Drowsiness Detector using HaarCascades Classifier

Ayyagari Venkata Sai Rohit[1], Chirag Chinvar[2]

[1]Student, Dept. of Electronics and Communications Engineering, Dayananda Sagar College of Engineering, Karnataka, India
[2]Student, Dept. of Electronics and Communications Engineering, Dayananda Sagar College of Engineering, Karnataka, India

---***---

Abstract - *Safety of passengers and the driver in a vehicle is becoming of vital importance because of the increasing number of road accidents. The number of road accidents because of a drowsy driver is among the major causes of car crashes. Not only does it cause damage to the vehicles but also causes a lot of fatalities and a monetary loss of thousands of dollars. Hence there arises a need for an effective method to detect and warn drowsy drivers. Several methods are being proposed to detect Drowsiness. This method makes use of a predefined method called HaarCascade which is a class in the OpenCV library. This method involves the raspberry pi to refer to a predefined image stored in it and compare it with the live stream, then if the live stream captured image doesn't match with the predefined image it triggers an alarm inside the car which alerts the driver until he comes out of his drowsy state. For the other vehicles on the road not to be in danger because of one drowsy driver, when a driver feels drowsy it also flashes the double indicator on the outside to indicate to the co-drivers to be cautious. This method proves to be cost-friendly because of the use of common components like a pc webcam, raspberry pi. A buzzer to alert the driver and LED to act as double indicators to simulate real-time conditions. OpenCV is a very versatile library that is available in python due to its rich set of inbuilt functions to perform very complex operations on images.*

Key Words: HaarCascade, OpenCV library, raspberry- pi, double indicators, python

1. INTRODUCTION

Computer vision is a technology that facilitates the computer to see the outside world. Computer vision is widely used for its security applications but nowadays finding its use in many other sectors. According to DOCTOR'S magazine, the most common method to detect the drowsiness is by monitoring the eye blink pattern of the driver. A webcam is kept in front of the driver in a way it doesn't obstruct the driver while driving. The webcam detects his face and then starts monitoring the eye blink pattern of the driver, it also indicates that his eyes are open. When the driver, starts to feel drowsy, the eye blink pattern of the driver changes from the usual pattern and even then, if he continues to be drowsy for about 3 seconds more, the camera detects it and sends a message to the raspberry pi and the raspberry pi verifies it and then triggers an alarm/buzzer and also displays that the eyes of the drivers are closed and he is feeling drowsy. This alarm/buzzer starts ringing and is heard until the driver wakes up from his drowsiness. To facilitate the working of this device during night times an infrared LED torch can be used which is invisible to the naked eye and the light can only be detected by the camera, hence there is nothing to distract the driver. The method used to detect the eye blink pattern is referred to as HaarCascades. HaarCascades is a method or class in the open CV library where a reference image of a person's eyes open and closed is stored. Then in the HaarCascades class, we load the image and the OpenCV program is instructed to refer to that image and to give the relevant output. OpenCV was preferred over MATLAB because OpenCV is open source and the number frames processed per second is more in OpenCV. The brain of this device is the raspberry pi B. Raspberry is a micro CPU that runs on Linux It has 40 GPIO pins which help to communicate with the outside world. Its key features include a 1GB RAM, clocked at 1.2GHz, consisting of the ARM A53 architecture. The webcam used to detect or monitor the driver's eyes can be any general-purpose webcam. A Pi camera would be a better option.

2. LITREATURE SURVEY

Concerning drowsy driver detection, several methods have been proposed. One of the many methods proposed was by Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt "Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio "who detected drowsy drivers using their eye-opening and eye closure ratio. They used an android application and the driver's phone camera to detect drowsiness. The phone camera keeps monitoring the driver's eye and sends the information to the local server which further evaluates the EAR. If any case the EAR falls below 0.25 it sets an alarm indicating the driver is fatigued.[2]

In another method proposed by Alireza Khammar, Mohsen Poursadeghiyan, Adel Mazloumi, Gebraiel NASL, Mohammad Mehdi Baneshi, and Mohammad Hossein Ebrahimi "Using Image Processing in the Proposed Drowsiness Detection System Design" they monitor the driver's eyes using a camera kept in front of the drivers' eyes. The camera captures the driver's eyes in real time and then the location of the eyes is calculated by the algorithm Viola-Jones, then for analysis, the image is converted to grayscale and then into binary. The technique adopted was if the eyes were open there would be a large number of dark pixels on the eye when it is compared to a closed eye. Once the controller detected the eyes were closed it calculated other parameters like blink duration, blink frequency, number of long blinks which was fed to a neural network which calculated the level of drowsiness.[3]

