Driver Drowsiness Detection System Using Python In OpenCv

Mr.R.Ganesh[1], Dineshwaran.P[2], Logavarthan.R[3], Suryaram.S[4]

[1]Assistant Professor, Dept of Meachtronics

[2, 3, 4]Dept of Meachtronics

[1, 2, 3, 4] Dr. Mahalingam college of engineering and technology

Abstract- *One of the contributing factors to the rise in accident rates is driver fatigue. The detection and alerting of the driver using a variety of facial recognition techniques has been suggested as a way to prevent accidents. Because of this, it is suggested that this method be used to lessen crashes caused by tired drivers and so improve traffic safety. Based on the system's visual data collection, this system deals with automatically detecting driver drowsiness. After capturing the driver in motion, the photos are further processed, and the presence of weariness is determined. In the event that driver weariness is detected, it quickly raises an alarm for the driver, notifies the owner of the car, and alerts other parties who are concerned with the safety of the vehicle. The method improves the safety procedures used to reduce accidents caused by drowsy drivers. Currently, every member of society wants a safe and secure system. Closed Circuit Television (CCTV), the most widely used technology, can be installed in a car.*

Keywords- OpenCV; Driver Drowsiness; Facial recognition; Safety system.

I. INTRODUCTION

Accidents caused by drowsy drivers, regardless of the time of day or night, are the most frequent sort of accident in today's society. The global accident death rate has increased to 21% as a result. This demonstrates the severity of the issue. The Drowsiness Detection is a safe device that can stop mishaps brought on by drivers who nod off behind the wheel. The goal of this Python project is to create a drowsiness detection model that can identify brief periods of eye closure in drivers. This project's implementation makes use of a pre-built model of a facial landmark for quick deployment on the edge of devices with lower computing efficiency. The project directly impacts the automotive industry. In order to determine if a driver is feeling sleepy or not, this work aims to construct a drowsiness detection model that uses the driver's eyes as the ROI (Region of Interest) and continually feels the eye lid in real-time.

The model will create a sound alarm if the driver starts to nod off to help the person awaken. Because of how serious this issue is demonstrated by the amount of accidents, we decided to create this project that aims to decrease accidents.

II. LITERATURE SURVEY

1)S. Gupta Real-time and static image facial expression recognition. The second international conference on inventive systems and control, 2018.

A technique for identifying emotions in both moving and still images is put forth by Gupta . As a result, they must first use OpenCV's Haar classifier to recognize faces in still images or moving movies before they can try emotion detection. Once the face has been located, the image a clipped image and examined more face landmarks to be discovered. The datasets are next classified utilising the SVM, a machine learning technique, and the eight emotions, and conditioned using facial landmarks. They used SVM to achieve an accuracy of roughly 93,7%. To increase accuracy, these face landmarks may be adjusted.[1]

2) V.K. Sharma (May 2019). construction of a face recognition system. A global intelligent computing and control systems conference was held in 2019.

For specialized applications like access as well as purchasing, criminal, and protection identities, Sharma proposed a facial recognition system. Real-time face recognition, face detection, feature extraction, and categorization will all be part of the identification process, which will be primarily concerned with facial recognition. They employed LBPH for face recognition and Haar-like for face detection, all within the Python environment, using OpenCV. Kivy used to create user interfaces that make the suggested executable software across a variety of platforms.[2]

3) The International Journal of Computer Applications Foundation of Computer Science (FCS) published a work titled work on Object Detection Using Open CV Python by

Bhumika Gupta, Ashish Chaube, Ashish Negi, and UmangGoel. 2017.

According to Bhumika Gupta et al., object detection focuses on finding objects or instances of a particular class (such humans, flowers, or animals) in digital photos and videos. It is a well-known computer technology related to image processing and computer vision. Face detection, character recognition, and vehicle estimation are just a few of the several well-researched object detection applications. There are several uses for object detection, including retrieval and surveillance. This study presents a range of essential concepts used in object identification with the use of the Python 2.7 OpenCV library, enhancing the efficacy and accuracy of object recognition..[3]

III. SYSTEM REQUIREMENTS

The hardware and operating system requirements for this software are listed here because it is software and must run on hardware and operating systems. Any version of Windows, Linux, or Mac OS can be used, hence it is platform-neutral. Any version of Windows, Linux, or Mac OS can be used, hence it is platform-neutral. Vscode community, for this to work, you must have Python installed on your PC.

