

A Low Cost Drivers Drowsiness Detection System For Autonomous Mobile Vehicle

S.Palani and S.Kothandaraman#*

*Assistant Professor, Department of Computer Science

Kanchi Shri Krishna College of Arts and Science, Kanchipuram-631551

Assistant Professor and Head, Department of Information Technology,

FSH, SRM University, SRM Nagar, Kattankulathur - 603 203

Abstract:

In the present study, a vehicle driver drowsiness warning system using image processing technique with monitoring the eye logic inference is developed and investigated. The principle of the proposed system is based on facial images analysis for warning the drowsy driver or inattention to prevent traffic accidents. The system uses a small monochrome security camera that points directly towards the driver's face and monitors the driver's eyes in order to detect fatigue. In such a case when fatigue is detected, a warning signal is issued to alert the driver. This report describes how to find the eyes, and also how to determine if the eyes are open or closed. (Monitoring the eye logic algorithm is proposed to determine the level of fatigue by determining the state of the eye whether opened or closed accordingly.) The detail of image processing technique and the characteristic also is present in this paper. The experimental results indicated that the proposed expert system is effective for increasing safe in drive.

Keywords: Drowsy, Binarization, noise,detection

1. Introduction:

Driver fatigue is a significant factor in a large number of vehicle accidents. Recent statistic estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes [9].The aim of this paper is to develop a prototype drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver's eyes in real-time. By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident. Detection of fatigue involves a sequence of images of a face, and the observation of eye movements and blink patterns. The analysis of face images is a popular research area with applications such as face recognition, virtual tools, and human identification security systems. This paper is focused on the localization of the eyes, which involves looking at the entire image of the face, and determining the position of the eyes, by a self developed image-processing algorithm. Once the position of the eyes is located, the system is designed to determine whether the eyes are opened or closed, and detect fatigue.

1.1 System Requirements:

The requirements for an effective drowsy driver detection system are as follows:

- A non-intrusive monitoring system that will not distract the driver.
- A real-time monitoring system, to insure accuracy in detecting drowsiness.
- A system that will work in both daytime and nighttimes conditions.

The above requirements are subsequently the aims of this paper. The paper will consist of a concept level system that will meet all the above requirements

1.2 Techniques for Detecting Drowsy Drivers:

Possible techniques for detecting drowsiness in drivers can be generally divided into the following categories:

- Sensing of physiological characteristics
- Sensing of driver operation
- Sensing of vehicle response
- Monitoring the response of driver.

Among these methods, the techniques that are best, based on accuracy are the ones based on human physiological phenomena [9].

1.3 Monitoring Physiological Characteristics:

This technique is implemented in two ways: measuring changes in physiological signals, such as

- Brain Waves
- Heart Rate
- Eye Blinking

1.4 Measuring physical changes such as

- Sagging Posture
- Leaning of the driver's head
- The open/closed states of the eyes [9].

The first technique, while most accurate, is not realistic, since sensing electrodes would have to be attached directly onto the driver's body, and hence be annoying and distracting to the driver. In addition, long time driving would result in perspiration on the sensors, diminishing their ability to monitor accurately.

The second technique is well suited for real world driving conditions since it can be non-intrusive by using optical sensors of video cameras to detect changes.

2.0 Other Methods:

Driver operation and vehicle behavior can be implemented by monitoring the steering wheel movement, accelerator or brake patterns, vehicle speed, lateral acceleration, and lateral displacement. These too are non-intrusive ways of detecting drowsiness, but are limited to vehicle type and driver conditions. The final technique for detecting drowsiness is by monitoring the response of the driver. This involves periodically requesting the driver to send a response to the system to indicate alertness. The problem with this technique is that it will eventually become tiresome and annoying to the driver.

Design:

This section aims to present my design of the Drowsy Driver Detection System.

