

REAL TIME DETECTING DRIVER'S DROWSINESS USING COMPUTER VISION

Kusuma Kumari B.M

Assistant Professor, Department of Computer Science, University College of Science, Tumkur University, Tumkur, Karnataka, India

Abstract

Driver inattention is one of the main causes of traffic accidents. Monitoring a driver to detect inattention is a complex problem that involves physiological and behavioral elements. Different approaches have been made, and among them Computer Vision has the potential of monitoring the person behind the wheel without interfering with his driving. In this paper I have developed a system that can monitor the alertness of drivers in order to prevent people from falling asleep at the wheel. The other main aim of this algorithm is to have efficient performance on low quality webcam and without the use of infrared light which is harmful for the human eye. Motor vehicle accidents cause injury and death, and this system will help to decrease the amount of crashes due to fatigued drivers. The proposed algorithm will work in three main stages. In first stage the face of the driver is detected and tracked. In the second stage the facial features are extracted for further processing. In last stage the most crucial parameter is monitored which is eye's status. In the last stage it is determined that whether the eyes are closed or open. On the basis of this result the warning is issued to the driver to take a break.

Keywords: Face Detection, Facial Feature Extraction, Eye Detection, Eye Status

1. INTRODUCTION

We know that there is a huge increase of public-private road transportation day by day as we move more steps in the modernize world. As we all know how tedious and bored driving is, when it is for long time distance and time. One of the main causes behind the accident is driver's unalertness due to long route continues-tedious driving without sleep and rest. Tired Driver losses his alertness and also due to physical cycle of human body can get nap while driving. Even Fraction of second's nap can be turning into dangerous and lives threaten accident may leads towards death also. To prevent this type of threat it is advisable to monitoring driver's alertness continuously and when we detect condition like pre-drowsy stage or drowsiness driver should be given feedback to be alert. That can act as warning for driver and driver should take enough rest for a time & we can prevent the life threat. Generally driver drowsiness is main factor in 25percent accidents and 60 percent in accidents that they redound to death [9].

National Highway Traffic Safety Administration (NHTSA) analysis data indicates that drowsiness while driving is a contributing factor for road accidents and it results in 4-6 times higher crash risk relative to alert drivers [1]. Most of the fatal road accidents occur at speeds greater than 70 Kmph. The World Health Organization (WHO) has reported that India has the worst road conditions in the world resulting approximately two and a half lakh deaths in 2010 and 2011 [2].

Research shows that driver fatigue and drowsiness is one of the major reasons for the increasing accidents [3]. Driver fatigue not only impacts the alertness and response time of the driver but it also enhances the chances of being involved in car accidents. The sleepy drivers fail to take right actions prior to a collision. An important irony in driver's fatigue is that the driver may be too drained to comprehend his own level of drowsiness. This significant problem is often ignored by the driver. Consequently, the use of supporting systems that examine a driver's level of vigilance is necessary to avoid road accidents. These systems should then alert the driver in the case of sleepiness or inattention. Some warning signs that can be measured as indications of driver fatigue are: daydreaming while on the road, driving over the centre line, yawning, feeling impatient, feeling stiff, heavy eyes and reacting slowly.

2. LITERATURE REVIEW

There are many technologies for drowsiness detection and can be divided into three main categories: biological indicators, vehicle behavior, and face analysis. The first type measures biological indicators such as brain waves, heart rate and pulse rate. These techniques have the best detection accuracy but they require physical contact with the driver. They are intrusive. Thus, they are not practical.

The second type measures vehicle behaviors such as speed, lateral position and turning angle [8]. These techniques may be implemented non-intrusively, but they have several

limitations such as the vehicle type, driver experience and driving conditions. Furthermore, it requires special equipment and can be expensive.

The third type is face analysis. Since the human face is dynamic and has a high degree of variability, face detection is considered to be a difficult problem in computer vision research. As one of the salient features of the human face, human eyes play an important role in face recognition and facial expression analysis. In fact, the eyes can be considered salient and relatively stable feature on the face in comparison with other facial features. Therefore, when we detect facial features, it is advantageous to detect eyes before the detection of other facial features. The position of other facial features can be estimated using the eye position. In addition, the size, the location and the image-plane rotation of face in the image can be normalized by only the position of both eyes.

