ISSN: 2277-9655
Impact Factor: 4.116

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

DROWSY DRIVER DETECTION USING IMAGE PROCESSING

Puja Seemar [*], Anurag Chandna
M. Tech Scholar[*] Dept. of Computer Science Engineering
Assistant Professor, Dept. of Computer Science Engineering
Roorkee College of Engineering, Roorkee, UK, India

DOI: 10.5281/zenodo.823118

ABSTRACT

Driver errors and carelessness contribute most of the road accidents occurring nowadays. The major driver errors are caused by drowsiness, drunken and reckless behavior of the driver. This paper focuses on a driver drowsiness detection system in Intelligent Transportation System, which focuses on abnormal behavior exhibited by the driver using Raspberry pi single board computer. The capability of driving support systems to detect the level of driver's alertness is very important in ensuring road safety. By observation of blink pattern and eye movements, driver fatigue can be detected early enough to prevent collisions caused by drowsiness. In the proposed system a non-intrusive driver drowsiness monitoring system has been developed using computer vision techniques. Based on the simulation results, it was found that the system has been able to detect drowsiness in spite of driver wearing spectacles as well as the darkness level inside the vehicle. Moreover the system is capable of detecting drowsiness within time duration of about two seconds. The detected abnormal behavior is corrected through alarms in real time. We have also done a transformation of an image from 2d to 3d using wavelet analysis. Here we also compared the wavelet technique with other techniques namely stereo-photogrammetric & edge information technique. The image conversion from 2d to 3d can be done by finding the edges of an image. This edge detection can be used in various fields of image processing, image analysis, image pattern recognition, and computer vision, as well as in human vision. In this, we did our experiment using wavelet technique and the results when compared with the stereo photogrammetry and edge information techniques we find that the wavelet technique gives better result. From the past work, we have observed that Wavelet analysis is an easy method for 2D to 3D analysis.

KEYWORDS: Image detection,Transformation, Photogrammetric, Wavelet Analysis.

I. INTRODUCTION

No one think about the journey without own vehicle. Driver fatigue causes the maximum accidents by cars, and lesser are by two wheelers. Four wheeler drivers easily go to resting mode and sometimes he/she enters to the drowsy state. Gaining information about behavior patterns generally inaccessible to unconscious introspection .The term drowsiness can be considered as the state of reduced alertness usually accompanied by performance and psychophysiological changes that result in loss of alertness. A system has been designed with computer vision research which is dedicated precisely for detecting human blink. It is of utmost importance to measure eye movement during psychophysical tasks and experiments to study; Eye movement control; Gaining information about behavior patterns generally in accessible to conscious introspection; Examining information processing strategies; controlling task performance during experiments requires fixation; Control task performances that require precise knowledge of the subject. In the trucking industry, 57% of fatal truck accidents are due to driver fatigue. It is the principal reason of heavy truck crashes. Seventy percent of American drivers details driving fatigued. The National Highway Traffic Safety Administration (NHTSA) estimates that there are 100 000 crashes that are caused by sleepy drivers and result in more than 1500 fatalities and 71 000 injuries. This problem will increase day by day.

So, there is a requirement of designing detection systems for the driver drowsiness or inattention and can produce some warning alarms to alert the driver and the further people in the vehicle. Driver's behaviors such as visual interruption, false determination on the environment and improper handling of emergencies just the accidents have close connection to crash.

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 project 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.

Working Principle

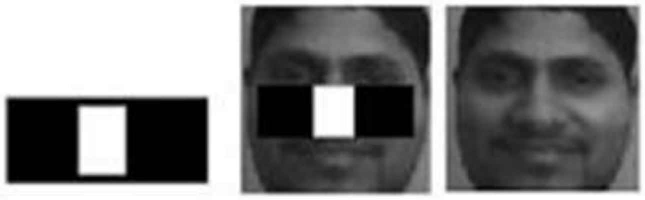
A Drowsy Driver Detection System has been developed, using a non-intrusive machine vision based concepts. The system uses a web camera that points directly towards the driver's face and monitors the driver's head movements in order to detect fatigue. In such a case when fatigue is detected, a warning signal is issued to alert the driver. The algorithm developed is unique to any currently published papers, which was a primary objective of the project. The system deals with detecting eyes, nose and mouth within the specific segment of the image. If these are not found for 5 consecutive frames, the system draws the conclusion that the driver is falling asleep.