2. 1. Concept Design

As seen in the various references [3],[5],[6],[7],[8],[9], there are several different algorithms and methods for eye tracking, and monitoring. Most of them in some way relate to features of the eye (typically reflections from the eye) within a video image of the driver. As the paper progressed, the basis of the horizontal intensity changes idea from paper [7] was used. One similarity among all faces is that eyebrows are significantly different from the skin in intensity, and that the next significant change in intensity, in the y-direction, is the eyes.

This facial characteristic is the centre of finding the eyes on the face, which will allow the system to monitor the eyes and detect long periods of eye closure.

2. 2. System Configuration

Each of the following sections describes the design of the drowsy driver.

2.2.1 Background and Ambient Light

Because the eye tracking system is based on intensity changes on the face, it is crucial that the background does not contain any object with strong intensity changes. Highly reflective object behind the driver, can be picked up by the camera, and be consequently mistaken as the eyes. Since this design is a prototype, a controlled lighting area was set up for testing.

Low surrounding light (ambient light) is also important, since the only significant light illuminating the face should come from the drowsy driver system. If there is a lot of ambient light, the effect of the light source diminishes. The testing area included a black background and low ambient light (in this case, the ceiling light was physically high and hence had low Illumination). This setup is somewhat realistic since inside a vehicle, there is no direct light, and the background is fairly uniform.

2.2.1 Camera

The drowsy driver detection system consists of a CCD camera that takes images of the driver's face. This type of drowsiness detection system is based on the use of image processing technology that will be able to accommodate

individual driver differences. The camera is placed in front of the driver, approximately 30 cm away from the face. The camera must be positioned such that the following criteria are met:

1. The driver's face takes up the majority of the image.
2. The driver's face is approximately in the centre of the image.

The facial image data is in 480x640 pixel formats and is stored as an array through the predefined Piccolo driver functions (as described in a later section).

2.2.3 Light Source

For conditions when ambient light is poor (night time), a light source must be present to compensate. The construction of an infrared light source using infrared LED, needs 50 LEDs to illuminate the entire face. To cut down cost, a simple desk light was used. Using the desk light alone could not work, since the bright light is blinding if looked at directly, and could not be used to illuminate the face.

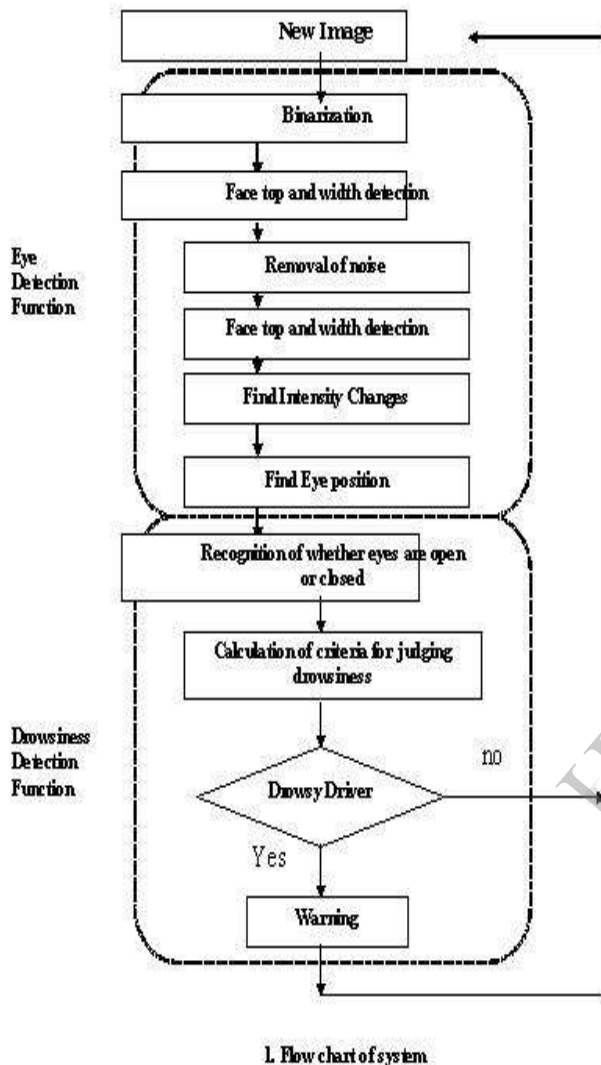
However, light from light bulbs and even daylight all contain infrared light; using this fact, it was decided that if an infrared filter was placed over the desk lamp, this would protect the eyes from a strong and distracting light and



