# REAL TIME DROWSY DRIVER DETECTION USING HAARCASCADE SAMPLES

Dr.Suryaprasad J, Sandesh D, Saraswathi V, Swathi D, Manjunath S

PES Institute of Technology-South Campus, Bangalore, India

## ABSTRACT

*With the growth in population, the occurrence of automobile accidents has also seen an increase. A detailed analysis shows that, around half million accidents occur in a year , in India alone. Further , around 60% of these accidents are caused due to driver fatigue. Driver fatigue affects the driving ability in the following 3 areas, a) It impairs coordination, b) It causes longer reaction times, and, c)It impairs judgment.  Through this paper, we provide a real time monitoring system using image processing, face/eye detection techniques. Further, to ensure real-time computation, Haarcascade samples are used to differentiate between an eye blink and drowsy/fatigue detection.*

## KEYWORDS

*Face detection, Eye Detection, Real-time system, Haarcascade, Drowsy driver detection, Image Processing, Image Acquisition .*

## 1. INTRODUCTION

According to survey [1],driver fatigue results in over 50% of the road accidents each year. Using technology to detect driver fatigue/drowsiness is an interesting challenge that would help in preventing accidents. In the past various efforts have been reported in the literature on approaches for drowsiness detection  of automobile driver. In the last decade alone, many countries have begun to pay great attention to the automobile driver safety problem. Researchers have been working on the detection of automobile driver's drowsiness using various techniques, such as physiological detection and Road monitoring techniques. Physiological detection techniques take advantage of the fact that sleep rhythm of a person is strongly correlated with brain and heart activities. However, all the research  till date in this approach need electrode contacts on the automobile drivers' head, face, or chest making it non-implementable in real world scenarios. Road monitoring is one of the most commonly used technique,  systems  based on this approach, include Attention assist by Mercedes, Fatigue Detection System by Volkswagon, Driver Alert by Ford, Driver Alert Control by Volvo. All the mentioned techniques monitor the road and driver behavior characteristics to detect the drowsiness of the automobile driver.  Few parameters used include whether the driver is following the lane rules , appropriate usage of the indicators etc.. , If there are aberrations in these parameters, above the tolerance level then the system concludes that the driver is drowsy. This approach is inherently flawed as monitoring the road to detect drowsiness is more of an indirect approach and also lacks accuracy.

Sundarapandian et al. (Eds) : ICAITA, SAI, SEAS, CDKP, CMCA-2013
pp. 45–54, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3805

In this paper we propose a direct approach that makes use of vision based techniques to detect drowsiness. The major challenges of the proposed technique include (a) developing a real time system, (b) Face detection, (c) Iris detection under various conditions like driver position, with/without spectacles, lighting, etc (d) Blink detection, and (e) Economy. The focus will be placed on designing a real-time system that will accurately monitor the open or closed state of the driver's eyes. 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 the observation of eye movements and blink patterns in a sequence of images of a face extracted from a live video. This paper contains 4 sections, Section 2 discusses about the high-level aspects of the proposed system, Section 3 discusses about the detailed implementation, Section 4 discusses about the testing and validation of the developed system and Section 5 presents the conclusion and future opportunities.

