SLEEP DETECTION SYSTEM USING MATLAB IMAGE PROCESSING

DIGVIJAY R, VIGNESH SRIDHARAN

BE 3rd Year EEE, Anna University

Abstract: The objective was to develop a non-invasive system for detecting the closing of eyes of a person driving an automobile and provide an alarm indication thus preventing road accidents from occurring. Live video relay of the driver's eyes is processed using Image Processing in MATLAB to detect whether the eye is closed for more than a fixed duration thus indicating conditions of fatigue, alcohol consumption etc. The system proves to be more accurate and safe compared to the existing sleep detection system developed using Infrared Sensors and Micro-processors.

I. CLAIMS

What is claimed is,

1. A non-invasive system for detecting the closing of eyes of a person driving an automobile thereby identifying the condition of the person (Fatigue, Alcohol induced, sleep derived) using live video relay obtained from a webcam and processed using Image Acquisition Toolbox and Image Processing Toolbox in MATLAB 2013a.

2. The method of Claim 1, wherein a further alarm system is incorporated for warning the driver or the surrounding personnel.

3. The method of Claim 1, Advantageous over the current system which uses Infrared Sensors and micro-processors,
   □ Greater accuracy over the current system which is susceptible to various other light bands like sunlight, headlights of other vehicles.
   □ Higher degree of safety over the current system which causes Photo-keratitis (Inflammation of the Cornea) and retinal damage because of long-term exposure to infrared rays.

4. The method of Claim 1 offers more flexibility in terms of changing the module design for different driving environments.

5. The method of Claim 1, is more efficient and economical than the existing method designed using Hardware components.

II. INTRODUCTION

With the ever increasing population and usage of automobiles, there is an increase in the number of fatalities as well. India, unfortunately, boasts of a very high number of 142,485 traffic-related fatalities [1]. There are a number of reasons that can be attributed to this astonishing statistic, a few of primary concern being Fatigue, Alcohol Consumption and Sleep Deprivation.

Hence, we developed a method to test for the closing of eyes of a person driving an automobile and provide an alarm indication if the eyes are detected to be closed for more than a specified amount of time. MATLAB 2013a Image processing techniques are adopted to detect the closure of the eye by sectioning only that portion of the driver's face from a live video relay obtained using a front camera.

III. DESCRIPTION

1. Processing a Static Image

The method is based on the Viola-Jones algorithm [2]. The project started off with detecting the eyes of a static image stored in the computer. The first step involved storing the image in a variable mentioning the location and the type of image. From the given image, only the eyes are sectioned out and processed to detect for closure or fatigue. The image is processed only to detect the eye region of the image by giving the position, width and height of the region as inputs to the rectangle () function. The position, width and height are obtained by using the Vision class in MATLAB. The built-in object detector function Cascade Object Detector is used to detect the eyes. The Eye Detect object is given as input to the step function along with the image and the values returned correspond to the X-Coordinate, Y-Coordinate, Width and Height of the eye region. The image is then cropped using the imcrop () function with one input as the n*4 matrix and the other being the image itself. The RGB image thus obtained is first converted to its equivalent grayscale form using the rgb2gray() function. This is followed by converting the thus obtained gray scale image to its black and white form using the im2bw () function. The BW image thus obtained is then dilated to get only the eyes. The purpose of performing the dilation function is to enhance the foreground features.

IM2 = imdilate (IM,SE) dilates the grayscale, binary, or packed binary image IM, returning the dilated image, IM2. SE is a structuring element object, or array of structuring element objects, returned by the STREL function. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (white pixels). Thus
areas of foreground pixels grow in size while holes within those regions become smaller.

**MATLAB Code**

```matlab
EyeDetect = vision.CascadeObjectDetector('EyePairBig');
l = imread('C:\Users\Home\Desktop\baby.jpg');
BB = step(EyeDetect, l);
a = size(BB);
disp(a);
figure, imshow(l);
l = rgb2gray(l);
rectangle('Position', BB, 'LineWidth', 4, 'LineStyle', '-', 'EdgeColor', 'b');
title('Eyes Detection');
Eyes = imcrop(l, BB);
figure, imshow(Eyes), title('Eyes cropped');
Eyes_gray = rgb2gray(Eyes);
figure, imshow(Eyes_gray), title('Gray scale');
Eyes_BW = im2bw(Eyes_gray, 0.12);
figure, imshow(Eyes_BW), title('Black white');
se = strel('square', 1);
Eyes_Dilated = imdilate(Eyes, se);
figure, imshow(Eyes_Dilated), title('Dilated');
```

2. Processing a live feed
The next step of the project was to perform the same on a live video feed obtained by either using an external USB operated camera or by using the built-in webcam. The accuracy of this method of eye detection is based on the sensitivity of the camera. It is found to have a direct relationship with the accuracy. The greater the accuracy needed, the better quality of webcam has to be used. The first step towards implementing this, is to first identify the webcam drivers installed and then configure the webcam to obtain the necessary video feed. The associated webcams were identified by using the imaqhwinfo() function. The next step was to configure the webcam and assign the video properties.