Viola Jones Algorithm

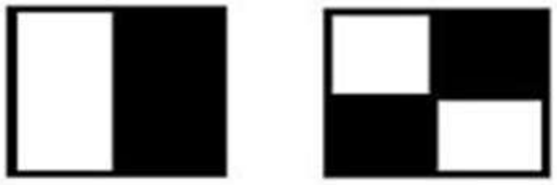
The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection. This algorithm is implemented in Opencv . The method is based on the Viola-Jones algorithm

Feature types and evaluation

The characteristics of Viola–Jones algorithm which make it a good detection algorithm are:
Robust – very high detection rate (true-positive rate) & very low false-positive rate always.
Real time – For practical applications at least 2 frames per second must be processed.
Face detection only (not recognition) - The goal is to distinguish faces from non-faces (detection is the first step in the recognition process).
The algorithm has four stages:
Haar Feature Selection
Creating an Integral Image
Adaboost Training
Cascading Classifiers
The features sought by the detection framework universally involve the sums of image pixels within rectangular areas. As such, they bear some resemblance to Haar basis functions, which have been used previously in the realm of image-based object detection. However, since the features used by Viola and Jones all rely on more than one rectangular area, they are generally more complex. The figure on the right illustrates the four different types of features used in the framework. The value of any given feature is the sum of the pixels within clear rectangles subtracted from the sum of the pixels within shaded rectangles. Rectangular features of this sort are primitive when compared to alternatives such as steerable filters. Although they are sensitive to vertical and horizontal features, their feedback is considerably coarser.



Haar Feature looks similar to the bridge of the nose



Haar Feature that looks similar to the eye region

Fig: Haar classifier

Haar Feature that looks similar to the bridge of the nose is applied onto the face
Haar Feature that looks similar to the eye region which is darker than the upper cheeks is applied onto a face 3rd and 4th kind of Haar Feature
Haar Features – All human faces share some similar properties. These regularities may be matched using Haar Features.
A few properties common to human faces:
The eye region is darker than the upper-cheeks.
The nose bridge region is brighter than the eyes.
Composition of properties forming matchable facial features:
Location and size: eyes, mouth, bridge of nose
Value: oriented gradients of pixel intensities
The four features matched by this algorithm are then sought in the image of a face (shown at left).
Rectangle features:
Value = Σ (pixels in black area) - Σ (pixels in white area)
Three types: two-, three-, four-rectangles, Viola & Jones used two-rectangle features
For example: the difference in brightness between the white &black rectangles over a specific area
Each feature is related to a special location in the sub-window

Creating an Integral Image

An image representation called the integral image evaluates rectangular features in constant time, which gives them a considerable speed advantage over more sophisticated alternative features. Because each feature's rectangular area is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, any three-rectangle feature in eight, and any four-rectangle feature in nine.
This is performed on a live video that was recorded by using the webcam of system or we can also use external USB camera. 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.
The vision Cascade Object Detector statement for detecting the face was used to initialize an object Face Detect. 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 Eye Detect. 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.

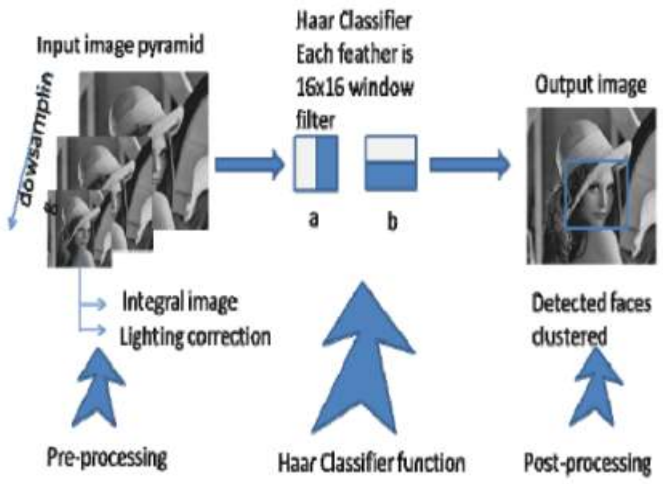


Fig: Face detection Flow based on haar Classifier.

Learning algorithm

The speed with which features may be evaluated does not adequately compensate for their number, however. For example, in a standard 24x24 pixel sub-window, there are a total of M = 162 , 336 {\displaystyle M=162,336} possible features, and it would be prohibitively expensive to evaluate them all when testing an image. Thus, the object detection framework employs a variant of the learning algorithm AdaBoost to both select the best features and to train classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.

$$h(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^{M} \alpha_j h_j(\mathbf{x})\right)$$

h (x) = sign (∑ j = 1 M α j h j (x)) {\displaystyle h(\mathbf {x})={\text{sign}}\left(\sum _{j=1}^{M}\alpha _{j}h_{j}(\mathbf {x})\riEach weak classifier is a threshold function based on the feature fi

$$h_j(\mathbf{x}) = \begin{cases} -s_j & \text{if } f_j < \theta_j \\ s_j & \text{otherwise} \end{cases}$$

f j {\displaystyle

The threshold valueϴi {\displaystyle \theta _{j}} and the polaritySj {\displaystyle s_{j}\in \pm 1} are determined in the training, as well as the coefficients {\displaystyle \alpha _{j}}aj.

To detect a feature:

1. Define and set up your cascade object detector using the constructor.
2. Call the step method with the input image, I, the cascade object detector object, detector, points PTS, and any optional properties. See the syntax below for using the step method.