## 2. SYSTEM DESIGN

The system architecture of the proposed system is represented in Fig 1,

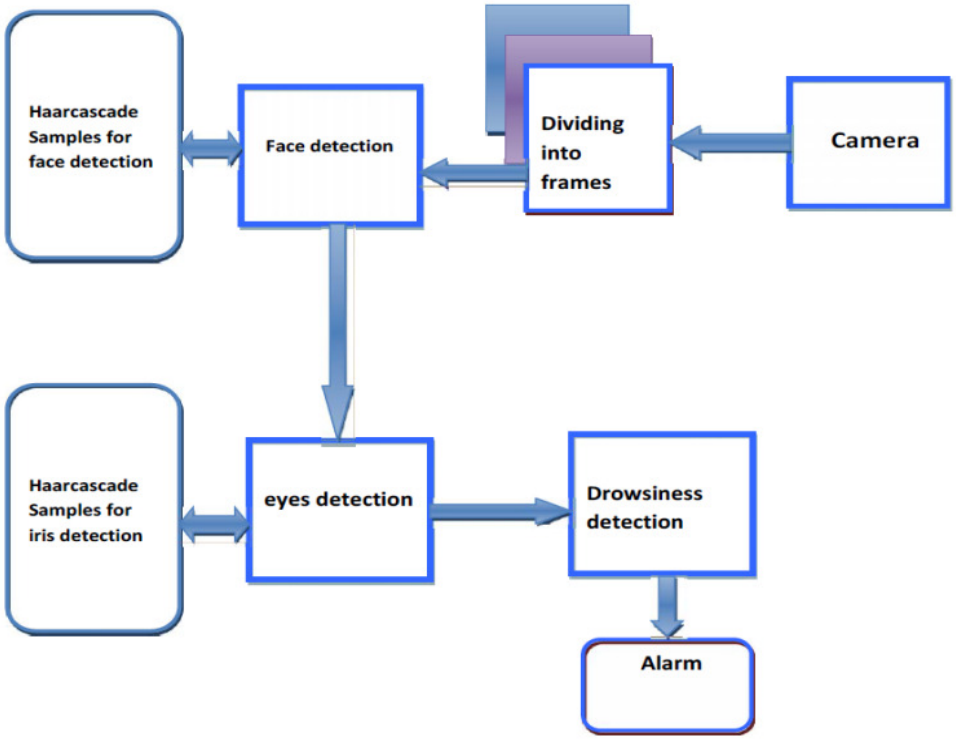

Fig 1. Proposed System Architecture

Fig 1. showcases the various important blocks in the proposed system and their high level interaction. It can be seen that the system consists of 5 distinct modules namely,(a) Video acquisition, (b) Dividing into frames, (c) Face detection, (d) Eye detection and (e) Drowsiness detection. In addition to these there are two external typically hardware components namely, Camera for video acquisition and an audio alarm. The functionality of each these modules in the system can be described as follows:

**Video acquisition:** Video acquisition mainly involves obtaining the live video feed of the automobile driver. Video acquisition is achieved, by making use of a camera.

**Dividing into frames:** This module is used to take live video as its input and convert it into a series of frames/ images, which are then processed.

**Face detection:** The face detection function takes one frame at a time from the frames provided by the frame grabber, and in each and every frame it tries to detect the face of the automobile driver. This is achieved by making use of a set of pre-defined Haarcascade samples.

**Eyes detection:** Once the face detection function has detected the face of the automobile driver, the eyes detection function tries to detect the automobile driver's eyes. This is achieved by making use of a set of pre-defined Haarcascade samples.

**Drowsiness detection:** After detecting the eyes of the automobile driver , the drowsiness detection function detects if the automobile driver is drowsy or not , by taking into consideration the state of the eyes , that is , open or closed and the blink rate.

As the proposed system makes use of OpenCV libraries, there is no necessary minimum resolution requirement on the camera. The schematic representation of the algorithm of the proposed system is depicted in Fig 2. In the proposed algorithm, first video acquisition is achieved by making use of an external camera placed in front of the automobile driver. The acquired video is then converted into a series of frames/images. The next step is to detect the automobile driver's face , in each and every frame extracted from the video.

As indicated in Figure 2, we start with discussing face detection which has 2 important functions (a) Identifying the region of interest, and (b) Detection of face from the above region using Haarcascade. To avoid processing the entire image, we mark the region of interest. By considering the region of interest it is possible to reduce the amount of processing required and also speeds up the processing, which is the primary goal of the proposed system.

