Driver Drowsiness Detector

A REPORT ON PROJECT BASED LEARNING
(SEMESTER -IV)

Submitted by

Yash Patwardhan (23361) *Amaan Naikwadi (23363)*

Vedant Deshpande (23362) *Aditya Kangune (23365)*

SECOND YEAR ENGINEERING

Society for Computer Technology and Research

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DHANKAWADI, PUNE – 43

A.Y. 2020-21

ABSTRACT

Driver in-alertness is an important cause for most accident related to the vehicles crashes. Driver fatigue resulting from sleep deprivation or sleep disorders is an important factor in the increasing number of the accidents on today's roads. Drowsy driver warning system can form the basis of the system to possibly reduce the accidents related to driver's drowsiness. The purpose of such a system is to perform detection of driver fatigue.

By placing the camera inside the car, we can monitor the face of the driver and look for the eye-movements which indicate that the driver is no longer in condition to drive. In such a case, a warning signal should be issued. We also describe a method that can determine if the eyes are open or closed. The main criterion of this system is that it must be highly non-intrusive and it should start when the ignition is turned on without having at the driver initiate the system. Nor should the driver be responsible for providing any feedback to the system. The system must also operate regardless of the texture and the color of the face. It must also be able to handle diverse condition such as changes in light, shadows, reflections etc.

Therefore, in this project, a light-weight, real time driver's drowsiness detection system is developed and implemented. The system records the videos and detects driver's face in every frame by employing image processing techniques. The system is capable of detecting facial landmarks, computes Eye Aspect Ratio (EAR) to detect driver's drowsiness based on adaptive thresholding. Machine learning algorithms have been employed to test the efficacy of the proposed approach. Thus, a drowsy driver warning system using image processing is proposed.

Key words – Driver Drowsiness, system, fatigue, image processing.

INTRODUCTION

Our team aims at creating a program using OpenCV to detect driver drowsiness. The objective of this intermediate Python project is to build a drowsiness detection system that will detect that a person's eyes are closed for a few seconds. This system will alert the user when drowsiness is detected.

Real Time Drowsiness behaviors which are related to fatigue are in the form of eye closing, head nodding or the brain activity. Hence, we can either measure change in physiological signals, such as brain waves, heart rate and eye blinking to monitor drowsiness or consider physical changes such as sagging posture, leaning of user's head and open/closed state of eyes.

The former technique, while more accurate, is not realistic since highly sensitive electrodes would have to be attached directly on the user' body and hence which can be annoying and distracting to the user. In addition, long time working would result in perspiration on the sensors, diminishing their ability to monitor accurately. The second technique is to measure physical changes (i.e. open/closed eyes to detect fatigue) is well suited for real world conditions since it is non-intrusive by using a video camera to detect changes. In addition, microsleeps that are short period of sleeps lasting 2 to 3 minutes are good indicators of fatigue. Thus, by continuously monitoring the eyes of the user one can detect the sleepy state of the driver.

LITERATURE SURVEY

1. In [1], it portrays a productive strategy for drowsiness recognition by three characterized stages. These three stages are facial highlights discovery utilizing Viola Jones. When the face is distinguished, the framework is made light invariant by fragmenting the skin part alone by considering just the chromatic segments. The following of eyes and yawning recognition are finished by relationship coefficient layout coordinating.
2. Once the detection algorithm has successfully detected a face and, subsequently, the eyes, it focuses on determining in what state the eyes are (closed or open). The proposed system monitors if the driver’s eyes are being closed for prolonged period of time. If that is the case, it concludes that the driver might be experiencing signs of drowsiness.
3. The classification method implemented in the system uses data from the setup stage as training data that is applied to a 2-class Support Vector Machine (SVM).
4. In [2], the use of the image processing is done to determine driver states. From the image of the face it is detected if the driver is awake or asleep. The technique of the behavior of the eyes, calculates the blinking frequency and the time interval of eyes closing in order to determine the rate of drowsiness
5. PERCLOS (Percent of the time Eyelids are Closed) index is used which measures the percentage of time a person's eyes are closed at 80% to 100% in a period.
6. The Yawning technique is based on the driver's yawn frequency. The mouth is compared with a reference point experimentally obtained by the programmer and the number of times the driver has yawned is calculated to generate a drowsiness index.

7. In [3], The first point of venture is to utilize the retinal reflection as it implies to finding the eyes on the confront, and after that utilizing the nonappearance of this reflection as a way of identifying when the eyes are closed.
8. Applying this calculation on sequential video outlines help within the calculation of eye closure period. Eye closure period for lazy drivers are longer than ordinary blinking.

