IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Drowsiness Detection System

SARTHAK K SRIVASTAVA*

Dr. AMOL K.KADAM**

***STUDENT, DEPARTMENT OF COMPUTER ENGINEERING BHARATI VIDYAPEETH DEEMED UNIVERSITY COLLEGE OF ENGINEERING PUNE**

****ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING BHARATI VIDYAPEETH DEEMED UNIVERSITY COLLEGE OF ENGINEERING PUNE**

Abstract -

We tried to figure out how we might help to improving road safety and came up with the solution of constructing the "ANTI-DROWSY SYSTEM," which will not only monitor the driver's eye movements but will also alert the driver based on the data. has a sleepy feeling This project is built on the notion of using the Raspberry Pi as a microcontroller, which will serve as a small computer on which we will instal the many modules essential for the project's success. This project also employs python programming, as well as the open cv platform for tracking face changes and the dlib library for storage. a library Using the "EDR" eye distance ratio in the open cv platform, we can figure out the movement of the eye using this technique.

***Keywords:-* Anti Drowsy System ,Raspberry Pie, Open Cv, Image Processing , Dlib , EAR.**

I. INTRODUCTION

As the demand for car ownership grows, so does the danger of being involved in a car accident. There are several elements that contribute to road accidents, but one of the most important aspects that we will explore is tiredness while driving, which has the potential to be devastating and catastrophic[1].

Drowsiness while driving is defined as a scenario in which a driver suffers weariness, which leads to sleepiness, making it difficult for the driver to focus on the road and perhaps resulting in a crash, endangering the driver's life.

Drowsiness or exhaustion while driving is deemed lethal because it impairs the driver's decision-making abilities and delays reaction time, causing the driver to incorrectly estimate incoming traffic and resulting in a crash[2].

As a result, we have a great urge to tackle this difficult problem utilising a simple computational technique.

II. PROPOSED SYSTEM

A. Stages in the proposed system

In the first phase, we'll use raspberry pie to set up the camera in the automobile so that it can easily detect our faces and apply landmark localization to monitor the impressions of the eyes. [3]

Using OpenCV, dlib, and Python, we may create our own drowsiness detector in the second part of the project. The method would be used to measure the impres' dimensions and changes.

In the third stage, we'll test how well our project works in a real-time situation or under real-world conditions. As we'll see, the drowsiness detector works well and dependably informs us when it starts to "snooze." In this situation, if the driver becomes tired, an alarm will sound, alerting the driver to the fact that he or she is not in a driving condition.

B. Working of the model

Working of the model is illustrated with the following diagram:-

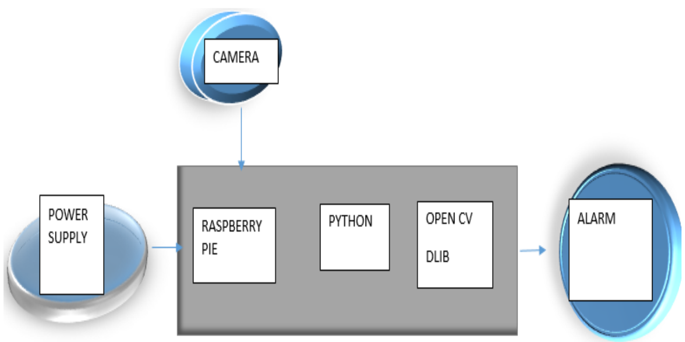


FIG1. BLOCK DIAGRAM

The flow of our proposed system's operation could be depicted using a diagram. [5]

1. We needed three essential hardware components to make an anti-drowsy system, which will act as pillars for our system: - 1 Raspberry Pi — This will serve as a control surface for our device, acting as a little computer.
2. Camera: This will be used to monitor the driver's eye impressions, allowing Raspberry Pie to determine whether or not he is tired.
3. The alarm will snooze as soon as computational [6] is completed.

Our approach is based on monitoring the emergence or disappearance of sclera in the eye, thus we must first estimate the "EDR," or proportionality of the distance between the eyes' landmarks. Unlike the previous method, it produces more accurate results when compared to other computational techniques that divide the eye into six parts for study.