One of the methods proposed by Steffin Abraham, Tana Luciya Joji, Sivaram M, D Yuvaraja in "Enhancing Vehicle Safety with drowsiness detection and collision avoidance" the technique they adopted to monitor drowsy drivers is by measuring the temperature of the auditory canal which in turn gives the temperature of the brain and also uses image processing to find out the driver's eye blinking rate to detect drowsiness. Once a driver is detected to be drowsy it sends a loud alarm in the car to wake the drive up. It also consists of an automatic lane change system and an automated braking system to prevent collisions and cause harm to other drivers. Besides, it also comes with a forward and sideward collision technology and is also equipped with an automatic headlight dimmer for the headlights to be used effectively, especially during night times.[4]

A method proposed By Jose Maria Armingol, Marco Javier Flores, and Arturo de la Escalera , in "Real-Time Drowsiness Detection System for an Intelligent Vehicle" In this the beginning approach The Viola-Jones algorithm, considered a standard approach for detecting facial features, is used to detect facial features. The stand apart feature of this paper is the detection of facial features when the face is not in the camera axis, this helps in avoiding many false alarms that may arise due to non-existence of the face in the camera axis. It uses a neural network to detect the face even if it's outside the camera features. The tracker is considered very efficient because of the rich dataset which consists of images of all possible cases of head orientation etc. To solve the problem of varying color images, different eye orientations, etc., they take the support of a vector machine that consists of certain pattern classifiers. They make use of the eye blinking frequency towards the end it activates an alarm to measure the drowsiness level if the device senses that the eye is closed for more than 0.25 seconds. [5]

3. ALGORITHM

3.1 Local Binary Pattern classifier

Local binary pattern or LBP is an alternate method to detect facial feature recognition, object detection, etc. The first step in an LBP is converting it to a grayscale image. The image is converted to a set of grids of 3x3 whose value is determined by $\sum_{n=0}^{7} s(i_n + i_c)2^n$.

Where i_c - Central Pixel value, i_n- Neighbour Pixel value, $s(z) = \begin{cases} 0, z < 0 \\ 1, z \geq 0 \end{cases}$

The calculated value is stored in a 2D series of the same height and width as the input image. Training of an LBP classifier requires less time compared to HaarCascades and CNN. It poses to have the least accuracy among the different methods used to image processing. The LBP classifier uses 4 parameters i.e.

1. Radius - It depicts the radius around the central pixel

2. Neighbor pixels - To the number of samples we provide to train the classifier

3. GRID X - The number of cells on the x-axis

4. GRID Y - The number of cells on the y-axis. [8] [9]

3.2 Convolution Neural Network

Convolution essentially implies the integration of two functions. An input image is converted to an array of pixels whose value depends on the resolution of the image and based on the resolution it sets various parameters like height width and dimensions. The process of detection involves an input image (the image to detect the features) which is converted to a matrix representation and this matrix is convoluted over the feature detector matrix (filter) to produce an activation map. This step reduces the size of the image and hence improves the processing speed. The main features are retained while some unwanted features might be lost. The CNN model takes the most time among LBP and HaarCascades to be trained but this comes with an added benefit of producing the most accurate results among the other object detection methods. Different operations can be performed on the input images by using different filters. [7] [10]

3.3 HaarCascades

HaarCascade classifier is another object detection algorithm which was proposed by Paul Viola and Michael Jones. It is a machine learning based model where a rich dataset of positive and negative images is fed to the classifier for it to be trained. Positive images are the objects of interest which our classifier is trained to detect and negative images are images of without the object of interest. HaarCascade classifier in comparison to other object detection techniques requires a significant amount of time to process an input image and give a desired output. Considering the processing time taken ideally the RAM consumption is on the higher side when compared to other methods like LBP, CNN. The output of the HaarCascade classifier is very accurate which makes it a widely preferred method for object detection. This model takes lesser time to be trained compared to the CNN model.

The technique used to identify a drowsy driver's eyes is HaarCasacades. HaarCasacades is an object detection algorithm that helps in the detection of objects from a picture, a video, or from a live stream. Open CV library has an inbuilt class that helps us use the HaarCascade feature. This feature is very widely used in the Face detection which can be used in the detection of any object if properly qualified.

This approach processes the image in certain stages-

3.3.1 Haar Feature Selection

This section involves the extraction of a region of interest or the selection of the object to be detected in other images. This is done by giving a large number of positive Images (an interest object) and negative images (images without an interest object) to train the model accordingly. It consists of a rectangle on the region of interest and also an adjacent rectangle.

