MANIPAL SCHOOL OF INFORMATION SCIENCES
(A Constituent unit of MAHE, Manipal)

Drowsy driver detection using machine learning

Reg. Number	Name	Branch
221039002	VISHAK V KOTEMANE	EMBEDDED SYSTEMS
221039004	SANDESH B S	EMBEDDED SYSTEMS

Under the guidance of

DR MOHAN KUMAR J
Associate Professor
Manipal School of Information Sciences,
MAHE, MANIPAL

11/09/2022

MANIPAL SCHOOL OF INFORMATION SCIENCES
MANIPAL
(A constituent unit of MAHE, Manipal)

Contents

LIST OF FIGURES

1. Introduction

Drowsy driving is the practise of operating a motor vehicle while feeling sleepy, and it can have negative consequences for all drivers. Driving when fatigued dramatically raises the likelihood of collisions, which results in an alarmingly high number of injuries and fatalities each year.

Sleep-deprived drivers remain responsible for about 40% of the road accidents , according to enforcement officers patrolling the highways and major roads in India. Driving while fatigued dramatically raises the likelihood of auto accidents. When a person has a microsleep, which lasts only a few seconds5, the car is more likely to veer off the road or collide with another vehicle. When these collisions take place at great speeds, the harm they do increases.

Even if a person doesn't actually fall asleep while driving, being drowsy is still risky. According to research, sleep deprivation causes mental impairment that is comparable to drunkenness, with 24 hours of insufficient sleep roughly equivalent to 0.10% blood alcohol concentration (BAC). Due to this impairment, a person is less aware of their surroundings and is more susceptible to distractions.

Their reaction time is slowed, which makes it more difficult for them to avoid road hazards. With the help of Open CV, a Raspberry Pi, and image processing, we demonstrate in this work a method to identify driver drowsiness. Image processing is the computer-based manipulation of digital images. This applies certain mathematical operations to photos or videos. A set of parameters, some altered photos, or films are what we get as the outcome of image processing. The Raspberry Pi is utilised in numerous projects for image and video processing because of its inexpensive cost.

The main focus of this study is on Haar cascade classifiers. characteristics akin to Haar, created by Viola and Jones. In a detection window, a Haar-like feature takes into account nearby rectangular sections, sums their pixel intensities, and then determines the difference between these sums. Subdivisions of an image are then categorised using this distinction. The detection of human faces would be an illustration of this. Typically, the cheeks are lighter than the area around the eyes. Consequently, a pair of adjacent rectangular sections above the eye and cheek regions can serve as one example of a Haar-like feature for face detection. The cascade classifier has a list of stages, each of which has a list of weak students. By dragging a window across the image, the system finds the problematic objects. Each stage of the classifier assigns a positive or negative label to the particular area that is determined by the current position of the window. Positive or negative results indicate the presence or absence of the specified object in the image, respectively. If the labelling produces a negative result, the classification of this particular area is finished, and the window's position is changed to the next one. The region advances to the next level of categorization if the labelling yields a favourable result. When all stages, including the final one, produce a result indicating that the object, the classifier issues a final verdict of positive. A true positive indicates that the classifier has correctly identified the object as present in the image.

A false positive occurs when the object is incorrectly identified during the labelling process as being present in the image. A true negative indicates that a non-object was accurately identified as not being the object in question, as opposed to a false negative, which occurs when the classifier is unable to identify the actual object from the image. Each stage of the cascade needs to have a low false negative rate in order for it to function properly because if the real object is mistakenly classed as a non-object, the classification of that branch will stop and the error cannot be undone. However, even if the nth step incorrectly identifies the non-object as the object, this error can be corrected in the n+1st and following stages of the classifier, so each level can still have a high false positive rate.