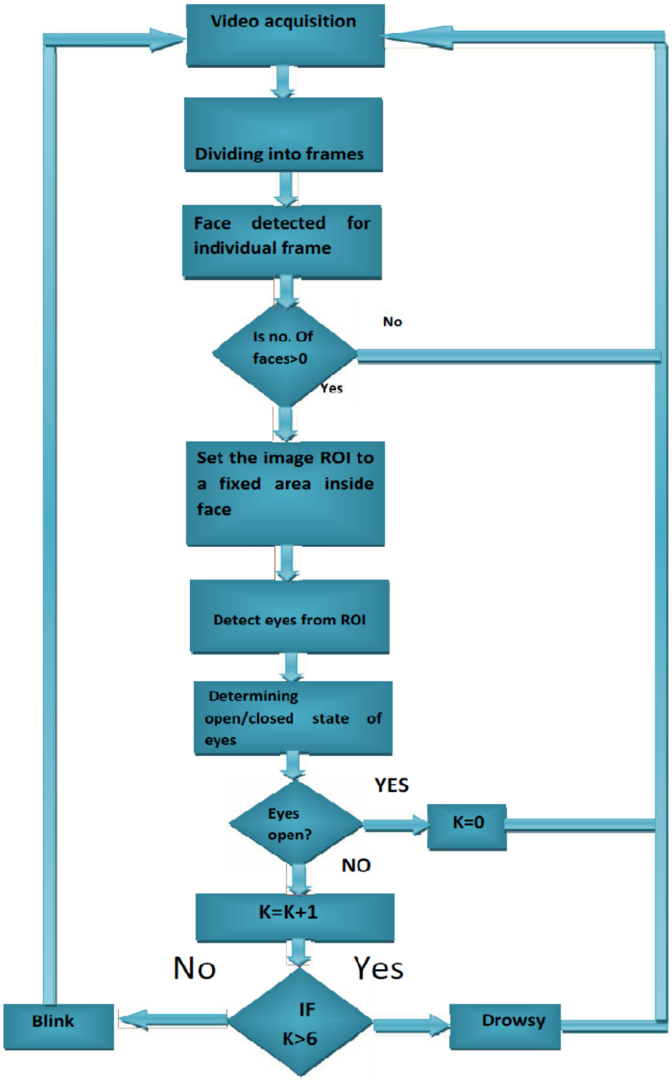

Fig 2. High Level System Flow

For detecting the face, since the camera is focused on the automobile driver, we can avoid processing the image at the corners thus reducing a significant amount of processing required. Once the region of interest is defined face has been detected, the region of interest is now the face, as the next step involves detecting eyes. To detect the eyes, instead of processing the entire face region , we mark a region of interest within the face region which further helps in achieving the primary goal of the proposed system. Next we make use of Haarcascade Xml file constructed for eye detection, and detect the eyes by processing only the region of interest . Once the eyes have been detected, the next step is to determine whether the eyes are in open/closed state, which is achieved by extracting and examining the pixel values from the eye region. If the eyes are detected to be open, no action is taken. But, if eyes are detected to be closed continuously for two seconds according to[6] , that is a particular number of frames depending on the frame rate, then it means that the automobile driver is feeling drowsy and a sound alarm is triggered. However, if the closed states of the eyes are not continuous, then it is declared as a blink.

# 3. IMPLEMENTATION

The implementation details of each the modules can be explained as follows:

## 3.1 Video Acquisition

OpenCV provides extensive support for acquiring and processing live videos. It is also possible to choose whether the video has to be captured from the in-built webcam or an external camera by setting the right parameters. As mentioned earlier, OpenCV does not specify any minimum requirement on the camera, however OpenCV by default expects a particular resolution of the video that is being recorded, if the resolutions do not match, then an error is thrown. This error can be countered, by over riding the default value, which can be achieved, by manually specifying the resolution of the video being recorded.

## 3.2 Dividing into frames

Once the video has been acquired, the next step is to divide it into a series of frames/images. This was initially done as a 2 step process. The first step is to grab a frame from the camera or a video file, in our case since the video is not stored, the frame is grabbed from the camera and once this is achieved, the next step is to retrieve the grabbed frame. While retrieving , the image/frame is first decompressed and then retrieved . However, the two step process took a lot of processing time as the grabbed frame had to be stored temporarily. To overcome this problem, we came up with a single step process, where a single function grabs a frame and returns it by decompressing.