3. PROPOSED SYSTEM

The algorithm proposed is a computer vision algorithm that aids in the detection of the current driver state of vigilance. It detects the current state of the driver eyes in every frame (open or closed). Applying this algorithm on consecutive video frames may aid in the calculation of eye closure period. Eye closure periods for drowsy drivers are longer than normal blinking, a fact that can be exploited to monitor a driver state of vigilance. It is also a very critical parameter because the closure of eyes for a little longer time could result in serve crash. So we will warn the driver as soon as closed eye is detected.

Algorithm Stages

The major stages of algorithm are as:

3.1 Image Capture

The image is captured from the video, where image is a numeric representation (normally binary) of a two dimensional image. The video is acquired by using a low cost web camera. The camera provides 30 frames per second at VGA mode. Then the recorded video is opened in MATLAB and the frames are grabbed, then the algorithm is run on every frame detecting the driver's vigilance.

3.2 Face Detection

In every given frame the face is detected using an specified algorithm. Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies. Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Early face-detection algorithms focused on the detection of frontal human faces, whereas newer algorithms attempt to solve the more

general and difficult problem of multi-view face detection. That is, the detection of faces that are either rotated along the axis from the face to the observer (in-plane rotation), or rotated along the vertical or left-right axis (out-of-plane rotation), or both.

The flow chart of the algorithm is represented in Fig 1.

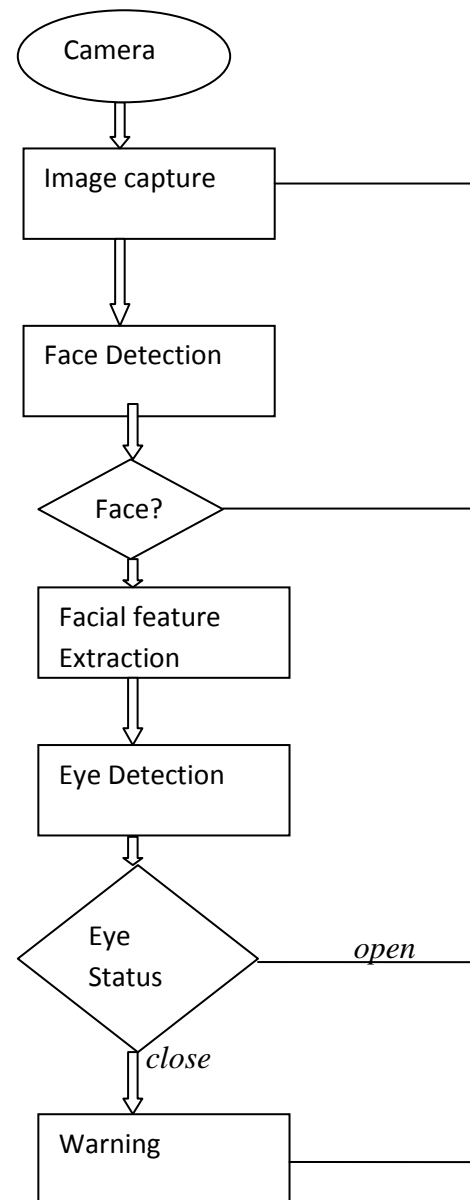


Fig -1: Diagram of the Proposed System

The newer algorithms take into account variations in the image or video by factors such as face appearance, lighting, and pose. Many algorithms implement the face-detection task as a binary pattern-classification task. That is, the content of a given part of an image is transformed into features, after which a classifier trained on example faces decides whether that particular region of the image is a face, or not. If the result

of face detection comes positive then the algorithm proceeds to the next, otherwise the flow of algorithm goes back to the image capture stage. The face detection is done by the Viola and Jones algorithm [6],[5]. For the accurate setup we used the cascade which is part of the OpenCV library. The results of face detector are as: Fig.2.



Fig-2: Input and output of face detector.

3.3 Facial Feature Detection

Facial Feature Detection (FFD) is to find the exact location of facial features, such as mouth and eyes corners, lip contour, jaw contour, and the shape of the entire face.

Face and facial feature detection are difficult problems, due to the large variations a face can have in a scene caused by factors such as intra-subject variations in pose, scale, expression, color, illumination, background clutter, presence of accessories, occlusions, hair, hats, eyeglasses, beard etc. The easiest case considers a single frontal face and divides it into region of interest like for mouth, eye and nose etc. The feature detection is used to determine the region of eyes so that their status can be determined easily and quickly. It is performed by segmenting the face that has been detected.

For the facial feature extraction firstly the search areas are defined according to the geometry of the face. Then in these search areas specific content is found by its own algorithm.