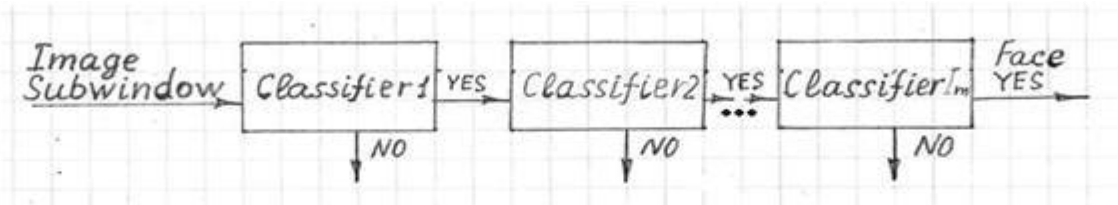

Fig 1.1: Stages of the cascade classifier

Therefore, XML (Extensible Markup Language) files that recognise eyes and faces are produced utilising the haar cascade principle. We can track eye movement with OpenCV-Python using these XML files.

2. Literature Survey

Paper 1

Puja Seemar, Anurag Chandna proposed DROWSY DRIVER DETECTION USING IMAGE PROCESSING. INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY on July, 2017.

The major driver errors are caused by drowsiness, drunken and reckless behavior of the driver. This paper focuses on a driver drowsiness detection system in Intelligent Transportation System, which focuses on abnormal behavior exhibited by the driver using Raspberry pi single board computer. The capability of driving support systems to detect the level of driver's alertness is very important in ensuring road safety. By observation of blink pattern and eye movements, driver fatigue can be detected early enough to prevent collisions caused by drowsiness. In the proposed system a nonintrusive driver drowsiness monitoring system has been developed using computer vision techniques. Based on the simulation results, it was found that the system has been able to detect drowsiness in spite of driver wearing spectacles as well as the darkness level inside the vehicle. Moreover the system is capable of detecting drowsiness within time duration of about two seconds. The detected abnormal behavior is corrected through alarms in real time. They have also done a transformation of an image from 2d to 3d using wavelet analysis. Here they also compared the wavelet technique with other techniques namely stereo-photogrammetric & edge information technique. The image conversion from 2d to 3d can be done by finding the edges of an image. This edge detection can be used in various fields of image processing, image analysis, image pattern recognition, and computer vision, as well as in human vision. In this, we did our experiment using wavelet technique and the results when compared with the stereo photogrammetry and edge information techniques they found that the wavelet technique gives better result.

Paper 2

Dwipjoy Sarkar, Atanu Chowdhury proposed A Real Time Embedded System Application for Driver Drowsiness and Alcoholic Intoxication Detection .Volume 10 Number 9 - Apr 2014.

This paper outlines a novel approach for the real time detection of car driver drowsiness and alcoholic intoxication. There are large numbers of road accidents which takes place due to fatigue or alcohol drinking of driver. Computer vision and alcohol gas sensor application is combined to an embedded system to achieve this goal. The proposed system is realized with an open source 5 megapixel digital camera supported embedded system board Raspberry-pi loaded with Raspbian-OS, and Python-IDLE with Open-CV installed. The Raspberry-pi system board is serially interfaced with another open source embedded system board Arduino Uno with I2C protocol, which will perform some task like issuing the alarm notification and switching off the car power source to stop the car upon receiving the positive detection message from Raspberry-pi.

Paper 3

Tejasweeni Musale,B. H. Pansambal proposed Real Time Driver Drowsiness Detection system using Image processing.International Journal for Research in Engineering Application & Management Vol-02, Issue 08, Nov 2016.

One of the major cause behind the road accidents is driver's drowsiness. Thus, countermeasure device is currently essential in many fields for sleepiness related accident prevention. Real-time driver drowsiness system alerts users when they are falling asleep. The project is designed to combat narcolepsy and microsleep. Microsleep strikes quickly. Users probably don't even realize that they are in the process of falling asleep, and almost certainly don't notice that eye blinking for longer than usual. The implemented project is mainly based on three components 1) Face and Eye detection: Performs scale invariant detection using Haar Cascade Classifier perform through a webcam. 2) Eye feature extraction: Eye features are extracted using Hough Circle and 3) Extract single eye and perform drowsiness detection on it. Whereas the complete system is implemented on Raspberry Pi which uses a webcam to monitor user's eye blink rate and average blink duration to detect drowsiness. The project is designed for a car safety which helps prevent accidents caused by the driver getting drowsy.

