**DROWSINESS RECOGNIZATION USING MATLAB**

BY

B. SHREEJA (18C91A0423)

K. SAI SREE (18C91A0449)

R. PRANITHA (18C91A0485)

G. MAMATHA (18C91A0432)

UNDER THE GUIDANCE OF

Mrs. BHARGAVI

ASSISTANT PROFESSOR



                      Department Of Electronics and Communication Engineering

      HOLY MARY INSTITUTE OF TECHNOLOGY AND SCIENCE

           (Approved by AICTE New Delhi, Affiliated to JNTU, Hyderabad, NAAC “A” grade)

                  BOGARAM (V), KEESARA (M), MEDCHAL DIST-501 301.

**CONTENTS**

1. ABSTRACT 3

2. INTRODUCTION. 4

3. IMAGE PROCESSING 5

3.1. IMAGE ACQUISITION.

3.2. FACE DETECTION

3.3. EYE DETECTION

3.4. DROWSINESS DETECTION

3.5. ALARM

3.6. SYSTEM

3.7. DESCRIPTION

3.8. FLOW CHART

4. MATLAB IMPLEMENTATION 11

5. EXPERIMENT AND RESULT 13

6. CONCLUSION 15

7. LIMITATIONS 15

8. FUTURE WORK 15

9. REFERENCES 16

ABSTRACT

Nowadays the driver safety in the vehicle is one of the most wanted system to avoid accidents. Our objective of the project is to ensure the safety system. For enhancing the safety, we are detecting the eye blinks of the driver and estimating the driver status and control the vehicle accordingly.

The principle of the proposed system in this project using MATLAB based on the real time facial images analysis for warning the driver of drowsiness or in attention to prevent traffic accidents. The facial images of driver are taken by a camera which is installed on the dashboard in front of the driver. An algorithm and an inference are proposed to determine the level of fatigue by measuring the eyelid blinking duration and face detection to track the eyes and warn the driver accordingly. If the eyes are found closed for 5 or 8 consecutive frames, the system draws the conclusion that the driver is falling asleep and issues a warning signal. The system is also able to detect when the eyes cannot be found.

Aim of this project is implementing the system as a prototype by capturing the live images of the eyes and fed them into the Microcontroller in which the MATLAB software is used to process the video and convert it into frames and process it accordingly. Some customized algorithms are coded in MATLAB for image segmentation of eyes from the entire image and image recognition of the eyes and face position.

**INTRODUCTION**

Driver fatigue is the major reason why half (50%) of road accidents takes place. It is an interesting challenge in today’s date to detect drowsiness in order prevent accidents. Various experiments have been done earlier about the drowsiness detection of driver. In the past few years, many countries became curious to pay high attention towards driver’s safety problems. Researchers have been making various efforts to invent techniques for the detection of drowsy driver such as monitoring of road and physiological techniques which requires the contact of electrode with our body such as chest, face making it an implantable method. In this thesis, we described the direct method which can detect drowsiness without any help of electrode using various detection function.

Aim of this project is implementing the system as a prototype by capturing the live images of the eyes and fed them into the Microcontroller in which the MATLAB software is used to process the image and convert it into frames and process it accordingly. Some customized algorithms are coded in MATLAB for image segmentation of eyes from the entire image and image recognition of the eyes and face position.

**IMAGE PROCESSING**

**3.1 IMAGE ACQUISITION**

It mainly involves obtaining the image of the Automobile driver. It can be acquired with the help of camera with diving into different frames. Live image is taken as its input and then it converts those images into the series of images which are further proceeded to make various operations.

**3.2 FACE DETECTION**

Face detection activity takes one of the frame at a time ‘t’ from frame grabber which later tries to detect the face of Automobile driver in every frame. And it can be done with the help of Vision Cascade samples.

**3.3** **EYE DETECTION**

After detecting the face of Automobile driver with the face detection function, the eyes detection can be done with the help of eyes detection function. This can be done with Voila Jones Algorithm

**3.4 DROWSINESS DETECTION**

Once the eyes of Automobile driver are detected, the drowsiness detection function detects whether the driver is drowsy or not, by taking into consideration whether the eyes are open or closed that is the state of the eyes.