## 3.3 Face detection

Once the frames are successfully extracted the next step is to detect the face in each of these frames. This is achieved by making use of the Haarcascade file for face detection. The Haarcascade file contains a number of features of the face , such as height , width and thresholds of face colors., it is constructed by using a number of positive and negative samples. For face detection, we first load the cascade file. Then pass the acquired frame to an edge detection function, which detects all the possible objects of different sizes in the frame. To reduce the amount of processing, instead of detecting objects of all possible sizes, since the face of the automobile driver occupies a large part of the image, we can specify the edge detector to detect only objects of a particular size, this size is decided based on the Haarcascade file, wherein each Haarcascade file will be designed for a particular size. Now, the output the edge detector is stored in an array. Now, the output of the edge detector is then compared with the cascade file to identify the face in the frame. Since the cascade consists of both positive and negative samples, it is required to specify the number of failures on which an object detected should be classified as a negative sample. In our system, we set this value to 3, which helped in achieving both accuracy as well as less processing time. The output of this module is a frame with face detected in it.

## 3.4 Eye detection

After detecting the face, the next step is to detect the eyes, this can be achieved by making use of the same technique used for face detection. However, to reduce the amount of processing, we mark the region of interest before trying to detect eyes. The region of interest is set by taking into account the following:

- The eyes are present only in the upper part of the face detected.
- The eyes are present a few pixels lower from the top edge of the face.

Once the region of interest is marked, the edge detection technique is applied only on the region of interest, thus reducing the amount of processing significantly. Now, we make use of the same technique as face detection for detecting the eyes by making use of Haarcascade Xml file for eyes detection. But, the output obtained was not very efficient, there were more than two objects classified as positive samples, indicating more than two eyes. To overcome this problem, the following steps are taken:

- Out of the detected objects, the object which has the highest surface area is obtained. This is considered as the first positive sample.
- Out of the remaining objects , the object with the highest surface area is determined. This is considered as the second positive sample.
- A check is made to make sure that the two positive samples are not the same.
- Now, we check if the two positive samples have a minimum of 30 pixels from either of the edges.
- Next , we check if the two positive samples have a minimum of 20 pixels apart from each other.

After passing the above tests, we conclude that the two objects i.e positive sample 1 and positive sample 2 , are the eyes of the automobile driver.

## 3.5 Drowsiness detection

Once the eyes are detected, the next step is to determine if the eyes are in closed or open state. This is achieved by extracting the pixel values from the eye region. After extracting , we check if these pixel values are white, if they are white then it infers that the eyes are in the open state, if the pixel values are not white then it infers that the eyes are in the closed state.

This is done for each and every frame extracted. If the eyes are detected to be closed for two seconds or a certain number of consecutive frames depending on the frame rate, then the automobile driver is detected to be drowsy. If the eyes are detected to be closed in non consecutive frames, then We declare it as a blink.

If drowsiness is detected, a text message is displayed along with triggering an audio alarm. But, it was observed that the system was not able to run for an extended period of time , because the conversion of the acquired video from RGB to grayscale was occupying too much memory. To overcome this problem, instead of converting the video to grayscale, the RGB video only was used for processing. This conversion resulted in the following advantages,

- Better differentiation between colors, as it uses multichannel colors.
- Consumes very less memory.
- Capable of achieving blink detection, even when the automobile driver is wearing spectacles.

Hence there were two versions of the system that was implemented; the version 1.0 involves the conversion of the image to grayscale. Currentlyversion 2.0 makes use of the RGB video for processing.

## 4. TESTING

The tests were conducted in various conditions including:

1. Different lighting conditions.
2. Drivers posture and position of the automobile drivers face.
3. Drivers with spectacles.

### 4.1 Test case 1: When there is ambient light

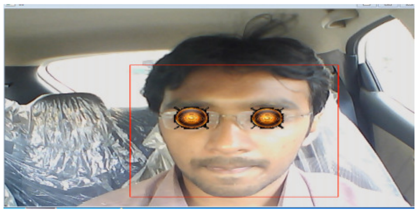

Fig 3. Test Scenario #1 – Ambient lighting

**Result:** As shown in Fig 3, when there is ambient amount of light, the automobile driver's face and eyes are successfully detected.