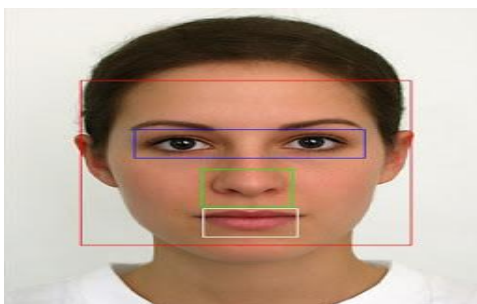


Fig: 3 the highlighted search areas for mouth, nose and eyes.

3.3.1 Nose Holes

Finding nose holes in an area given from face's geometry depends on the angle between camera and face. If there isn't a direct line of sight between nose holes and camera, it is

obviously not possible to track them. Nose holes colors have a significant saturation, depending on its color black. The threshold must be defined and over geometry or clustering two centers of saturation can be found.

3.3.2 Mouth

Detecting the middle of the mouth isn't as simple as it is thought. There are a lot of possibilities, going over gradient horizontal and/or vertical decent, hue or saturation. At the moment it is implemented utilizing the distinct hue of lips. Light reflects on lips and this point is fetched by a defined hue value. In contrast to the methods, this method is not light independent, thus intensity and direction of the light can influence results. A better method should be included in the future.

3.3.3 Eyes and Pupils

A lot of ways can be developed to find pupils in the given area surrounding the eyes. It can also be done using hue or saturation, which leads – controlled conditions given – to good results, but it highly depends on the current light situation. Different pupils were used for testing and the best results were gained by pupils directly from the tester, which was not really surprising. Obtaining them is not that simple though. An algorithm from Anirudh S.K. called “Integrodifferential operator” [4] was used, which requires too much calculation time to be used in real-time environments, but is fast enough for getting the first pupils, so it takes time only first time for finding the Eigen pupil. Once the Eigen pupil is there the rest is very simple. But sometimes due the lack of lighting conditions it takes a little longer time to find the pupil. Still this algorithm is very accurate and then it perform incredibly well under very low lighting condition. This feature of operating under low lighting conditions is very useful in our system. Then for tracking the Eigen pupil found by the Integro differential operator, there are few steps:

- The face is detected from the image.
- In the detected face the geometric region of the eye is defined.
- The Eigen pupil is resized according to the size of the newly found eye region.
- Then Eigen pupil is run as mask on the whole eye region and the mean difference is found at every point.
- The point which gives the minimum mean difference is defined as the pupil of the eye.

This method of finding the pupil is very quick and easy. It takes very less time as compared to the Integro differential operator.

3.4 Eye Detection

Eye detection is the essence of eye tracking and iris recognition – without it, there's no way to identify the eye itself. It sounds simple, but it's really quite complicated. In

this stage the eyes are detected in the specified region by the feature detection. In the beginning it looks for the Eigen eye. This process is time taking and it is done just once. After the detection of Eigen eye it is just matched in the other frames for the same candidate.

3.5 Detection of Eyes Status

To detect the eyes are whether open or close was a quite challenging task. Many different approaches were implemented for the detection of the status of the eyes. Some of them are listed below:

3.5.1 Eye Status Detection using Edge Detector

Status Determination using edge detection In this novel and simple approach, the eye region from the detected face is subjected to the edge detector. The advantage of using MATLAB is that it has built in edge detector. so I applied “Canny Edge detector” [7] and “Sobel Edge detector”. The results of both the edge detectors are presented below. The Canny Edge detector has a lot of noise and it makes some extra edge. On the other hand the Sobel Edge detector gives better results in case of the noisy image. Both the edge detectors provide the binary output.

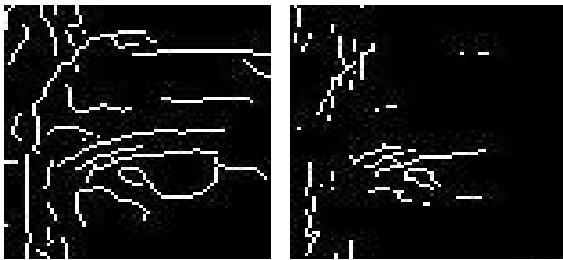


Fig:- 4 Eye edge diagram using a) canny edge detector and b) sobel edge detector.

The eye status detection is performed by calculating the sum of the binary image. As the sum of the image will be higher in case of open eyes, because there are more edges than the closed eye. The problem arises when the lighting conditions and the background interfere with the image. It results in the change in number of edges.