3. Objective(s)

1. To learn the drowsy driver image concepts and OpenCV.

2. To implement an algorithm for drowsy driver detection using OpenCV.

3. To implement a machine learning model to identify the drowsy driver.

4. Flowchart Proposed modules

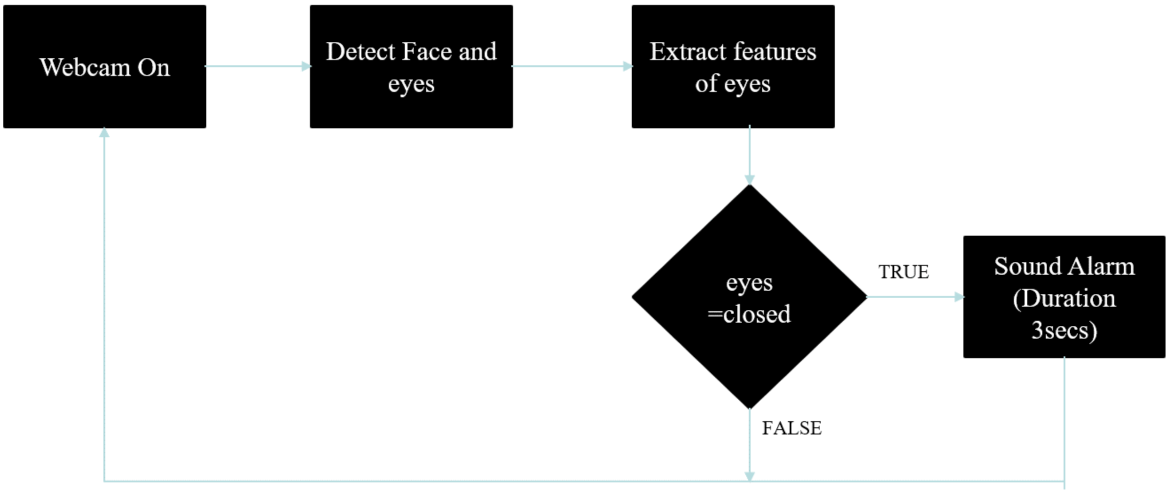

Fig 1.2 Block diagram for webcam detection

1. The webcam will be kept in a while loop, by keeping it turned on in all the conditions except when character ‘Q’ is input.
2. The next step is for the webcam to detect the face and eyes of the driver.
3. Once the face detection is completed, we will proceed ahead with the feature extraction of the eyes.
4. This feature extraction is done for determining whether the eyes are closed or opened.
5. If the eyes are closed, an alarm will be rung for a time duration of 3 secs to alert the driver and an alert ‘Warning eyes are closed’ is displayed.
6. There will be no changes in the alarm status if the eyes are open.

5. Proposed Work/Discussion

VIOLA-JONES ALGORITHM:

Detection happens inside a detection window. Then the detection window is moved across the image as follows:

1. Set the minimum window size and sliding step corresponding to that size.
2. For the chosen window size, slide the window vertically and horizontally with the same step. At each step, a set of *N* faces recognition filters is applied. If one filter gives a positive answer, the face is detected in the current widow.
3. If the size of the window is the maximum size stop the procedure. Otherwise, increase the size of the window and corresponding sliding step to the next chosen size and go to step 2.

From the collection of N filters, each face recognition filter has a set of cascade-connected classifiers. Each classifier evaluates whether a rectangular subset of the detection window resembles a face. If so, the subsequent classifier is used. If every classifier returns a yes, the face is recognised by the filter, which returns a yes. If not, the subsequent filter in the sequence of N filters is applied.
White parts are given a negative weight, whereas grey areas are given a positive weight. Scaling of the Haar feature extractors takes into account the size of the detection window.

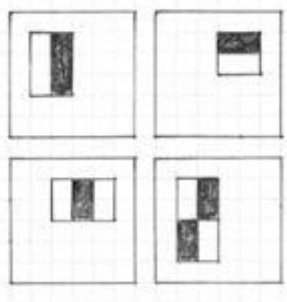

Fig 1.3 Example rectangle features shown relative to the enclosing detection window

The cascade architecture is very efficient because the classifiers with the fewest features are placed at the beginning of the cascade, minimizing the total required computation. The most popular algorithm for features training is AdaBoost.