3.3.2 Creation of integral images

This process is to boost the process of selecting the Region of Interest.

3.3.3 Adaboost Training

This section is responsible for selecting the required characteristics from all the positive and negative images provided by the user. It selects the best features and trains the model. In this stage, it also groups the objects to be detected in “strong classifiers” and the unwanted parts as a “weak classifier”. The output of this is stored as an XML file.

3.3.4 Cascade classifier

This is the final stage where it consists of many stages where each stage is made to slide over the positive and negative images. The classifier slides over a particular location and if the object is found it returns a true and indicates that the required object was found else when it returns a false it indicates that no object was detected and jumps to the next location and continues this process. [1]

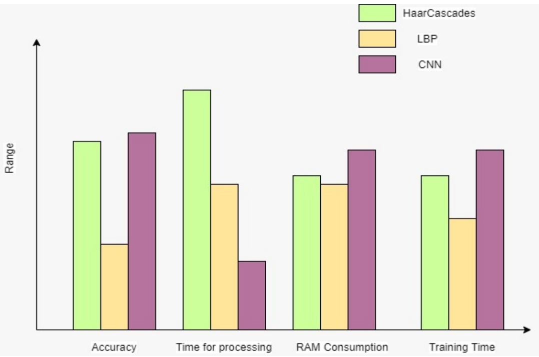

Chart-1: Comparison of the algorithms

4. IMPLEMENTATION

To test the accuracy of the device a driver's cabin setup was made and a camera was mounted on a stand to hold it and it used to detect the driver's eyes. Once the driver sat in front of the webcam i.e. in the driver's seat it detected his eyes and started printing eyes open on the raspberry pi console window. Once the driver started feeling drowsy i.e. his eye blink pattern changed from normal to being drowsy the raspberry pi detects this drowsy state of the driver and it starts to trigger a buzzer (in a real case scenario would be a speaker inside the car). This buzzer/alarm will be triggered until the driver is out of his drowsy state. It will also turn on LED's (in real case scenario would be the double indicators) to warn the other drivers in proximity for them to become more cautious. Once the driver returns to normal state the alarm stops ringing and the LED's stop flashing.

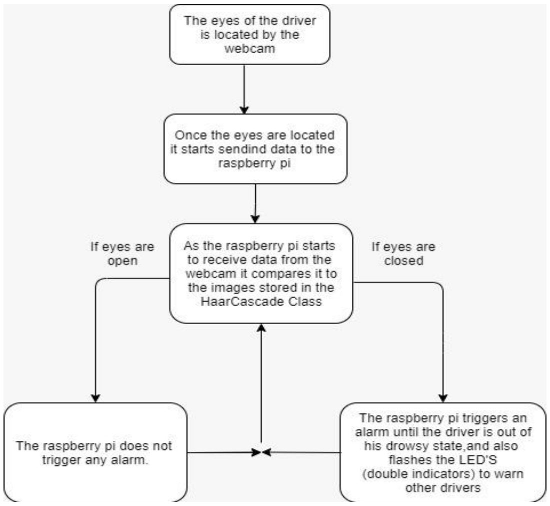
The eyes of the driver is located by the webcam
Once the eyes are located it starts sendind data to the raspberry pi
If eyes are open
As the raspberry pi starts to receive data from the webcam it compares it to the images stored in the HaarCascade Class
If eyes are closed
The raspberry pi does not trigger any alarm.
The raspberry pi triggers an alarm until the driver is out of his drowsy state,and also flashes the LED'S (double indicators) to warn other drivers

Fig. 1: Flow Chart of the working of the device

5. RESULTS

To test the accuracy of the device it was tested on 50 people out of which 16 people wore spectacles. The participant's age was in the range of 17-40 years. This test was conducted in a place with ambient lighting. The participants were made to sit on a chair in front of the camera. Once they sat and the raspberry pi detected their eyes it began to print eyes open on the console, once the participant closed their eyes, the raspberry pi detected the change in the eye blink pattern and then triggered a buzzer and flashed the LED's until the driver opened his eyes or he was out of his drowsy state. Out of the 34 participants who did not have spectacles the device gave proper alarms for 23 people but for the other 11 people gave false alarms at times. When it was tested with people wearing spectacles out of 16 people the device responded properly only to 9 people and gave many false alarms for the rest of the participants. There was a lag of about 2 seconds for the raspberry pi to detect the change in eye blink pattern, this was because of the use of an ordinary pc webcam and not a pi camera. This device would work in low lighting conditions with an infrared lamp. The infrared lamp cannot be detected by the naked eyes hence does not cause disturbance to the driver. After rigorous testing with many people, it was seen that there is a need to add additional parameters to detect a drowsy person and only the eye blink pattern was not sufficient to conclude a person to be drowsy.