Fig 1. Photograph of Drowsy Driver Detection System prototype

provide strong enough light to illuminate the face. A wideband infrared filter was placed over the desk lamp, and provides an excellent method of illuminating the face.

Working principles:



1. Flow chart of system

3.1 Eye Detection Function:

After inputting a facial image, pre-processing is first performed by binarizing the image.

The top and sides of the face are detected to narrow down the area of where the eyes exist.

Using the sides of the face, the centre of the face is found, which will be used as a reference when comparing the left and right eyes.

Moving down from the top of the face, horizontal averages (average intensity value for each y coordinate) of the face area are calculated. Large changes in the averages are used to define the eye area.

The following explains the eye detection procedure in the order of the processing operations.

3.2 Binarization:

The first step to localize the eyes is binarizing the picture. Binarization is converting the image to a binary image. A binary image is an image in which each pixel assumes the value of only two discrete values. In this case the values are 0 and 1, 0 representing black and 1 representing white. Examples of binarized images are shown in Figure.

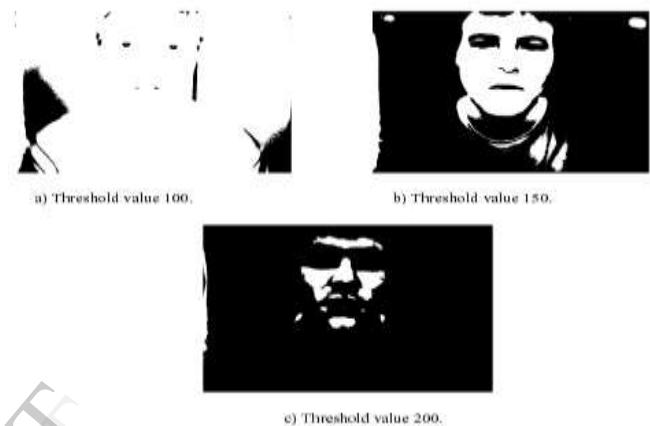


Fig 2. Examples of binarization using different thresholds

With the binary image it is easy to distinguish objects from the background. The grey scale image is converting to a binary image via thresholding. The output binary image has values of 0 (black) for all pixels in the original image with luminance less than level and 1 (white) for all other pixels. Thresholds are often determined based on surrounding lighting conditions, and the complexion of the driver.

After observing many images of different faces under various lighting conditions a threshold value of 150 was found to be effective. The criteria used in choosing the correct threshold was based on the idea that the binary image of the driver's face should be majority white, allowing a few black blobs from the eyes, nose and/or lips. Figure 2 demonstrates the effectiveness of varying threshold values. Figure 2a, 2b, and 2c use the threshold values 100, 150 and 200, respectively. Figure 2b is an example of an optimum binary image for the eye detection algorithm in that the background is uniformly black, and the face is primary white. This will allow finding the edges of the face, as described in the next section.

3.3. Face Top and Width Detection:

The next step in the eye detection function is determining the top and side of the driver's face. This is important since finding the outline of the face narrows down the region in which the eyes are, which makes it easier (computationally) to localize the position of the eyes.

The first step is to find the top of the face such that Assume the person's face is approximately in the centre of the image, the initial starting point used is (100,240). The starting

x-coordinate of 100 was chosen, to insure that the starting point is a black pixel (no on the face). The following algorithm describes how to find the actual starting point on the face, which will be used to find the top of the face.

Starting at (100,240), increment the x-coordinate until a white pixel is found. This is considered the left side of the face.