We used a variety of cutting-edge technologies for this project, each of which will be examined in this chapter along with a full explanation of why it was chosen. The project's modules and features will be project's modules and features will be explanation. Let's first examine the language utilized in this project, though. We selected Python because it is a very recent language and has a lot of capabilities like machine learning and computer vision.

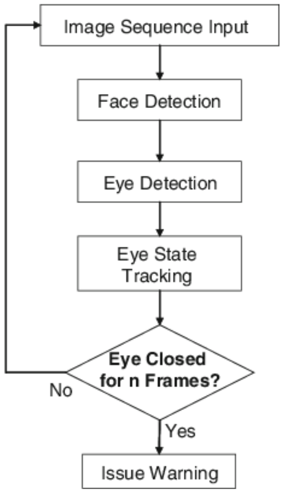
IV. RESEARCH METHODOLOGY

In the past, the facial detection was not prominent and if it was carried out, it was not accurate. So, the previous model of drowsiness detection was of average performance. At that time multiple problems were loaded these systems, one of the most important was the accuracy of the facial detection algorithm. Because of which the performance of the system was heavily compromised. So now in this model, the facial detection process is carried out by the facial detection algorithm,[4]shape_predictor_68_face_landmarks.dat which is pre-trained by the Dlib model for face landmark detection. Dlib has excellent Face Detection and Face Landmarks algorithms built in. Also, it provides pretrained model for facial landmark detection. The ROI (Region of Interest)that is the eyes of driver is created using the face_utils of the imutils module of Python.The Euclidean distance for eye ratio is calculated using the distance of the scipy.spatial module. The alarm sound is carried out by the pygame module which is initiated when driver is detected to be drowsy or sleepy.

IMPLEMENTATION

The thorough design is actually converted into functional code during the project's implementation phase. The phase's goal is to convert the design into the best possible solution in an appropriate language used in programming. This chapter goes over the project's implementation elements and gives details on the programming language and development environment that were used. In addition to giving a summary of the primary components of the project and their orderly progression, it also does so. Following are the duties needed for the implementation phase: Investigation of the system and its limitations. The creation of transitional methods. An assessment of the changeover procedure. Making the best decisions possible when choosing the platform. Selecting the right language for application development.

DROWSINESS DETECTION SYSTEM DESIGN



Driver drowsiness detection systems identify sleepy or drowsy drivers and awaken them to prevent accidents. Traditional systems make use of a variety of sensors, including a stress sensor to determine the driver's grip on steering wheel and other sensors to determine the driver's physiological state, including temperature, blood pressure, and heart rate. To detect eye closure, we do, however, provide a relatively straightforward camera-based implementation of DDD. The DDD system finds a human face first, then eyes, and finally tracks the status of the eyes. If closed eyes are present in more than n consecutive frames (n is often close to 10), it issues a warning.

V. SYSTEM ARCHITECTURE

This concept employs a webcam to capture live video feed during the operation of a heavy vehicle (including an automobile). This data will be utilized as an input forthe driver's sleepiness detection system. The feed will be processed using the OpenCV module, and Dlib will be used to identify the landmarks, in this example, the eyes. When the eye closure is measured and the eye aspect ratio is calculated using the Euclidean distance formula, a warning is generated if the value is less than a predetermined threshold value, and eventually an alarm is activated.

VI. WORKING PRINCIPLE

Use a webcam to capture a picture as input: We'll use a camera to capture images. We employ the cv2 technique offered using OpenCV.Use the function VideoCapture(0) to access the camera and set the capture object cap.The picture is kept in a frame variable after using read() to read each frame.Create a Region of Interest (ROI) and Detect a Face in the Image: The OpenCV object detection method accepts greyscale images as input.Eyes are defined from the Model by using face_utils.FACIAL_LANDMARKS_68_IDXS["EYE_L/ R"].Eyes Open or Closed classification by the classifier: Whether or not the eyes are open will be determined by the classifier using the eye aspect ratio. Simple Euclidean math is used to do this. Score Calculation to Determine Drowsiness in a Person We will use the score essentially as a value to calculate the subject's duration of eye closure. Using the cv2.putText() function, we are displaying the outcome on the screen in real time, along with the user's status.