### 4.2 Test case 2: Position of the automobile drivers face

#### 4.2.1 Center Positioned

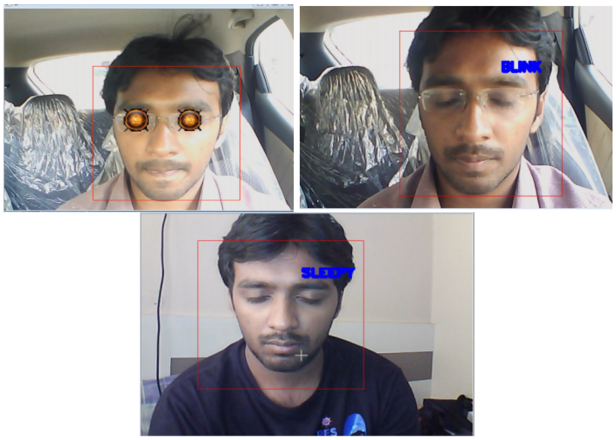

Fig 4. Test Scenario Sample#2 driver face in center of frame

**RESULT** : As shown in Fig 4,When the automobile driver's face is positioned at the Centre, the face, eyes , eye blinks, and drowsiness was successfully detected.

### 4.2.2 Right Positioned

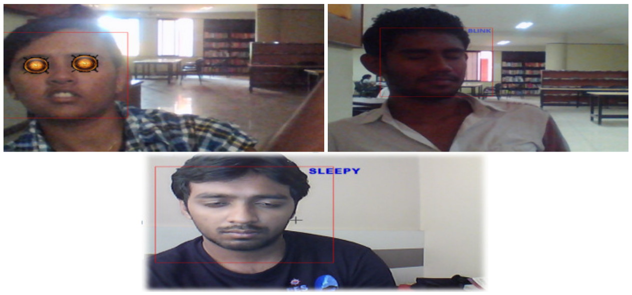

Fig 5. Test Scenario Sample#3 – driver face to right of frame

**RESULT**: As shown in Fig 5,When the automobile driver's face is positioned at the Right, the face, eyes , eye blinks, and drowsiness was successfully detected.

### 4.2.3 Left Positioned

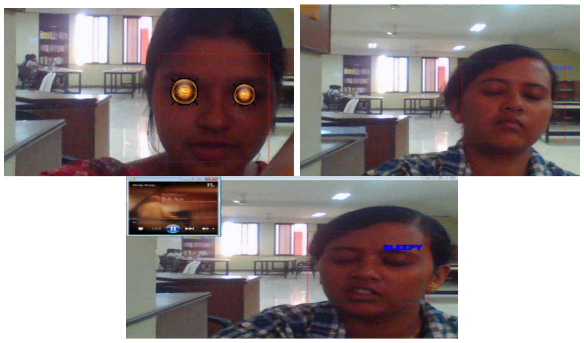

Fig 6. Test Scenario Sample#4- driver face to left of frame

**RESULT** : As shown in screen snapshot in Fig 6,when the automobile driver's face is positioned at the Left, the face, eyes , eye blinks, and drowsiness was successfully detected.

## 4.3 Test case 3: When the automobile driver is wearing spectacles

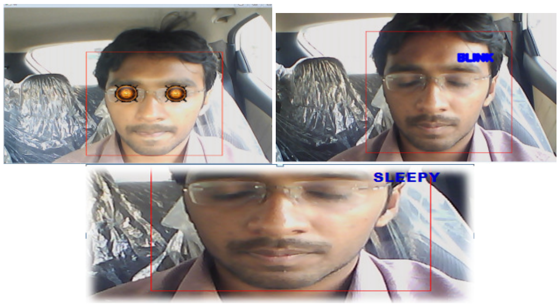

Fig 7. Test Scenario Sample#5 – Driver with spectacles

**RESULT** : As shown in screen snapshot in Fig 7,When the automobile driver is wearing spectacles, the face, eyes , eye blinks, and drowsiness was successfully detected.