If the initial white pixel is followed by 25 more white pixels, keep incrementing x until a black pixel is found.

Count the number of black pixels followed by the pixel found in step2, if a series of 25 black pixels are found, this is the right side.

The new starting x-coordinate value (x_1) is the middle point of the left side and right side.



Fig 3

- ★ Starting point (100,240)
- A Count 25 white pixels from this point
- B Count 25 black pixels from this point
- x_1 Middle point of A and B

Fig 3. Demonstrates the algorithm

Using the new starting point ($x_1, 240$), the top of the head can be found. The following is the algorithm to find the top of the head:

Beginning at the starting point, decrement the y-coordinate (i.e.; moving up the face).

Continue to decrement y until a black pixel is found. If y becomes 0 (reached the top of the image), set this to the top of the head.

Check to see if any white pixels follow the black pixel.

If a significant number of white pixels are found, continue to decrement y.

If no white pixels are found, the top of the head is found at the point of the initial black pixel.

Once the top of the driver's head is found, the sides of the face can also be found.

Below are the steps used to find the left and right sides of the face.

1. Increment the y-coordinate of the top (found above) by 10. Label this

$y_1 = y + \text{top}$.

2. Find the centre of the face using the following steps:

At point (x_1, y_1), move left until 25 consecutive black pixels are found, this is the left side (l_x).

At point (x_1, y_1), move right until 25 consecutive white pixels are found, this is the right side (r_x).

The centre of the face (in x-direction) is: $(r_x - l_x)/2$. Label this x_2 .

3. Starting at the point (x_2, y_1), find the top of the face again. This will result in a new y-coordinate, y_2 .

4. Finally, the edges of the face can be found using the point (x_2, y_2).

i. Increment y-coordinate.

ii. Move left by decrementing the x-coordinate, when 5 black consecutive pixels are found, this is the left side, add the x-coordinate to an array labeled 'left_x'.

iii. Move right by incrementing the x-coordinate, when 5 black consecutive pixels are found, this is the right side, add the x-coordinate to an array labeled 'right_x'.

iv. Repeat the above steps 200 times (200 different y-coordinates).

The result of the face top and width detection is shown in Figure 4, these were marked on the picture as part of the computer simulation.



Fig 4

As seen in Figure 4, the edges of the face are not accurate. Using the edges found in this initial step would never localize the eyes, since the eyes fall outside the determined boundary of the face. This is due to the blobs of black pixels on the face, primarily in the eye area, as seen in Figure 2b. To fix this problem, an algorithm to remove the black blobs was developed.

3.4. Removal of noise:

The removal of noise in the binary image is very straightforward. Starting at the top, (x_2, y_2), move left on pixel

by decrementing $x2$, and set each y value to white (for 200 y values). Repeat the same for the right side of the face. The key to this is to stop at left and right edge of the face; otherwise the information of where the edges of the face are will be lost. Figure 5, shows the binary image after this process



Fig 5. Binary picture after noise removal

After removing the black blobs on the face, the edges of the face are found again. As see below, the second time of doing this results in accurately finding the edges of the face.



Fig 6. Face edges found after second trial

3.5 Finding Intensity Changes on the Face:

The next step in locating the eyes is finding the intensity changes on the face. This is done using the original image, not the binary image. The first step is to calculate the average intensity for each y – coordinate. This is called the horizontal average, since the averages are taken among the horizontal values. The valleys (dips) in the plot of the horizontal values indicate intensity changes. When the horizontal values were initially plotted, it was found that there

were many small valleys, which do not represent intensity changes, but result from small differences in the averages. To correct this, a smoothing algorithm was implemented. The smoothing algorithm eliminated and small changes, resulting in a more smooth, clean graph. After obtaining the horizontal average data, the next step is to find the most significant valleys, which will indicate the eye area. Assuming that the person has a uniform forehead (i.e.; little hair covering the forehead), this is based on the notion that from the top of the face, moving down, the first intensity change is the eyebrow, and the next change is the upper edge of the eye, as shown below.