SYSTEM ARCHITECTURE

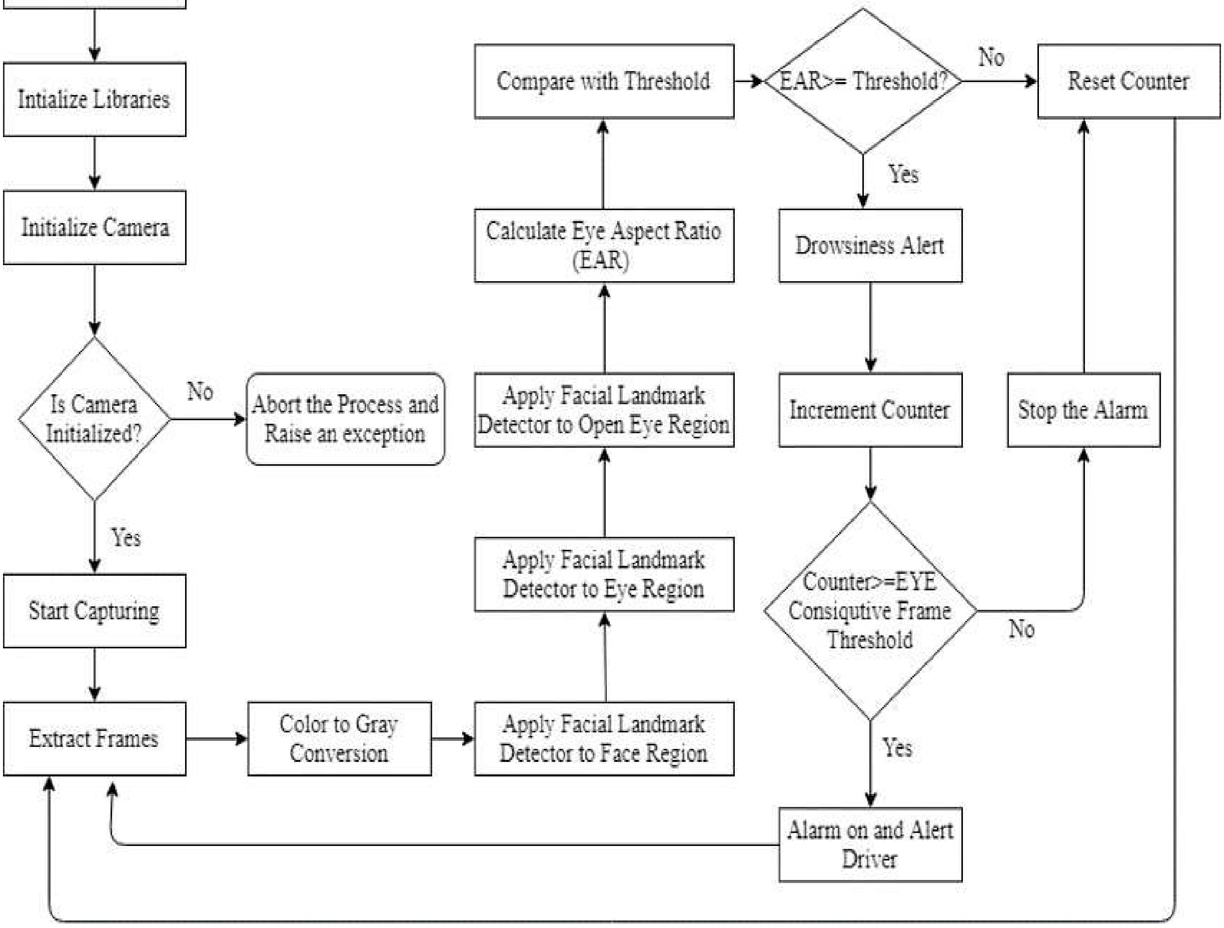

Fig. Workflow of system

The various detection stages are discussed as:

Face Detection: For the face Detection it uses Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel.

Eye detection: In the system we have used facial landmark prediction for eye detection Facial landmarks are used to localize and represent salient regions of the face, such as:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more. In the context of facial landmarks, our goal is detecting important facial structures on the face using shape prediction methods. Detecting facial landmarks is therefore a two step process:

• Localize the face in the image.
• Detect the key facial structures on the face ROI

Localize the face in the image: The face image is localized by Haar feature-based cascade classifiers which was discussed in the first step of our algorithm i.e. face detection.

Detect the key facial structures on the face ROI: There are a variety of facial landmark detectors, but all methods essentially try to localize and label the following facial regions:

• Mouth
• Right eyebrow
• Left eyebrow
• Right eye
• Left eye
• Nose

Drowsiness Detection:

The last step of the algorithm is to determine the person’s condition based on a pre-set condition for drowsiness. The average blink duration of a person is 100-400 milliseconds (i.e. 0.1-0.4 of a second). Hence if a person is drowsy his eye closure must be beyond this interval. We set a time frame of 5 seconds. If the eyes remain closed for five or more seconds, drowsiness is detected and alert pop regarding this is triggered.

EXPERIMENTATION RESULTS

Implementation of drowsiness detection with Python and OpenCV was done which includes the following steps: Successful runtime capturing of video with camera. Captured video was divided into frames and each frame was analyzed. Successful detection of face followed by detection of eye. If closure of eye for successive frames were detected, then it is classified as drowsy condition else it is regarded as normal blink and the loop of capturing image and analyzing the state of driver is carried out again and again. In this implementation during the drowsy state the eye is not surrounded by a circle or it is not detected, and a corresponding message is shown.

