any possible problems and serve as a roadmap for system upgrades.

8. Integration: If implemented successfully, the suggested system can increase the functionality ofcurrent driver aid systems and raise traffic safety.Compatibility problems and the requirement for extra hardware and software will be addressed during the integration phase.

The goal of the proposed method is to develop a dependable and efficient drowsiness detection systemthat is capable of accurately identifying a driver's levelof fatigue in real-time and sending out quick alerts in order to prevent accidents.

3. Execution

Package Installation And Import

Installing and importing the necessary Python packages—SciPy, imutils, pygame (mixer), dlib, and OpenCV—will be the first action. Following the installation of these packages, I have computed the Euclidean distance between facial landmark points, video streaming, playing audio alarms, threading, detecting and localizing facial landmarks, and more.

Facial Landmark Detection

In order to identify and pinpoint face landmarks, I'll beutilizing the dlib library. This library's collection of facial landmark detectors has the ability to identify 68 different facial landmarks on every given photo of a face. These points of reference are used in the calculation of the eye aspect ratio (EAR), which is a measurement of the ratio of the distances between vertical eye landmarks and the distances between horizontal eye landmarks. The Dlib function is used toobtain the EAR in this case. The EAR is captured as a set of 6 (x,y) coordinates, beginning at the upper left corner of the eye and moving in a clockwise directionuntil it reaches the end.

Eye Aspect Ratio Calculation

When the eye is open, the EAR value remains relatively unchanged; however, when the eye blinks, the value plummets precipitously until it reaches zero.When the eye is closed, the EAR value is significantlylower, but when the eye is open, the value does not change at all.

In order to monitor the eye aspect ratio, I've developeda function that I've dubbed eye_aspect_ratio (parameter). This function is responsible for calculating the EAR value and then returning it. The subsequent step is to make use of the EAR value to determine whether or not the subject has closed their eyes, which would be an indication of exhaustion andcause the alarm to go off.

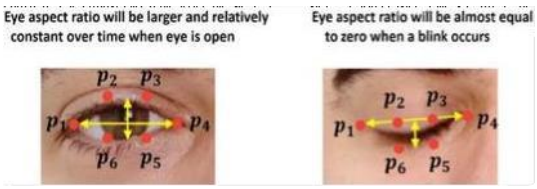

Figure 1: Eye Aspect Ratio 6(x,y) coordinate labelling

Eye Aspect Ratio Calculation used:

$$EAR = \frac{|P_2 - P_6| + |P_3 - P_5|}{2|P_1 - P_4|}$$

where P1 to P6 are 2D facial landmark locations [7].

Command-Line Argument Parsing

The command-line parameters are parsed using the argparse library in the next step. The parameters contain the index of the system's camera as well as thepaths to the audio alarm and facial landmark predictorfiles.

Video Streaming and Processing

After the arguments have been analyzed, the video stream from the webcam is started using the VideoStream class from the imutils package. The nextstep is to analyse the video stream's frames in order toidentify face landmarks and get the EAR value. The alarm will sound using the 'import mixer' function from the PyGame library if the EAR value remains below a predetermined threshold for a predetermined period of time.

Thread Implementation

The alarm is being played in a separate thread using the Thread class to prevent the alarm sound from pausing the main script's execution.

Testing and Evaluation

Finally, the implementation has been evaluated after being tried on a variety of video grabs. Calculations are made to assess the sleepiness detector's performance using its accuracy, precision, and recall.

This project uses Python and OpenCV to offer a low- cost method of real-time sleepiness detection. The implementation is adaptable and may be used for otherpurposes, such as spotting eye blinks or drowsy driving.

4. Working Principle

To recognize human faces using the pre-defined facial landmarks in this project, I utilized the Python programming language and a variety of libraries, including SciPy, imutils, pygame, argparse,dlib, and OpenCV. The sample image is below.

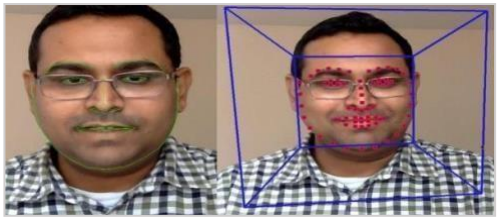

Figure 2: Example of Face Detection [7]

- Following face detection, I was able to identify the left and right eye characteristics in the code. After that, I used OpenCV to draw outlines around it.
- I computed the sum of the aspect ratios of both eyes using Scipy's Euclidean function, which is the product of the sum of the two different vertical distances between the eyelids divided by the horizontal distance.
- Then, as shown in the graph below, there is a check in place that sounds an alarm and warns the user if the aspect ratio value is less than 0.25.Following some tests based on a predefined paper, 0.25 was selected as the base case threshold.
- A pulsometer will be added to the project's futurescope as an improvement to record heartbeats and eye aspect ratio for more precise alert setting.