### 4.4 Test case 4: When the automobile driver’s head s tilted

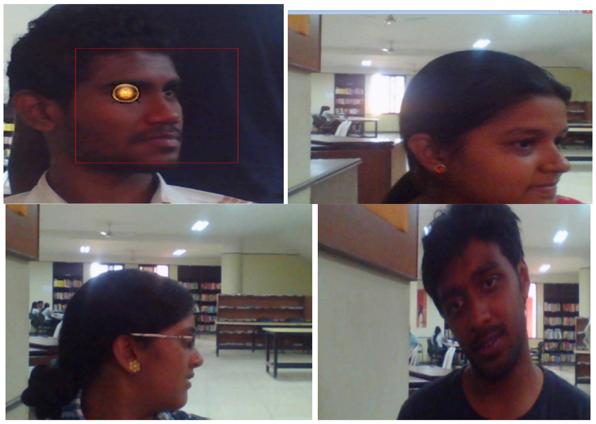

Fig 8. Test Scenario Sample#6 – Various posture

**RESULT** : As shown in screen snapshot in Fig 8,when the automobile driver's face is tilted for more than 30 degrees from vertical plane, it was observed that the detection of face and eyes failed.

The system was extensively tested even in real world scenarios, this was achieved by placing the camera on the visor of the car , focusing on the automobile driver. It was found that the system gave positive output unless there was any direct light falling on the camera.

## 5. CONCLUSION

The primary goal of this project is to develop a real time drowsiness monitoring system in automobiles. We developed a simple system consisting of 5 modules namely(a) video acquisition, (b) dividing into frames, (c) face detection, (d) eye detection, and (e)drowsiness detection . Each of these components can be implemented independently thus providing a way to structure them based on the requirements.

Four features that make our system different from existing ones are:

(a) Focus on the driver, which is a direct way of detecting the drowsiness
(b) A real-time system that detects face,iris, blink, and driver drowsiness
(c) A completely non-intrusive system, and
(d) Cost effective

### 5.1 Limitations

The following are some of the limitations of the proposed system.

- The system fails, if the automobile driver is wearing any kind of sunglasses.
- The system does not function if there is light falling directly on the camera.

## 5.2 Future Enhancement

The system at this stage is a “Proof of Concept” for a much substantial endeavor. This will serve as a first step towards a distinguished technology that can bring about an evolution aimed at ace development. The developed system has special emphasis on real-time monitoring with flexibility, adaptability and enhancements as the foremost requirements.

Future enhancements are always meant to be items that require more planning, budget and staffing to have them implemented. There following are couple of recommended areas for future enhancements:

- **Standalone product:** It can be implemented as a standalone product, which can be installed in an automobile for monitoring the automobile driver.

- **Smart phone application:** It can be implemented as a smart phone application, which can be installed on smart phones. And the automobile driver can start the application after placing it at a position where the camera is focused on the driver.

## REFERENCES

[1] Ruian Liu,et.a;, "Design of face detection and tracking system," Image and Signal Processing (CISP), 2010 3rd International Congress on , vol.4, no., pp.1840,1844, 16-18 Oct. 2010

[2] Xianghua Fan, et.al, "The system of face detection based on OpenCV," Control and Decision Conference (CCDC), 2012 24th Chinese , vol., no., pp.648,651, 23-25 May 2012

[3] Goel, P, et.al., "Hybrid Approach of Haar Cascade Classifiers and Geometrical Properties of Facial Features Applied to Illumination Invariant Gender Classification System," Computing Sciences (ICCS), 2012 International Conference on , vol., no., pp.132,136, 14-15 Sept. 2012

[4] Parris, J., et.al, "Face and eye detection on hard datasets," Biometrics (IJCB), 2011 International Joint Conference on , vol., no., pp.1,10, 11-13 Oct. 2011

[5] Peng Wang., et.a;., "Automatic Eye Detection and Its Validation," Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on , vol., no., pp.164,164, 25-25 June 2005

[6] Picot, A. et.al., "On-Line Detection of Drowsiness Using Brain and Visual Information," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.42, no.3, pp.764,775, May 2012

[7] Xia Liu; Fengliang Xu; Fujimura, K., "Real-time eye detection and tracking for driver observation under various light conditions," Intelligent Vehicle Symposium, 2002. IEEE , vol.2, no., pp.344,351 vol.2, 17-21 June 2002.