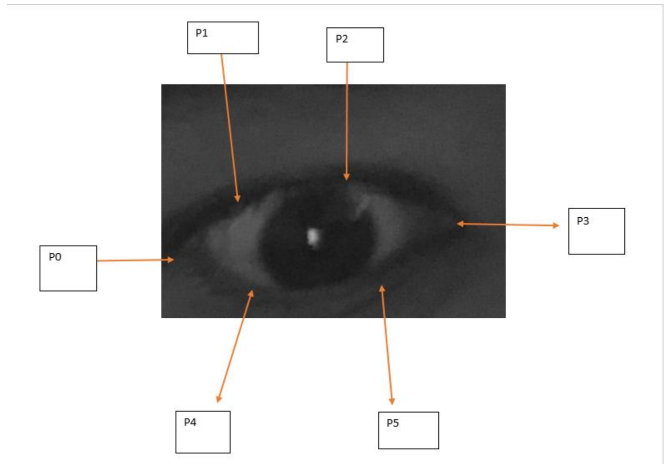


FIG2. COORDINATES OF EYES

The coordinates provide the eye's height and width, and we can get the relationships from them using the distance formula[8]. The major goal is to set off the alarm when all of the coordinates of the eye's facial landmarks are on a linear straight line. This may be accomplished by employing the following relations, which give us the value of EDR[9].

$$EDR= ||p1-p4||+||p2-p5||/2||p0-p3||$$

The above equation can be described as follows: the numerator represents the intersection point of the vertical eye, while the denominator represents the intersection point of the horizontal eye. This equation yields the crossing point, which can be used to determine whether EDR is zero or not.

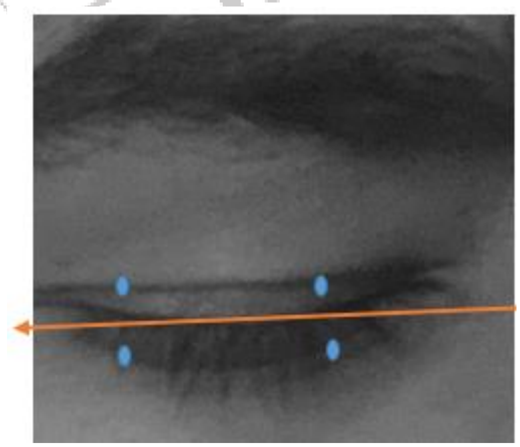


FIG3. EYE ASPECT RATIO

If the driver closes or blinks, the EDR approaches zero, indicating that the eye is closed. If we use this timing and compute that, for example, if the driver's eye is closed for a certain amount of time, consider "x seconds" and if the duration reaches "y seconds," an alarm will be triggered based on the duration that the driver had closed his eyes.Let's now understand how our algorithm works step by step.

Step 1: Take an image from a camera as an input.

We'll use a webcam to capture photographs as input. So, in order to gain access to the webcam, we created an infinite loop that captures each frame. To access the camera and set the capture object, we utilise OpenCV's cv2.VideoCapture(0) method (cap). Each frame is read with cap.read(), and the image is saved in a frame variable.

Step 2: Create a Region of Interest by detecting a face in the image (ROI)

To find the face in the image, we must first convert it to grayscale, as the OpenCV object detection algorithm only accepts grayscale images as input. To detect the objects, we don't need colour information. To detect faces, we'll use the Haar cascade classifier. Our classifier face = cv2 is set with this line.for (x,y,w,h) **in** faces:

Step 3: Use the ROI to find the eyes and input them to the classifier. The process for detecting eyes is the same as for detecting faces. We first configure the cascade classifier for eyes in leye and reye, then use left eye = leye to detect the eyes. detectMultiScale(gray). From the complete image, we must now extract only the eyes data. This can be accomplished by first removing the eye's boundary box .

Only the eye's image data is stored in l eye. This will be sent into our CNN classifier, which will determine whether or not the eyes are open. Similarly, the right eye will be extracted into r eye.

Step 4 – The classifier will determine whether or not the eyes are open.

The ocular state is predicted using a CNN classifier. We must do specific procedures in order to input our image into the model, as the model requires the correct dimensions to begin with. First, we use r eye = cv2.cvtColor(r eye, cv2.COLOR BGR2GRAY) to convert the colour image to grayscale. The image is then resized to 24*24 pixels.