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

VII. SIMULATION RESULT

Since the model for facial recognition in this project has already been trained, there is no need for dataset training. The 4shape_predictor_68_face_landmarks.dat is used to find faces in frames and images. The result is dependent on the condition of the algorithm's artificial object, the driver's eyes. Depending on how the eyes are placed, the following outcome is produced.

Test case	Eye status	Eye Lid Position	Result
Case1	Not detected	Open	No Alarm
Case 2	Detected	Open	No Alarm
Case 3	Not Detected	Close	No Alarm
Case 4	Detected	Close	Alarm

Table 1 Model Testing

RESULTS OF DROWSINESS

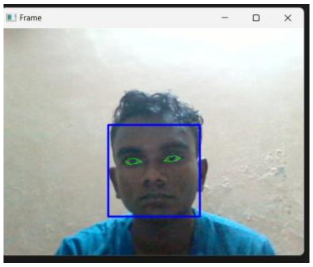


Fig 1 Perfect Eye Detection

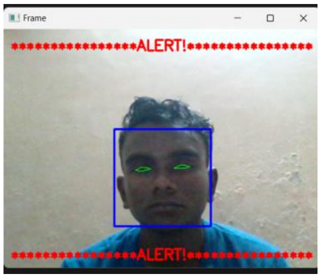


Fig 2 Generation of Alerts and Detection.

The optimal placement of the face and eyes is seen in the illustration up above. When the eye status is detected and the eyelid position is open, this is the situation that is defined. The counter value is generated perpendicular to the image frame, and when the eyes are closed for more than 20 seconds (the threshold value set forth in the model), an alert message and alarm are also generated. The alert generation case is shown in Figure 5.2.2 in the manner previously mentioned.

TESTING METHODOLOGIES

Upon completion of the implementation, A testing strategy should created and procededusing specifictest data set. While the goals of each test vary, they are all designed to ensure that the system's components have all been correctly integrated and are carrying out their assigned tasks. In order to verify that the product works, testing is done. The intended function in full. The organization's internal testing process serves as the final step in the verification and validation process. The main testing operations are focused on the analysis and modification of the source code. The list that follows contains the test cases that were run for this project. Each test case includes an explanation of the description, the actions to be taken, the anticipated outcome, the status, and screenshots. As part of the approach for software testing, numerous different types of testing methodologies or procedures are employed.

VIII. CONCLUSION

The driver's eye movement is monitored by the Drowsiness Detection Model, which can determine whether the driver is sleepy. The facial detection algorithm's inputs come from the facial recognition Dlib model, which has already been pre-trained. The area of interest is discovered utilising a model that considers the aspect ratio of the eye. Using the EAR function, the aspect ratio of the eye is determined.The detection counter's value must surpass the threshold value set in the driver code in order for the alert to be produced. The main objective of this initiative is to development is to lessen the amount of crashes brought on by drowsy drivers.This system can be enhanced and connected to the Internet of Things (IoT), in particular smart home controllers, allowing for real-time warning and mitigation while automating the alerting process and cutting out the human element.

IX. FUTURE SCOPE

This model's accuracy heavily depends on the camera's quality. If the driver's eyes are not readily visible for the detection, the detection's quality will decline. It might occur as a result of any form of obstruction between the eyes and the camera, including sunglasses or eyeglasses with light reflection. Furthermore, precision is hampered if the driver is not correctly facing the camera.By incorporating some defense mechanisms into the face recognition process, we can enhance this system. Photo and video attacks,which are simple and low-effort, are the most popular face recognition mocking attacks. Awide range of applications for this project are possible thanks to technological advancements like the ability to combine tiny size and high computing power.

REFERENCES

[1] S. Gupta Real-time and static image facial expression recognition. In the 2018 ICISC, the second international conference on inventive systems and control, 2018

[2] V. K. Sharma (May 2019). construction of a face recognition system. A global conference on intelligent computing and control systems was held in 2019.

[3] Study on Object Detection Using Open CV Python by Bhumika Gupta, Ashish Chaube, Ashish Negi, and Umang Goel was published in the International Journal of Computer Applications Foundation of Computer Science (FCS). 2017.

[4] International Journal of Computational Vision and Robotics, Kartik Umesh Sharma and Nileshsingh V. Thakur, "A Review and an Approach for Object Detection in Images,"2017.

[5] Richard Socher, Brody Huval, Bharath Bhat, Christopher D. Manning, and Andrew Y. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification", Computational Intelligence and Communication Networks International Conference, 2018.