

Drowsiness Detection System

Dr.CH.Ramesh¹, Andhavarapu Priyanka², Gudla Likitha³, Gujju Priya Devi⁴, Sampathirao Sai Kiran⁵

¹Professor, Department of computer science engineering, Aditya institute of technology and management, Tekkali, Srikakulam, Andhra Pradesh

^{2,3,4,5}Students from final year, Department of computer science engineering, Aditya institute of technology and management, Tekkali, Srikakulam, Andhra Pradesh

Abstract—Drowsiness detection is a safety technology that can help drivers avoid accidents caused by falling asleep while driving. One of the potential applications of intelligent car systems is drowsy driver detection. Drowsiness detection mostly relies on assumptions about blink rate and ocular closure. We use machine learning to identify genuine human behaviour during sleepiness episodes in our study. We'll use OpenCV in this Python project to collect photos from a camera and feed them into a Deep Learning model that will classify whether the person's eyes are 'Open' or 'Closed.' A convolutional neural network is a sort of deep neural network that excels at picture classification.

IndexTerms: Open CV, Haar cascade classifiers, Blink rate.

INTRODUCTION

Every day, thousands of people spend hours driving on highways and country roads. Truck drivers, bus drivers, and taxi drivers all rely on driving for a living, and regular individuals also drive for long periods of time. As a result, individuals are more likely to fall asleep while driving as a result of stress or lack of sleep. Individuals, communities, and countries are all affected negatively by road traffic injuries and deaths. They entail large expenses to sometimes overburdened health-care systems, occupy precious hospital beds, consume resources, and result in severe productivity and prosperity losses, with far-reaching social and economic effects..

The major goal of this research is to create a non-intrusive technology that can detect human sleepiness and deliver an early warning. When driving long distances, drivers who do not take regular stops are at risk of becoming drowsy, which they often do not detect early enough. Eye closure, head nodding, and brain activity are examples of real-time drowsiness

behavior linked to tiredness.

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements:

- Processor : Any Processor Above 500MHz
- Ram : 2GB
- Disk space : 100mb above
- Input device : Webcam
- Output device : Speaker

Software Requirements:

- Operating System : Windows or any other OS
- Programming : Python 3.9
- Environment : Anaconda Spyder

Library used:

- Open CV
- Dlib
- Pygame
- Scipy
- Numpy
- Imutils
- Argparse

Raspberry Pi 3 Model B+ :

The Raspberry Pi 3 Model B+ is the most latest change to the Raspberry Pi 3 family, with a 64-bit quad-core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE functionality through a separate PoE HAT. The dual-band wireless LAN has modular compliance certification, which allows the board to be integrated into end devices with much reduced wireless LAN compliance testing, decreasing both cost and time to market.



Webcam:

A webcam is a video camera that transmits live video or images through a computer network, like the Internet. Webcams are often small cameras that are either built into the hardware, attached to a user's monitor, or rest on a desk. During a video chat session with two or more participants, webcams can be used for live audio and video chats.

Users utilizing webcam software can record videos or stream them online. These streams typically use compressed formats since streaming video over the Internet uses a lot of bandwidth. A webcam's maximum resolution is also less than that of the majority of handheld video cameras since larger resolutions would be compressed during transmission. Webcams have a lesser resolution than most video cameras, making them more expensive, but the quality is still decent enough for video chats.



Buzzer:

A buzzer is an efficient component to include the features of sound in our system or project. It is an extremely small & solid two-pin device thus it can be simply utilized on breadboard or PCB. So in most applications, this component is widely used. There is two kinds of buzzers commonly available like simple and readymade. Once a simple type is power-driven then it will generate a beep sound continuously. A readymade type looks heavier & generates a Beep.

A buzzer is a useful tool for incorporating sound characteristics into our project or system. It is a very compact and sturdy two-pin gadget, making it easy to use on a breadboard or PCB. As a result, this component is frequently employed in most

applications. There are two types of buzzers that are frequently found: simple and ready-made. When a simple type is powered up, a continual beeping sound is produced.



METHODOLOGY

Many applications, including human-computer interfaces, iris recognition, driver drowsiness detection, security, and biology systems, require the preliminary step of eye detection. This study provides a description of template-based eye detection. There are correlations between the template and several areas of the face image. The area around the eyes is the part of the face that correlates most strongly with the template. It's simple and straightforward to use the procedure. Through a variety of simulation outcomes, the method's efficacy is shown in both open-eye and closed-eye scenarios. This paper suggests a brand-new, basic eye detection method. To detect the eye region from a face image, an eye template is employed. Using cross correlation approach, the template is matched with the ocular region. The method doesn't call for prior understanding of the eye or difficult calculations. It is a simple process that hardware can easily implement. There are different types of methodologies that have been developed to find out if the person is in state of drowsiness or not.

Physiological level approach:

This method uses electrodes to collect data on brain activity, heart rate, and pulse rate in an intrusive manner. The heart rate variability and various sleepiness situations are calculated using ECG. the relationship between several signals, including ECG and EEG (electroencephalogram), and EMG (electromyogram) are made and then the output is generated whether the person is drowsy or not.

Behavioral based approach:

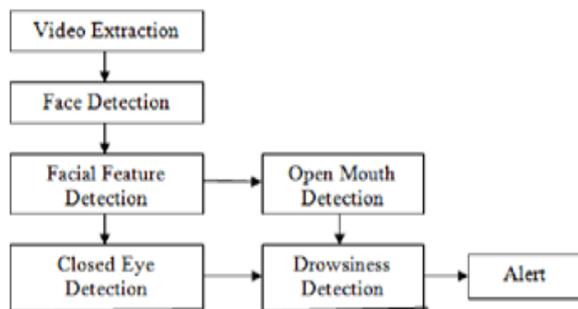
Through the use of a camera, a person's eye blinking frequency, head posture, etc. are observed, and if any of these signs of drowsiness are seen, the individual is informed.