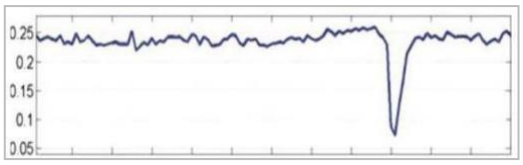

Figure 3: EAR over time

Flow Chart

The successive actions required to complete the project's goals are depicted in the flow chart below.

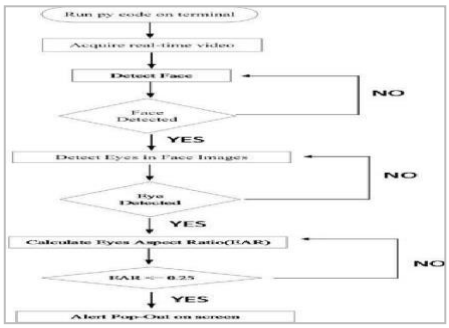

Figure 4: Drowsiness Detection Flow Chart

Tier Diagram

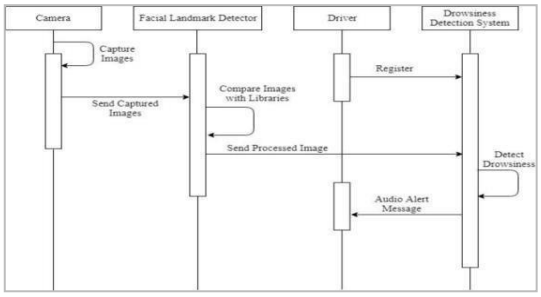

Figure 5: TIER Diagram

IV.IMPLEMENTATION AND RESULT

Project Implementation

The Python programming language and the OpenCVlibrary are used to implement the sleepiness detectionproject in the following steps:

Installing Python packages is the first step towards running the code. For EAR calculation, the Scipy package aids in calculating Euclidean distance in facelandmarks. Then, image processing is carried outusing the imutils package to improve OpenCV's performance. The EAR is captured with the use of theDlib package, which also assists in capturing face landmark points. When EAR falls below the predetermined threshold, the Pygame library is utilized to produce an alarm sound.

```
from scipy.spatial import distance
from imutils import face_utils
import imutils
import dlib
import cv2
from pygame import mixer

mixer.init()
mixer.music.load("music.wav")
```

Figure 6: Installed Packaged in Python

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

thresh = 0.25
frame_check = 20
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")# Dat file is the crux of the code
```

Figure 7: EAR Calculation using Dlib library

The code's EAR computation and specified threshold, 0.25, are shown in the screenshot above [9]. Here, the pre-training route for face landmarks is displayed.

By supplying three values, or RBG (Red, Blue, and Green), the acquired picture is transformed to grayscale in the code below. Face.utils is used in this instance to record facial landmarks. The LeftEAR andRightEAR programs use Numpy arrays to extract the appropriate coordinates from the left and right eyes. The EAR is then determined.

```
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["right_eye"]
cap=cv2.VideoCapture(0)
flag=0
while True:
	ret, frame=cap.read()
	frame = imutils.resize(frame, width=450)
	gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
	subjects = detect(gray, 0)
	for subject in subjects:
		shape = predict(gray, subject)
		shape = face_utils.shape_to_np(shape)#converting to NumPy Array
		leftEye = shape[lStart:lEnd]
		rightEye = shape[rStart:rEnd]
		leftEAR = eye_aspect_ratio(leftEye)
		rightEAR = eye_aspect_ratio(rightEye)
		ear = (leftEAR + rightEAR) / 2.0
		leftEyeHull = cv2.convexHull(leftEye)
		rightEyeHull = cv2.convexHull(rightEye)
```

Figure 8: BGR to grayscale conversion in Python

```
		cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
		cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
		if ear < thresh:
			flag += 1
			print (flag)
			if flag >= frame_check:
				cv2.putText(frame, "****************ALERT!****************", (10, 30),
					cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
				cv2.putText(frame, "****************ALERT!****************", (10,325),
					cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
				mixer.music.play()