3.5.2 Eye Status Detection using Correlation

The in correlation approach the eye region is correlated with the previous eye region. The result will be different in the case of change of status of eye. It was implemented with the built-in function of 2D correlation in MATLAB. But the positioning of eye in each frame and the external factors affect the correlation results. The experimental results show that this system is also not very good for the implementation.

3.5.3 Proposed Method for Eye Status Detection

In this technique, the first step is to calculate the average intensity for each x – coordinate. These average values are found for both the eyes separately. When the plot of these average values was observed it was found that there are two significant intensity changes. The first intensity change is the eyebrow, and the next change is the upper edge of the eye, as shown in the figure. Thus with the knowledge of the two valleys the position of the eyes in the face was found. The state of the eyes (whether it is open or closed) is determined by distance between the first two intensity changes (valleys) found in the above step. When the eyes are closed, the distance between the x – coordinates of the intensity changes is larger if compared to when the eyes are open as shown in Fig 5.

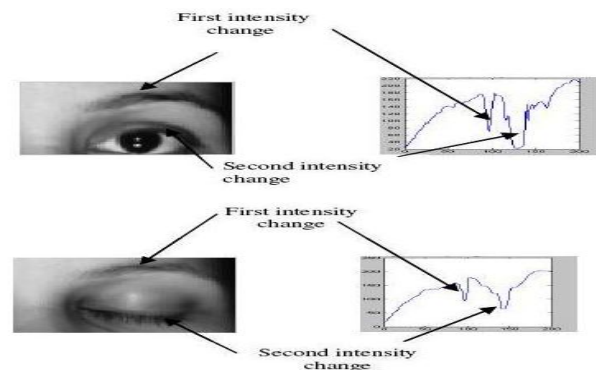


Fig:-5 Average intensity variations on the face when eyes are open and close.

4. DROWSINESS DETECTION

The eye status defines the drowsiness state of the driver. If the eyes are closed then the driver is fatigued and need to take some rest. Some of the systems don't warn the driver on the first closed eye is detected, but for a vehicle moving at a high speed the closed eye even for a moment can be very crucial. Because if the vehicle is moving on 100 Miles/Hour, which means its 44.7 Meters/Second. So if we waste a second or so that could prove to be lethal



Fig:-6 Tracking for eyes open.



Fig:-7 Tracking for eyes closed.

For the Fig 6 and 7 we can see that in case of eyes closed the pointer does not exactly points to the eyes. This is because the use of matching algorithm for the eyes tracking. It will match it to the eye brows, because they also high concentration of dark pixels. For drowsiness detection the results of the eyes status algorithm is very accurate.

5. CONCLUSIONS

Drowsy Driving is a growing problem in all over the world, and the risk, danger and often tragic outcomes of drowsy driving are sobering. We make final conclusion for our idea to catch driver's sudden napped stage or unalerness based on measurement of time duration between each blink signal is successful and also low cost in nature and also very easy to design. It can works as a boon to driver though this system.

REFERENCES

- [1]. S.G. Klauer , T. A. Dingus, Neale, V. L., Sudweeks, J.D., and Ramsey, DJ,"The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data," Virginia Tech Transportation Institute, Technical Report # DOT HS 810594
- [2]. <http://www.financialexpress.com/news/india-suffers-from-the-highest-number-of-road-accidents-who/786419/>
- [3]. The Royal Society for Prevention of Accidents, "Driver Fatigue and Road Accidents, A Literature Review and Position Paper, February 2001"
- [4]. A. G. Daimler, The Electronic Drawbar, June 2001. [Online]. Available: <http://www.daimler.com>.
- [5]. Rapid Object Detection using a Boosted Cascade of Simple Features ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001 by Paul Viola and Michael Jones
- [6]. International journal of computer vision 57(2), 2004 Robust Real Time Face Detection by Paul Viola and Michael Jones
- [7]. Wen-Bing Horng, Chih-Yuan Chen, Yi Chang, et al, "Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching " Proc. of the 2004 IEEE

International Conference on Networking, Sensing & Control, pp. 7-12, 2004.

[8]. Xiao Fan a,b,*, Yanfeng Sun a, Baocai Yin a, Xiuming Guo "Gaborbased dynamic representation for human fatigue monitoring in facial image sequences" Pattern Recognition Letters 31 (2010) 234–243

[9]. Bergasa L. M., Nuevo J. u., Sotelo M. A., Barea R., and Lopez "Visual Monitoring of Driver Inattention", Studies in Computational Intelligence (SCI). (2008).