This involved setting the FramesPer Trigger and Returned Color Space properties of the video object. The live feed was then obtained using the start(video-object) function.

The vision.Cascade Object Detector statement for detecting the face was used to initialize an object FaceDetect. The next step was to crop the image such that only the face is retained static for further eye detection. This is achieved by visualizing the live video feed as individual frames and processing each frame distinctly. The vision.Cascade Object Detector for detecting the eye region was used to initialize an object EyeDetect. The video capturing was initially performed for the first 50 frames. The video was converted to individual frames using the getsnapshot() function which returns a matrix corresponding to an RGB image. The next step involved was similar to identifying the eye region in a static image, the difference being instead of the image being stored in the computer memory, it is stored virtually in a MATLAB script.

Since the getsnapshot() function works by contacting the webcam every time it is called, the processing time is increased. In order to minimize the time taken by the getsnapshot(), the triggerconfig() property of the video object was set to manual mode. The EyeDetect object is given as input to the step function along with the image and the values returned correspond to the X-Coordinate, Y-Coordinate, Width and Height of the eye region.

The image is then cropped using the imcrop() function with one input as the n cross 4 matrix and the other being the image itself. The RGB image thus obtained is first converted to its equivalent grayscale form using the rgb2gray() function. This is followed by converting the thus obtained gray scale image to its black and white form using the im2bw() function.
The BW image thus obtained is then dilated to get only the eyes. The dilated image thus obtained is a matrix. The sum of all the elements of the matrix are obtained using the sum() function. The value of this sum is compared with a temporary value(20), which is used to decide whether the eyes are closed or not. This temporary value is the specified time frame to check for closure of the eyes. This is done by calculating the sum of the elements of the matrix for a particular set of frames(50).

If the value does not exceed the threshold and stays constant for a short time it corresponds to a normal blink. If the value stays for more than 5 frames it means that the eye has been closed for more than 5 frames indicating a drowsy state. The MATLAB code immediately sends a warning message that the driver is drowsy. This can be improved by interfacing an audio amplifier and speaker to produce a warning sound or can be interfaced with a vibrator to produce a vibration.

MATLAB Code

```matlab
vid = videoinput('winvideo',2);
set(vid, 'FramesPerTrigger', 1);
set(vid, 'TriggerRepeat', inf);
set(vid, 'ReturnedColorspace', 'rgb');
set(vid, 'Timeout',50);
vid.FrameGrabInterval = 5;

%start the video acquisition here
triggerconfig(vid, 'manual');
start(vid);
for ii=1:50
    FaceDetect = vision.CascadeObjectDetector('FrontalFaceLBP');
    im = getSnapshot(vid);
    trigger(vid);
    figure(1);imshow(im);
    BB=step(FaceDetect,im);
    if numel(BB)>0
        continue;
    end;
    %a=size(BB);
    %disp(a);
    rectangle('Position', BB, 'LineWidth', 4, 'LineStyle', '-', 'EdgeColor', 'r');
    title('Face Detection');
    Face=imcrop(1,BB);

    %figure(2);imshow(Face);
    EyeDetect = vision.CascadeObjectDetector('EyePairSmall');
    BB2=step(EyeDetect,Face);
    b=size(BB2);
    if numel(BB2)>0
        disp('Not detected');
        continue;
    else
        disp('Detected');
    end;

    Eyes=imcrop(Face, BB2);
    figure(3);imshow(Eyes);
    Eyes_BW = im2bw(Eyes,0.12);
    figure(1);
    imshow(Eyes_BW);
    se=strel('square',1);
    Eyes_Dilated=imdilate(Eyes_BW,se);
    figure(1);
    imshow(Eyes_Dilated);title('Dilated');
    end;
    stop(vid);
    delete(vid);
    clear all
```

CONCLUSION

The sleep detection system proposed has a specific set of advantages over the existing method and proves to be more efficient and economical in comparison.
Although there is a constraint on the quality of the camera required for processing the live video feed, this method offers a non-invasive system for sleep detection. An improvement over the existing method, this system is indeed a feasible and easily implementable alternative.

REFERENCES


★★★