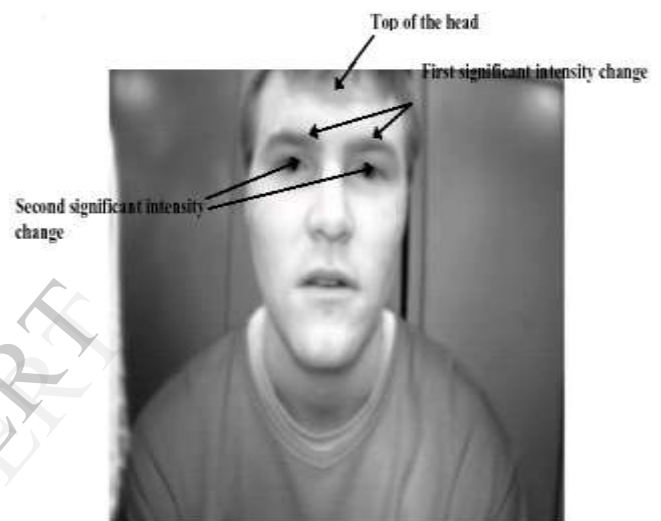
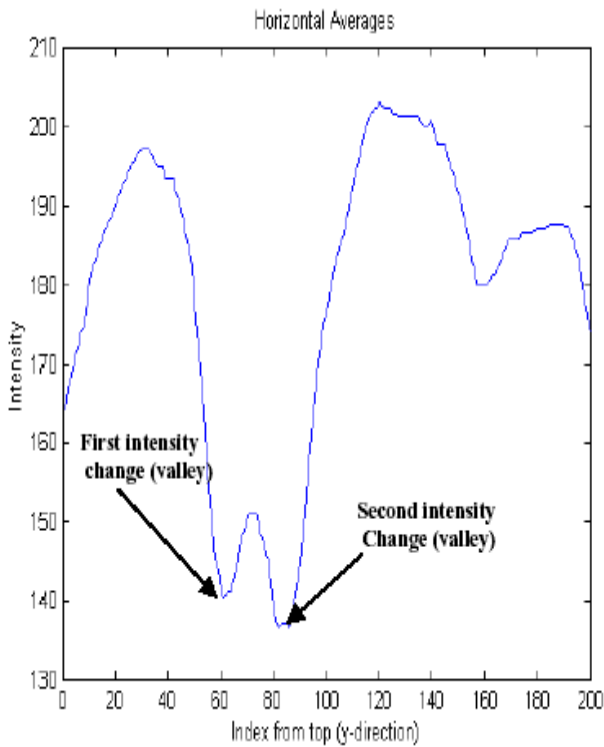


Fig 7. Labels of top of head, and first two intensity changes

The valleys are found by finding the change in slope from negative to positive. And peaks are found by a change in slope from positive to negative. The size of the valley is determined by finding the distance between the peak and the valley. Once all the valleys are found, they are sorted by their size.

3.6 Detection of Vertical Eye Position:

The first largest valley with the lowest y – coordinate is the eyebrow, and the second largest valley with the next lowest y -coordinate is the eye. This is shown in Figures.



a) Graph of horizontal averages of the left side of the face



b) Position of the left eye found from finding the valleys in a

4. Drowsiness Detection Function:

4.1 Determining the State of the Eyes:

The state of the eyes (whether it is open or closed) is determined by distance between the first two intensity changes found in the above step. When the eyes are closed, the distance between the y – coordinates of the intensity changes is larger if compared to when the eyes are open. This is shown in Figure.