An algorithm is developed in the software to generate a frame with a square for the face, indicating a face image for image processing. And this method is performed to detect drowsiness patterns more efficiently. To make face detection more accurate, feature-based technique was selected rather than other techniques. Of all the haar-like features, face detection is regarded easy and efficient. Because of the face's haar-like characteristics, we used the face detection method based on the cascade of ad boost technique trained classifiers. There are several methods, such as cascade and integral image, for better face detection results.

A person driving a car

Description automatically generated with medium confidence

Once the face has been detected, it is cropped out to detect the eyes on it. Eye region receives an evident function

on the face's horizontal projection curve, according to research. Therefore, the face region is removed from the image to obtain the location of the eyes and then the face horizontal projection is worked out. For an image with X number of rows and Y number of columns, the horizontal projection is a sum of the intensities of the pixel in each line

By the application of Harris method, the pupil is roughly located. After the pupil is identified in the initial frame, the region of eye is located by Kalman filtering.

Firstly, the extraction of the eye candidates happens by using the following steps. It starts with identifying the values denoted by A, B, C by using the below equation.

Where represents the gradient of the image in the respective, x and y directions. Following that, the value of R is obtained by using the equation below.

The points with the highest R values are selected as the suitable eye candidates points in the specified region. Among these points, the eye area is well detected in the next phase. Colour entropy is used to remove irrelevant ones in the eye region.

The eye image has many characteristics. Variations in the concentrations of lightness and the shape of the eye element make it difficult to detect these characteristics. One helpful factor for identifying eye location is the uneven variation in sclera lightness concentrations. In this region, the blood vessels add to this entropy on the sclera surface. Some points with the largest R values are indicated to guarantee right pupil detection. These points are regarded as candidates for pupils. Colour entropy is assessed for each candidate to eliminate the ones irrelevant to the pupil, and then colour entropy calculation happens. Entropy probability is then derived from equation given below.

Where p(x) represents the probability of the pixel of. Each pixel's probability function, , is the pixel intensity frequency in the selected area divided by the total pixel frequency. Two regions in the candidates for eye with the highest entropy are concluded as the region including the pupil. The centre of these regions is considered as the pupil's centre.

Graphical user interface

Description automatically generated

After the eyes have been detected, the system will continue to identify patterns of drowsiness. The separation of the ROI as the eyes will start processing the patterns. Then the extraction method starts, this analyses the closing time of the eye and the opening distance of the eye.

As it can be seen in Figure 4, the program requires 4 frames, in which the first frame captures the driver's eyes, places the eyes inside second one, and lastly, the program analyses that the eye is closed in the third frame and the externally fed template is in the fourth frame. The template is used as a reference to see if the eyes are opened or closed.

Fig.Eye detection

Graphical user interface, application, PowerPoint

Description automatically generated

The process demonstrated by Figure 4 same as the earlier one, except the eyes are open in this situation which can be identified by a green circle in the third frame.

Graphical user interface

Description automatically generated

Blinking rate of eyes is used as the key element to identify drowsiness pattern. Blinking of eyes are influenced by two major factors, they are, rate of blinking and opening of the eye. The below two equations were developed based on the two factors mentioned

Rate of blinking or the flicker frequency is the number of blinks made by the driver during a given time. According to the below equation.

In an individual with drowsiness, the blinking frequency is about twenty-one per minute whereas the individual blinks only about fifteen times per minute in the ordinary state.

Opening of the eye is the amplitude in the closing point between the two eyelids when the eyes are opened and closed. The process starts when the eyelids cover the pupils of the eyes. That the eyelids above and below are open.

And PERCLOS is used to calculate this factor to determine the percentage of eyelid closure. The following mathematical formula is used as described in equation 2.