Proposed system:

For many years, image processing has been used to prepare films and images for various real-time applications. It became considerably simpler to construct real-time applications that can function similarly to people with much greater accuracy and less expenditure with the advent of new processors with high processing capabilities and high definition cameras. To accomplish this, a variety of tools and libraries have been developed to aid researchers and developers in quickly and easily developing their systems.

The preparation of films and images for various real-time applications has long been accomplished through image processing. With the introduction of new processors with strong processing capabilities and high definition cameras, it became noticeably simpler to create real-time apps that can perform similarly to people with much greater precision and less expense. To achieve this, a variety of tools and libraries have been created to assist academics and developers in swiftly and easily constructing their systems. A system that detects drowsiness and delivers data to the user so they may react to the events are captured, processed, and recognised A series of facial photographs, along with the study of eye movements and blink patterns, are used to detect exhaustion. The signs of driver weariness may be picked up early enough to prevent an accident by keeping a watch on the eyes. A series of facial images, along with the study of eye movements and blink patterns, are used to detect exhaustion. A common research topic is the analysis of facial images, which has applications in face recognition, virtual tools, and human identification security systems. Estimates of the Region of Interest (ROI) are made based on the eye's state (open or closed).

BLOCK DIAGRAM



WORKFLOW OF THE PROPOSED SYSTEM

In this project, we'll be gathering webcam photos using Open CV and feeding them into a Deep Learning model to determine whether or not the subject's eyes are "Open" or "Closed." For this project, the approach we'll use is as follows:

Step – 1: Take an Image from a Camera as Input: Using a webcam, we will take an image as input. We therefore created an infinite loop to access the webcam and capture each frame. We employ the cv2 technique offered by OpenCV. using the camera and setting the capture object with VideoCapture(0) (cap). We read each frame using cap.read(), then put the image in a frame variable.

Step – 2: Region of Interest (ROI) creation and face detection in an image We must first convert the image to grayscale in order to recognise the face in it because the OpenCV object detection algorithm only accepts grayscale images as input. To find the objects, we don't need to identify their colours. Haar cascade classifier will be used to find faces. To set our classifier, use this line. face = cv2.CascadeClassifier(' path to our haar cascade xml file'). Then we perform the detection using faces = face.detectMultiScale(gray). It returns an array of detections with the x, y, height, and width of the object's boundary box. We can now draw boundary boxes for each face as we iterate through the faces. 1. For the faces of (x,y,w,h): 2. cv2.rectangle(frame, (x, y), (x + w, y + h) (100,100,100)

Step – 3: To feed the classifier, identify the eyes in the ROI: Eyes are detected using the same method as faces. In order to detect the eyes, we first set the left eye = l_eye flag on the cascade classifier for the eyes in l_eye and r_eye, respectively. detectMultiScale(gray) Only the eyes' data from the entire image must now be extracted. This may be done by removing the eye's boundary box, and then using the following code, we can extract the eye's picture from the frame. 1. l_eye equals frame

[y: y+h, x: x+w] All that is contained in l_eye is the eye's picture data. This will be fed into our shape predictor classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into r_eye.

Step – 4: We are employing a shape predictor classifier to predict whether the eyes are open or closed in step 4 of the classification phase. We must carry out the following activities in order to input our image into the model because the model requires the proper starting dimensions. First, we convert the color image into grayscale using

`r_eye=cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)`
 Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images `cv2.resize(r_eye, (24,24))`. We normalize our data for better convergence `r_eye = r_eye/255` (All values will be between 0-1) Expand the dimensions to feed into our classifier. Now we predict each eye with our model `lpred = model.predict_classes(l_eye)` If the value of `lpred[0] = 1`, it states that eyes are open, if value of `lpred[0] = 0` then, it states that eyes are closed.

Step – 5: Calculate Score to Check whether Person is Drowsy The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using `cv2.putText()` function which will display real time status of the person. `cv2.putText(frame, "Open", (10, height-20), font, 1, (255,255,255), 1, cv2.LINE_AA)` A threshold is defined for example if score becomes greater than 15 that means the person's eyes are closed for a long period of time. This is when we beep the alarm using `sound.play()`.

RESULT



CONCLUSION

In this Python project, we have built a drowsy driver alert system that you can implement in numerous ways. We used Open CV to detect faces and eyes using a haar cascade classifier. This work will be extended on real time video stream to recognize drowsiness of driver and give him the alert. The system is capable of accurate positioning eye point. Using four parameters of eye states can effectively detect the driver's fatigue status. In order to improve the accuracy grade, our system should be using some other methods as a supplementary means, such as

- Road tracking,
- Head position,
- The rotation rate and the grip force of the steering wheel, which are the main directions to improve our system accuracy.

REFERENCE

- [1] National Highway Traffic Safety Administration. Advanced Countermeasures for Multiple Impairments, February 2014, Available at: <http://www.nhtsa.gov>
- [2] <https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/>
- [3] Chen Terrence, Zhou Xiang Sean, Comaniciu Dorin, Huang Thomas S. Total Variation Models for Variable Lighting Face Recognition [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(9):15191524
- [4] <http://mrl.cs.vsb.cz/eyedataset>
- [5] <https://www.tutorialspoint.com/python/index.html>
- [6] https://www.tutorialspoint.com/object_oriented_python/object_oriented_python_libraries.htm
- [7] https://www.tutorialspoint.com/python3/python_modules.htm
- [8] https://www.tutorialspoint.com/opencv/opencv_overview.htm