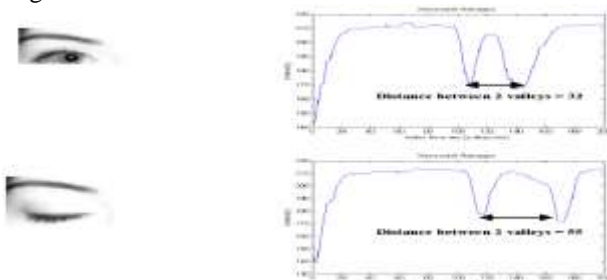


Fig 8. Comparison of open and closed eyes

The limitation to this is if the driver moves their face closer to or further from the camera. If this occurs, the distances will vary, since the number of pixels the face takes up varies, as seen below. Because of this limitation, the system developed assumes that the driver’s face stays almost the same distance from the camera at all times.

4.2 Judging Drowsiness:

When there are 5 consecutive frames find the eye closed, then the alarm is activated, and a driver is alerted to wake up. Consecutive number of closed frames is needed to avoid including instances of eye closure due to blinking. Criteria for judging the alertness level on the basis of eye closure count is based on the results found in a previous study [9].

4.3 Limitations:

With 80% accuracy, it is obvious that there are limitations to the system. The most significant limitation is that it will not work with people who have very dark skin. This is apparent, since the core of the algorithm behind the system is based on binarization. For dark skinned people, binarization doesn’t work.

Another limitation is that there cannot be any reflective objects behind the driver. The more uniform the background is, the more robust the system becomes. For testing purposing, a black sheet was put up behind the subject to eliminate this problem.

For testing, rapid head movement was not allowed. This may be acceptable, since it can be equivalent to simulating a tired driver. For small head movements, the system rarely loses track of the eyes. When the head is turned too much sideways there were some false alarms.

5. Conclusion:

A non-invasive system to localize the eyes and monitor fatigue was developed. Information about the head and eyes position is obtained through various self-developed image processing algorithms. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for too long, a warning signal is issued. In addition, during monitoring, the system is able to automatically detect any eye localizing error that might have occurred. In case of this type of error, the system is able to recover and properly localize the eyes. The following conclusions were made:

- Image processing achieves highly accurate and reliable detection of drowsiness.
- Image processing offers a non-invasive approach to detecting drowsiness without the annoyance and interference.
- A drowsiness detection system developed around the principle of image processing judges the driver’s alertness level on the basis of continuous eye closures.

All the system requirements described above were met.

6. Bibliography:

- [1] Davies, E.R. "Machine Vision: theory, algorithms, and practicalities", Academic Press: San Diego, 1997.
- [2] Dirt Cheap Frame Grabber (DCFG) documentation, file dcfg.tar.z available from <http://cis.nmclites.edu/ftp/electronics/cookbook/video/>
- [3] Eriksson, M and Papanikolopoulos, N.P. "Eye-tracking for Detection of Driver Fatigue", IEEE Intelligent Transport System Proceedings (1997), pp 314-319.
- [4] Gonzalez, Rafel C. and Woods, Richard E. "Digital Image Processing", Prentice Hall: Upper Saddle River, N.J., 2002.
- [5] Grace R., et al. "A Drowsy Driver Detection System for Heavy Vehicles", Digital Avionic Systems Conference, Proceedings, 17th DASC. The AIAA/IEEE/SAE, I36/1- I36/8 (1998) vol. 2.
- [6] Perez, Claudio A. et al. "Face and Eye Tracking Algorithm Based on Digital Image Processing", IEEE System, Man and Cybernetics 2001 Conference, vol. 2 (2001), pp 1178-1188.
- [7] Singh, Sarbjit and Papanikolopoulos, N.P. "Monitoring Driver Fatigue Using Facial Analysis Techniques", IEEE Intelligent Transport System Proceedings (1999), pp 314-318.
- [8] Ueno H., Kanda, M. and Tsukino, M. "Development of Drowsiness Detection System", IEEE Vehicle Navigation and Information Systems Conference Proceedings, (1994), ppA1-3,15-20.
- [9] Weirwille, W.W. (1994). "Overview of Research on Driver Drowsiness Definition and Driver Drowsiness Detection," 14th International Technical Conference on Enhanced Safety of Vehicles, pp 23-26.