Integral image

- A **summed-area table** is a data structure and algorithm for quickly and efficiently generating the sum of values in a rectangular subset of a grid.

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

Fig 1.4: An image and its integral image.

SELECTING HAAR-LIKE FEATURES:

- A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region, and calculates the difference between these sums. This difference is then used to categorize subsections of an image.

  Sum = I(C) + I(A)-I(B)-I(D)

  , where points A, B, C, and D belong to the integral image I, as shown in the figure.

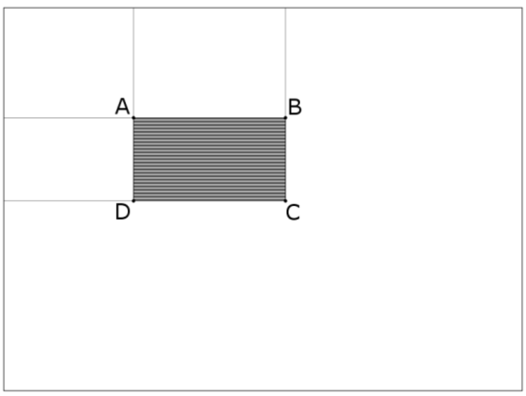

Fig 1.5: Finding the sum of the shaded region

AdaBoost Classifier:

- AdaBoost is used during the training phase of the Viola-Jones object detection algorithm to choose a subset of features and build the classifier.

- A vast collection of photos is created, whose dimensions match those of the detection window.
- Both positive and negative examples for the desired filter (such as just front views of faces) must be included in this set (nonfaces).

Cascading decision:

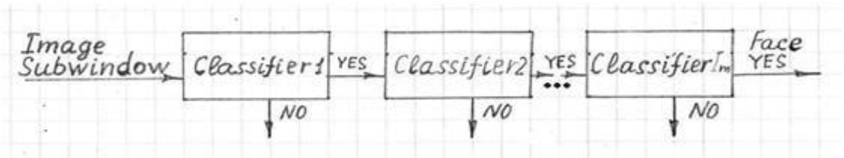

Fig 1.6: Viola Jones filters

- The classifier decision is defined as:

$$C_m = \begin{cases} 1, & \sum_{i=0}^{I_m-1} F_{m,i} > \theta_m \\ 0, & \text{otherwise} \end{cases}$$

$$F_{m,i} = \begin{cases} \alpha_{m,i}, & \text{if } f_{m,i} > t_{m,i} \\ \beta_{m,i}, & \text{otherwise} \end{cases}$$

$f_{m,i}$ is the weighted sum of the 2-D integrals. is the decision threshold for the i-th feature extractor. $\alpha_{m,i}$ and $\beta_{m,i}$ are constant values associated with the i-th feature extractor. θ_m is the decision threshold for the m-th classifier.

6. Conclusions

- The driver's eyes are located, monitored, and used to quickly detect tiredness as part of the real-time drowsiness detection system.
- This method enables the machine to recognize closed eyes (sleep condition).
- After determining the sleep condition, the real-time system sounds an alert.

7. Scope for further work

1. We are currently developing a hardware solution for the above implementation wherein we will be implementing a camera and raspberry pi board for real time drowsiness detection.
2. A machine learning model for detection of the facial features for drowsiness can also be implemented.
3. A future scope of transfer of captured real time snapshot of the driver to a database.

8. References

1. Danisman, T., Bilasco, I.M., Djeraba, C. and Ihaddadene, N., 2010, October. Drowsy driver detection system using eye blink patterns. In *2010 International Conference on Machine and Web Intelligence* (pp. 230-233). IEEE.
2. Sarkar, D. and Chowdhury, A., 2014. A real time embedded system application for driver drowsiness and alcoholic intoxication detection. *International Journal of Engineering Trends and Technology*, *10*(9), pp.461-465.
3. Musale, T. and Pansambal, B., 2016. Real Time Driver Drowsiness Detection system using Image processing. *Research in Engineering Application & Management (IJREAM)*, *2*(08).