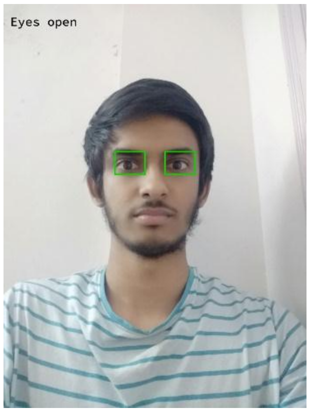

Fig. 2: Cascade working eyes open

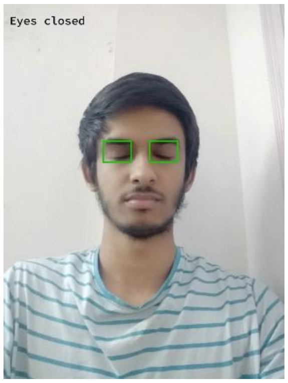

Fig. 3: Cascade working eyes closed

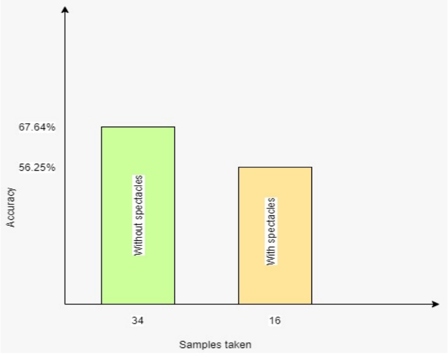

Chart-2: Accuracy metrics concerning glasses.

6. FUTURE SCOPE

The drowsy driver detector today is a novice model, improvements can be made on the connectivity front, we plan to use the concepts of the Internet of Things to give the device an online presence and help gather data with consent that will help us enhance the operational dynamics of the entire module. Government agencies that deal with the traffic branch of the society can use the product to monitor driving health and road safety and make improvements whenever necessary. Automobile industries can use this product when properly deployed as a key safety feature.

REFERENCES

[1] Haar Cascades Will Berger

[2] Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt "Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio". Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur – India.February 26-28, 2019.Posted: 14 Jun 2019.

[3] Mohsen Poursadeghiyan, Adel Mazloumi, Gebraiel NASL, Mohammad Mehdi Baneshi, Alireza Khammar, and Mohammad Hossein Ebrahimi "Using Image Processing in the Proposed Drowsiness Detection System Design. "Iran J Public Health.vol. 47, no. 9, pp.1370-1377. Aug. 2018.

[4] Steffin Abraham, Tana Luciya Joji, Sivaram M, D Yuvaraja "Enhancing Vehicle Safety with drowsiness detection and collision avoidance".International Journal of Pure and Applied Mathematics. Volume 120 No. 6 2018.ISSN- 2295-2310.

[5] Marco Javier Flores, Jose Maria Armingol and Arturo de la Escaler "Real-Time Drowsiness Detection System for an Intelligent Vehicle". Intelligent Vehicles Symposium, 2008 IEEE, July 2008, DOI: 10.11.09.

[6] Open CV documentation

[7] Prabhu Understanding of Convolutional Neural Network (CNN) Deep Learning.

[8] Rithika Chowta Object detection: LBP cascade classifier generation

[9] Adrian Rosebrock Local Binary Patterns with Python & OpenCV

[10] Kadir, Kushsairy & Kamaruddin, Mohd & Nasir, Haidawati & Safie Sairul & Bakti, Zulkifli. (2014) "A comparative study between LBP and Haar-like features for Face Detection using OpenCV ".335-339.10.1109/ICE2T.2014.7006273.

BIOGRAPHIES

Ayyagari Venkata Sai Rohit, a young, but highly skilled, AI / ML, VLSI, IoT, Deep Learning and Robotics engineer. He has a profound understanding of programming languages such as Java, Python, Verilog and VDHL to name only a few. Even his technology umbrella equips him with platforms like Arduino and MatLab. He was working on 7 projects, 6 internships, 4 research papers at the age of just 20, and attended hackathons and workshops won. In short, he's the culmination of a large variety of lessons.

Chirag Chinvar, a 19-year-old tech enthusiast, has explored the technological walk of life in his horizon. His comprehension of IoT, Embedded, Python, C language and computer vision is praiseworthy. His Drowsiness detector project won a bronze medal at an InSef-organized state-level festival and participated at several college-level tech fairs. At the end of the day, his go-doer goes on all sides with absolute determination to get the job done.