We used model = load model('models/cnnCat2.h5') to load our model. Now we use our model lpred = model.predict classes(l eye) to predict each eye. If the value of lpred[0] equals 1, the eyes are open; if the value of lpred[0] equals 0, the eyes are closed.

Step 5: Calculate a score to see if the person is sleepy.

The score is just a number that we'll use to figure out how long the person has been closed-eyed. As a result, if both eyes are closed, we will continue to increase the score, however if both eyes are open, we will drop the score. We're using the To draw the result on the screen, use the cv2.putText() function, which will show the person's current state in real time. cv2.putText(frame, "Open," (10, 20), font, 1, (255,255,255), 1, cv2.LINE AA)

A threshold is established, for example, if the score exceeds 15, it indicates that the person's eyelids have been closed for an extended amount of time. This is when we use sound to beep the alarm. play()

C: METHODOLOGY

The goal of this systematic review research is to identify and classify the best methodologies, metrics, tools, and classification methods for detecting tiredness in drivers. All of the information acquired from primary studies has been organised. We collect important information and identify research gaps in existing research studies after completing a systematic review of empirical studies.

To extract meaningful and relevant information from the buckets of data, a systematic and well-organized search is undertaken. Research articles, case studies, the National Highway Traffic Safety Administration's report on traffic accidents, and the reference lists of connected publications were all thoroughly investigated. Websites with information on road safety, the consequences of driver fatigue, the causes of fatigue, and sleepiness detection techniques

• (Biological* OR physiological*) AND (Drows* OR Fatigu*)

• (Drowsy* OR Fatigued*) AND (Automobile* OR Vehicle*)

• (Behavioral*) AND (Drows* OR Fatigu*)

• (Classif**) AND (Drows* OR Fatigu*)

• (SVM*) AND (Drows* OR Fatigu*)

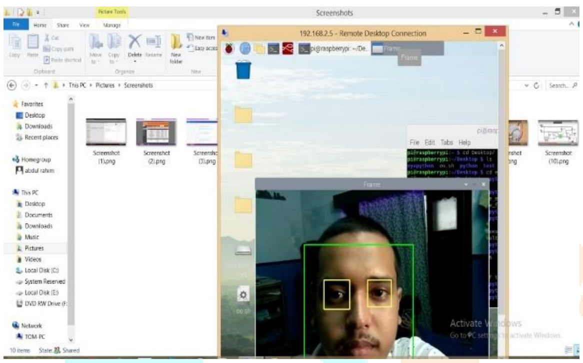
• (CNN*) AND (Drows* OR Fatigu*)

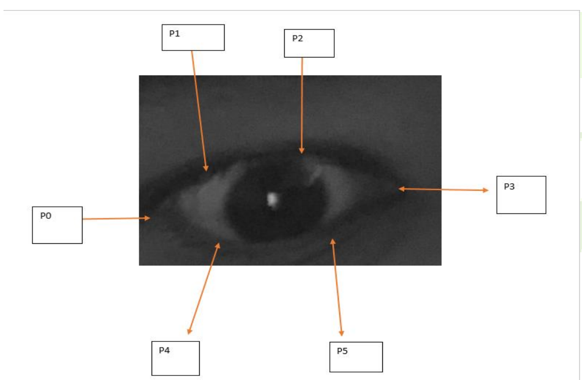
• (HMM*) AND (Drows* OR Fatigu*)The initial search yielded 1020 research publications, of which we chose 105 based on titles that were relevant to our study. The abstracts of selected papers are evaluated, and 74 additional research papers are extracted. The retrieved papers are then thoroughly examined, with 41 research papers being selected as our major study. Figu illustrates the entire selection procedure.

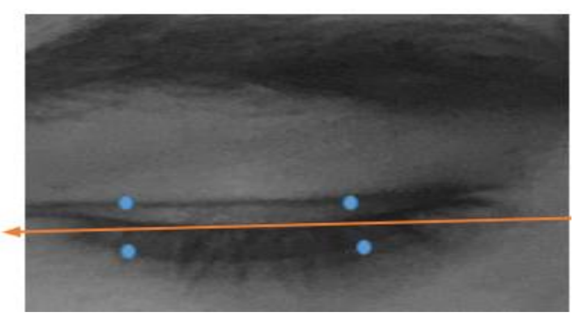
The answers to key quality control queries are sought by a thorough examination of whole research articles. Quality assessment questions in current systematic research include: a) Is the publication relevant to the research domain? & b) have the articles been published in a reputable journal or at a reputable conference? The classification of sleepiness detection systems is based on drowsiness metrics and classifications.