Once the drowsiness in driver is detected, necessary steps should be taken immediately to avoid fatal accidents. In this project two different techniques are used to handle this situation. Firstly, an alarm will be activated and then the vehicle itself will be controlled using the accelerator and the brake

**3.5. ALARM**

The device will generate an alarm sound after detection to alert the individual for falling asleep, using Arduino and a buzzer in this prototype. The alarm will vary depending on the pattern identified corresponding to level of sensitivity detected in the individual. The driver is therefore not accustomed to the monotonous tone of the alarm and as a result ignore the warning.

**3.6. SYSTEM**

The system includes a camera that captures the image. The type of camera integrated in this prototype is model DC-B615 belongs to the brand HP and the camera has a higher resolution and excellent clarity as well as linked to a laptop via USB. Using MATLAB software, the computer will perform image processing to detect sleepiness in the individual. At last, an alarm and a braking system is integrated to the system by using an Arduino. Figure 5 shows the system.

A computer on a table

Description automatically generated with medium confidence

**3.7. DESCRIPTION**

An image here is processed with the help of Viola Jones Algorithm. In the first step, the face is detected and then the eyes are sectioned and are processed to detect drowsiness/ fatigue. With the help of rectangular () function, it measures the length and width of an eye. The position, length and width can be obtained with the help of Vision class. ‘Cascade Object Detector’ which is an inbuilt object detector in MATLAB is used to detect eye. The obtained image is then cropped using ‘imcrop( )’ command in MATLAB. Coloured image is converted into grey scale image using ‘rgb2grey()’ function. In order to create box, we used ‘bbox( )’ function. ‘immadjust( )’ help in adjusting the level of contrast. Accuracy of the project depends on the quality of Web-Camera. The processing time is increased with the ‘getsnapshot( )’ function in MATLAB. By defining the region of interest for detection is done by using Viola Jones Algorithm in order to reduce computational requirements of the system. Using MATLAB Image processing, sleep detection system can be explained.

**3.8. FLOW CHART**

Diagram

Description automatically generated

**MATLAB CODE**

|  |
| --- |
| clear all  clf('reset');  cam=webcam(); %create webcam object  right=imread('RIGHT.jpg');  left=imread('LEFT.jpg');  noface=imread('no\_face.jpg');  straight=imread('STRAIGHT.jpg');  detector = vision.CascadeObjectDetector(); % Create a detector  for face using Viola-Jones  detector1 = vision.CascadeObjectDetector('EyePairSmall');  %create detector for eyepair  while true % Infinite loop to continuously detect the face  vid=snapshot(cam); %get a snapshot of webcam  vid = rgb2gray(vid); %convert to grayscale  img = flip(vid, 2); % Flips the image horizontally  bbox = step(detector, img); % Creating bounding box using  detector  if ~ isempty(bbox) %if face exists  biggest\_box=1;  fori=1:rank(bbox) %find the biggest face  ifbbox(i,3)>bbox(biggest\_box,3)  biggest\_box=i;  end  end  faceImage = imcrop(img,bbox(biggest\_box,:)); % extract the face from the image  bboxeyes = step(detector1, faceImage); % locations of the eyepair using detector  subplot(2,2,1),subimage(img);  hold on; % Displays full image  fori=1:size(bbox,1) %draw all the regions that contain face  rectangle('position', bbox(i, :), 'lineWidth', 2, 'edgeColor', 'y');  end  subplot(2,2,3),subimage(faceImage); %display face image  if ~ isempty(bboxeyes) %check it eyepair is available  biggest\_box\_eyes=1;  fori=1:rank(bboxeyes) %find the biggest eyepair  ifbboxeyes(i,3)>bboxeyes(biggest\_box\_eyes,3)  biggest\_box\_eyes=i;  end  end  bboxeyeshalf=[bboxeyes(biggest\_box\_eyes,1),bboxeyes(bigges t\_box\_eyes,2), bboxeyes(biggest\_box\_eyes,3)/3,bboxeyes(bigge st\_box\_eyes,4)]; %resize the eyepair width in half  eyesImage = imcrop(faceImage,bboxeyeshalf(1,:)); %extract the half eyepair from the face image eyesImage = imadjust(eyesImage); %adjust contrast  r = bboxeyeshalf(1,4)/4;  [centers, radii, metric] = imfindcircles(eyesImage, [floor(r-r/4)  floor(r+r/2)], 'ObjectPolarity','dark', 'Sensitivity', 0.93); % Hough Transform  [M,I] = sort(radii, 'descend')  eyesPositions = centers;  subplot(2,2,2),subimage(eyesImage); hold on;  viscircles(centers, radii,'EdgeColor','b');  if ~isempty(centers)  pupil\_x=centers(1);  disL=abs(0-pupil\_x); %distance from left edge to center  point  disR=abs(bboxeyes(1,3)/3-pupil\_x);%distance from right edge  to center point  subplot(2,2,4);  ifdisL>disR+16  subimage(right);  else if disR>disL  subimage(left);  else  subimage(straight);  end  end  end  end  else  subplot(2,2,4);  subimage(noface);  end  set(gca,'XtickLabel',[],'YtickLabel',[]);  hold off;  end |