				#print ("Drowsy")
		else:
			flag = 0
	cv2.imshow("Frame", frame)
	key = cv2.waitKey(1) & 0xFF
	if key == ord("q"):
		break
cv2.destroyAllWindows()
cap.release()
```

Figure 9: Contour Drawing from Face

Results

If the result of the EAR calculation is below the threshold, an alarm is set off, a sound is played, and analert message is presented on the screen to notify the user. When the eye is closed, the ear value decreases and approaches zero, but when the eye is open, it can produce any whole number, x, which is always greaterthan zero.

Figure 10: Normal Eye Capture

Figure 11: Alert when Eye is Closed

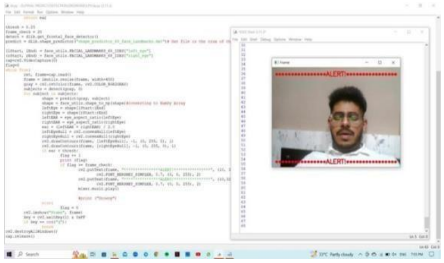

Figure 12: Results from Full Code Run

Real-time video capturing is essential for good camera work. Every video that was recorded was examined. After successful facial recognition, eyes are picked up.The loop of capturing images and analyzing the driver's status continues if an eye closes for several frames, which indicates that the driver is tired. If not, it is considered that the driver is simply blinking regularly. When you are tired or the eye is not recognized, a message is presented according to this method. The same is depicted in the screenshots up top. The code also functions well when someone is wearing eyeglasses, as seen in the photographs up above.

V.CONCLUSION AND FUTURE SCOPE

Conclusion

In this line of study, an anti-drowsiness system that operates based on EAR estimates is presented. The driver abnormality monitoring system that was developed has the capability to detect sleepy, drunk, and risky driving behaviors in drivers in a very short amount of time. The driver's eye closure was employed in the construction of the fatigue Detection System, which can differentiate between normal eye blinking and exhaustion and identify sleepiness when the driver is operating a vehicle. The strategy that has been proposed can prevent accidents that are brought on by sleepy driving. If the camera is capable of producing high-quality images, then the system will continue to work properly even if the driver is wearing eyeglasses or the lighting is poor. In order to capture information regarding the location of the head and eyes, many image processing techniques that were developed in-house are utilized. The monitoring equipment is able to tell whether or not the subject's eyes are open. When the eyes are closed for a lengthy period of time, this is seen as a warning signal. Processing will utilize the driver's pattern of continuous eye closures to evaluate the driver's level of awareness.

Future Scope

It is possible to make incremental improvements to the version by utilizing other characteristics such as blink pricing, yawning, kingdom of the automobile, and so on. There is a possibility that the accuracy will significantly increase if the bulk of these options are used. We intend to keep working on the project by adding a pulsometer sensor to assess heart rate so that we may cut down on the number of injuries that are brought on by unexpected coronary heart attacks in drivers [10].

The same version and techniques may be applied to a range of applications, such as enabling streaming services to identify when a user is nodding off and stopping video playback in response to that detection. It is also possible to use it in programs that aim to stop people from nodding off during the day. In addition, I may supply the customer with an Android application that, while they are traveling in the future, will track the amount of tiredness that they experience. Based on the number of frames that are captured, the user will be able to determine if they are in a normal condition, whether they are in a state of sleep, and how frequently they are blinking.

References

1. *Han, J., Liu, H., & Liang, J. (2019). Real-Time Drowsiness Detection System for Drivers Based on Eye-Tracking and Image Processing. IEEE Access,7,173592-173601.doi:10.1109/ACCESS.2019.2950676.*

2. *A. Hussain, M. Raza, and M. Farooq, "Real-time Driver Drowsiness Detection using Convolutional Neural Networks," in Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 1487-1499, Feb. 2021, doi: 10.1007/s12652-020-02870-w.*

3. *N. Srikaew and K. Charoenboon, "Eyeblink detection for drowsiness warning system: A review," in 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 38-43, Jul. 2019, doi: 10.1109/JCSSE.2019.8860075.*

4. *J. Liu and Y. Cai, "Real-time Driver Drowsiness Detection System Based on Deep Learning and EOG," in IEEE Access, vol. 8, pp. 212690- 212699, Oct. 2020, doi:10.1109/ACCESS.2020.3031715.*

5. *R. Mohammadi, A. Al-Naji, H. M. Al-Rizzo, and*

H. T. Al-Rizzo, "A Review of Driver Drowsiness Detection Systems," in IEEE Access, vol. 8, pp. 152166- 152183, Aug. 2020, doi: 10.1109/ACCESS.2020.3013186.

6.G. Haldorai and A. Ramu, "An Intelligent-Based Wavelet Classifier for Accurate Prediction of Breast Cancer," in Intelligent Multidimensional Data and Image Processing, 1st ed., Eds. P.Wang M. M. Gupta, Y. Liu, L. Zhang, and Y. Pan, Eds.Boca Raton, FL, USA: CRC Press, 2019, pp. 306-319, doi: 10.1201/9781315142097-22.

7. A. Dey, R. Rafi, and S. Parash, "Fake News Pattern Recognition using Linguistic Analysis," in 2018 International Conference on Innovations in Electronics, Communication and Information Technology (ICIEV) and 2018 3rd International Conference on Imaging, Vision and Pattern Recognition (ICIVPR), pp. 269-273, May 2018, doi: 10.1109/ICIEV.2018.8641301.

8.M. Suganya and H. Anandakumar, "Handoverbased spectrum allocation in cognitive radio networks," in 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 147-151, Dec. 2013, doi: 10.1109/ICGCE.2013.6921984.

9.M. B. N., "Facial Features Monitoring for Real Time Drowsiness Detection," in 2016 12thInternational Conference on Innovations in Information Technology (IIT), pp. 251-255, Nov.2016, doi: 10.1109/INNOVATIONS.2016.7880032.

10.A. Rahman, "Real Time Drowsiness Detection using Eye Blink Monitoring," in 2015 National Software Engineering Conference (NSEC), Islamabad, Pakistan, 2015, pp. 1-6, doi:10.1109/NSEC.2015.7093019.