The current state of drowsiness systems is summarised in tabular form. Answers to research questions are generated, as well as the benefits.

D: OUTPUT







E: FUTURE SCOPE

Future research may concentrate on the use of external factors for tiredness measurement, such as vehicle states, sleeping hours, weather conditions, mechanical data, and so on.

Driver sleepiness is a major hazard to highway safety, and it is especially problematic for commercial motor vehicle operators. This major safety issue is exacerbated by 24-hour operations, high annual mileage, exposure to hazardous environmental conditions, and demanding work schedules. Monitoring the driver's tiredness and vigilance levels, as well as providing feedback on their performance.

• The device's software technology should be optimised for sensitivity and specificity. False negatives should be avoided by detecting reduced awareness levels accurately and reliably. False positives should be reduced by identifying safe driving and operator vigilance in an accurate and reliable manner.

• The gadget should be durable, dependable, and capable of running continuously for long periods of time, such as a shift. The expense of maintenance and replacement should be kept to a minimum.

• The device should be able to monitor driver or operator behaviour in real time, and it should be able to operate accurately in a variety of operational situations, including day, night, and lighted conditions. Conditions in the operator cab, such as humidity, temperature, vibration, noise, and so on, should not impair accuracy.

• Regardless of the clear safety benefits that fatigue detection systems provide, the technology's acceptance is contingent on the operator's perception that the benefits outweigh the costs. The following factors have an impact on user acceptance:

• Ease of use: the technology should be simple to use and intuitive to understand. In all operating settings, the operator should be conversant with the capabilities, constraints, and operational factors. Operators of varying cognitive and physical abilities should be able to interpret the device output readily and properly. The operator's view of the road and other controls should not be obstructed in any way.

• Perceived value: the operator should see the technology as helping to make driving safer and more aware, but not to the point of over-reliance. The device should help the operator with his or her individual fatigue management plan. It should be obvious that the equipment is fully safe to use and has no bad impact on the operator's health.

• Advocacy: The willingness of operators to purchase and endorse the technology is a vital component of user acceptance. When the intended users – operators, fleet managers, trucking associations, safety departments, and others – endorse the device's perceived safety benefits, market uptake will improve.

III. CONCLUSION

As the demand for car use grows, so does the danger of being involved in a car accident. There are several elements that contribute to road accidents, but one of the most important factors that we will cover is tiredness while driving, which has the potential to be lethal and catastrophic.

So, our proposed model is an attempt to handle this complex problem in a timely and accurate manner, which will not only lower the likelihood of a catastrophic event, but will also inform the driver whether or not he is fatigued.

REFERENCES

[1] Vesselenyi1, S. M. (2017). Driver drowsiness detection using ANN image processing. *iop*.

[2] A. Subbarao, K. (2019). Driver Drowsiness Detection System for. *ijitee*, 5.

[3] Galarza, E. &. (2018). Real Time Driver Drowsiness Detection Based on Driver's Face Image Behavior Using a System of Human Computer Interaction Implemented in a Smartphone. *research gate*.

[4] Juhua Zhu, B. L. (2003). General Illumination Correction. *ieee*.

[5] Mohamad-Hoseyn Sigari, 1. M. (2013). A Driver Face Monitoring System for Fatigue and. *ijvt*, 11.

[6] Nawal Alioua1, A. A. (2011). Driver's Fatigue and Drowsiness Detection to. *Springer-Verlag*, 8.

[7] Rahmati, M. A. (2018). Driver drowsiness detection using face expression recognition. *ieee* (p. 341). kuala lumpur: ieee.

[8] Rukhsar Khan, S. M. (2019). Human Drowsiness Detection System. *ijeat*, 4.

[9] Sontakke, M. K. (2015). Efficient Driver Fatigue Detection and Alerting System. *ijsrp*, 4.

[10] Xiao-Gang JIANG, J.-Y. Z.-H.-H. (2005). FPGA Implementation of Image Rotation Using Modified. *ieee*.