**EXPERIMENT AND RESULT**

The number of blinks an individual does in different kinds of situation is shown in the table below. Table 1 Number of blinks per minute

Table

Description automatically generated

This data is essential since it is required to be expressed coherently in MATLAB because of the research of the system phases, we can get the hierarchical understanding of how the system will operate.

The figure shows the result of a test run. The person is said to have drowsiness based on the blinking rate

Tests were conducted in a well-lit room as well as a less lit room (refer to figure 9 in appendix). The results suggested that the light has a very little impact in the process since both setting had a similar percentage of accuracy

When tested with different types of individuals, it was found that the system is unable to detect drowsiness pattern accurately for people with spectacles. Since the system uses blinking rate as the key factor, it is a challenge for it to detect the eyes under spectacles, especially in a less lit room.

Graphical user interface, application

Description automatically generated

Table

Description automatically generated

The figure below shows a falsely identified blink, when tested with an individual wearing spectacle.

The table above states that the average accuracy of the system is 77.94% and the average accuracy when tested with people without wearing glasses is 88.24% and the average accuracy for people wearing glasses is 67.50%.

The table above states that the average accuracy of the system is 77.94% and the average accuracy when tested with people without wearing glasses is 88.24% and the average accuracy for people wearing glasses is 67.50%.

The table above states that the average accuracy of the system is 77.94% and the average accuracy when tested with people without wearing glasses is 88.24% and the average accuracy for people wearing glasses is 67.50%.

Therefore, in order to obtain high accuracy, the camera should be placed in a much shorter range, preferably on the dashboard with an alignment of about 50 degrees.

**CONCLUSION**

In this way, we have successfully implemented drowsiness detection using MATLAB and Viola Jones Algorithm. The developed system has been successfully tested and its limitations are identified.

**LIMITATIONS**

Limitations of the proposed system are as follows: If the driver is using sunglasses then the computation doesn’t work.If there is the striking light directly on the web-camera then the system doesn’t work.

**FUTURE WORK**

It is required to make the speed of vehicle slow or slow down the speed of vehicle in real time drowsiness detection. In order to create continous monitoring, threshold drowsiness detection should be kept aside. While monitoring the drowsiness continuously, when the level exceeds certain value a signal is generated which directly controls the braking of vehicle.

**REFERENCE**

1. Australian Transport Safety Bureau, 2002. Australian Transport Safety Bureau Annual Review 2002. [Online] Available at: https://www.atsb.gov.au/media/36022/annual\_review\_20 02.pdf [Accessed 21 January 2019].\
2. Safety Policy & Education Branch, Ontario, 2004. Ontario Road Safety Annual Report 2004. [Online] Available at: http://www.bv.transports.gouv.qc.ca/per/1022891/12\_200 4.pdf [Accessed 21 January 2019].
3. Sunday Times, 2011. “Fatigue and Drowsy Driving”: An experts’ report. [Online] Available at: http://www.sundaytimes.lk/110501/News/nws\_01102.htm l [Accessed 21 January 2019]