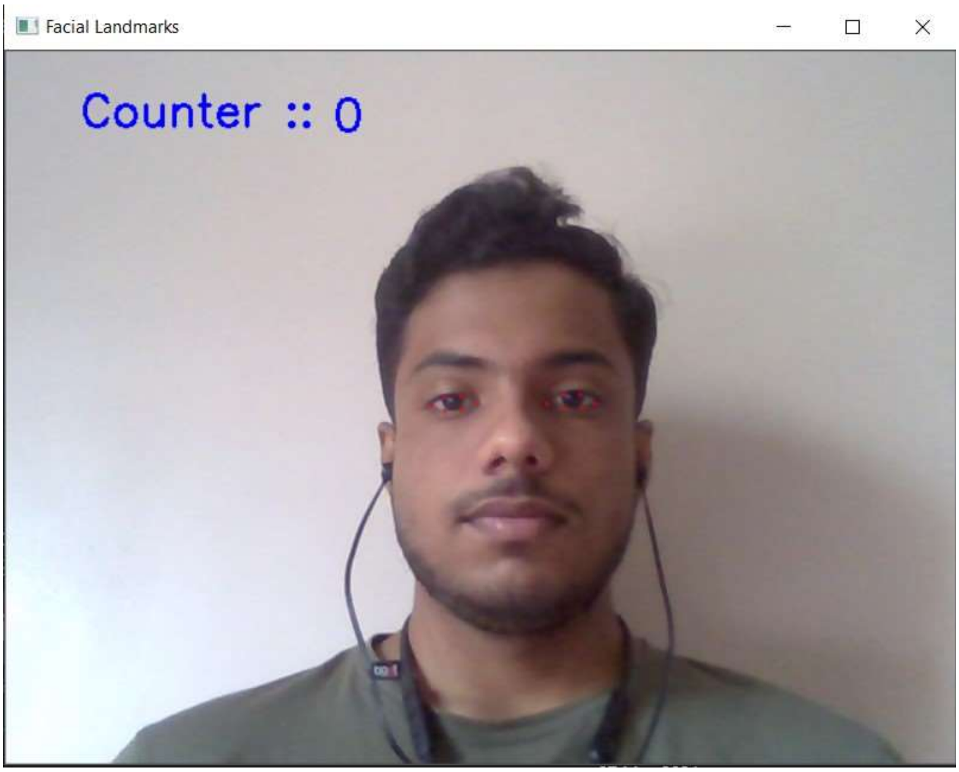
Facial Landmarks
Counter :: 0

Fig 1


Facial Landmarks
Counter :: 36
Driver Drowsy !

FIG 2

CHAPTER 5

CONCLUSION

A real-time eye blink detection algorithm was presented. We quantitatively demonstrated that Haar feature-based cascade classifiers and regression-based facial landmark detectors are precise enough to reliably estimate the positive images of face and a level of eye openness. While they are robust to low image quality (low image resolution to a large extent) and in-the wild.

Limitations:

1. Use of spectacles: In case the user uses spectacle then it is difficult to detect the state of the eye. As it hugely depends on light hence reflection of spectacles may give the output for a closed eye as opened.
2. Multiple face problem: If multiple face arises in the window then the camera may detect more number of faces undesired output may appear. Because of different condition of different faces. So, we need to make sure that only the driver face come within the range of the camera. Also, the speed of detection reduces because of operation on multiple faces.
3. The module used requires high-definition images to detect faces.
4. Accuracy of detecting faces can be improved via Machine Learning (YOLO object detection)

REFERENCES

1. *Machine Learning for OpenCV 4: Intelligent Algorithms for Building Image Processing Apps Using OpenCV 4, Python, and Scikit-learn, 2nd Edition*

   Book by Aditya Sharma, Michael Beyeler, and Vishwesh Ravi Shrimali
2. https://opencv.org/
3. https://www.tutorialspoint.com/opencv/index.htm
4. https://www.geeksforgeeks.org/opencv-python-tutorial/
5. https://www.geeksforgeeks.org/opencv-python-program-face-detection/
6. *[1], Aleksandar Coli´c, Oge Marques and Borko Furht, Design and Implementation of a Driver Drowsiness Detection System A Practical Approach*
7. *[2], Eddie E. Galarza1,1 , Fabricio D. Egas1 , Franklin M. Silva1 , Paola M. Velasco1 , Eddie D. Galarza1 , 1 Universidad de las Fuerzas Armadas – ESPE, Real Time Driver Drowsiness Detection Based on Driver's Face Image Behavior Using a System of Human Computer Interaction Implemented in a Smartphone.*
8. *[3], Aayushi Ahlawat, Simran Kaur, Sejal Rana. Drowsiness Detection System Using OpenCV, International Journal of Creative and Research Thoughts.*