Use the step syntax with input image, I, the selected Cascade object detector object, and any optional properties to perform detection.

BBOX = step(detector,I) returns BBOX, an *M*-by-4 matrix defining *M* bounding boxes containing the detected objects. This method performs multiscale object detection on the input image, I. Each row of the output matrix, BBOX, contains a four-element vector, [x y width height], that specifies in pixels, the upper-left corner and size of a bounding box. The input image I, must be a grayscale or truecolor (RGB) image.
BBOX = step(detector,I,roi) detects objects within the rectangular search region specified by roi. You must specify roi as a 4-element vector, [*x y width height*], that defines a rectangular region of interest within image I. Set the 'UseROI' property to true to use this syntax.

MATLAB code for eye detction:

```
im = imread(‘image.jpg’);% Reading image from which eyesneed to be detected
fd = vision.CascadeObjectDetector('EyePairBig');
bbox = step(fd, im);
vo = insertObjectAnnotation(im,'rectangle',bbox,'EYES');
imshow(vo)
if ~isempty(bbox) &&isrow(bbox)
imc = imcrop(im,bbox);
figure;
imshow(imc)
imc  = rgb2gray(imc);
imc1 = ~im2bw(imc,.10);
imshow(imc1)
end
```

Here is a simplified version of the learning algorithm is reported
Input**:** Set of positive and negative training images with their labels . If image is a face , if not

1. Initialization: assign a weight to each image .
2. For each feature with
    1. Renormalize the weights such that they sum to one.
    2. Apply the feature to each image in the training set, then find the optimal threshold and polarity

       that minimizes the weighted classification error. That is where
    3. Assign a weight to that is inversely proportional to the error rate. In this way best classifiers are considered more.
    4. The weights for the next iteration, i.e. , are reduced for the images that were correctly classified.

3. Set the final classifier to

Cascade architecture

- On average only 0.01% of all sub-windows are positive (faces)
- Equal computation time is spent on all sub-windows
- Must spend most time only on potentially positive sub-windows.
- A simple 2-feature classifier can achieve almost 100% detection rate with 50% FP rate.
- That classifier can act as a 1st layer of a series to filter out most negative windows
- 2nd layer with 10 features can tackle “harder” negative-windows which survived the 1st layer, and so on…
- A cascade of gradually more complex classifiers achieves even better detection rates. The evaluation of the strong classifiers generated by the learning process can be done quickly, but it isn’t fast enough to run in real-time. For this reason, the strong classifiers are arranged in a cascade in order of complexity, where each successive classifier is trained only on those selected samples which pass through the preceding classifiers. If at any stage in the cascade a classifier rejects the sub-window under inspection, no further processing is performed and continue on searching the next sub-window. The cascade therefore has the form of a degenerate tree. In the case of faces, the first classifier in the cascade – called the attentional operator – uses only two features to achieve a false negative rate of approximately 0% and a false positive

rate of 40%.The effect of this single classifier is to reduce by roughly half the number of times the entire cascade is evaluated.

In cascading, each stage consists of a strong classifier. So all the features are grouped into several stages where each stage has certain number of features.

The job of each stage is to determine whether a given sub-window is definitely not a face or may be a face. A given sub-window is immediately discarded as not a face if it fails in any of the stages. A simple framework for cascade training is given below:

- User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
- User selects target overall false positive rate Ftarget.
- P = set of positive examples
- N = set of negative examples

```
F(0) = 1.0; D(0) = 1.0; i = 0

while F(i) > Ftarget
  i++
  n(i) = 0; F(i)= F(i-1)

  while F(I) > f x F(i-1)
    n(i) ++
    use P and N to train a classifier with n(I) features using Adaboost
    Evaluate current cascaded classifier on validation set to determine F(i) & D(i)
    decrease threshold for the ith classifier
      until the current cascaded classifier has a detection rate of at least d x D(i-1) (this also affects F(i))
    N = ∅
    if F(i) > Ftarget then
      evaluate the current cascaded detector on the set of non-face images
      and put any false detections into the set N.
```

The cascade architecture has interesting implications for the performance of the individual classifiers. Because the activation of each classifier depends entirely on the behavior of its predecessor, the false positive rate for an entire cascade is:

F = ∏ i = 1 K f i . {\displaystyle F=\prod _{i=1}^{K}f_{

Similarly, the detection rate is:

D = ∏ i = 1 K d i . {\displaystyle D=\prod _{i=1}^{K}d_{i}.

Thus, to match the false positive rates typically achieved by other detectors, each classifier can get away with having surprisingly poor performance. For example, for a 32-stage cascade to achieve a false positive rate of 10 − 6 {\displaystyle 10^{-6}} each classifier need only achieve a false positive rate of about 65%. At the same time, however, each classifier needs to be exceptionally capable if it is to achieve adequate detection rates. For example, to achieve a detection rate of about 90%, each classifier in the aforementioned cascade needs to achieve a detection rate of approximately.

RESULTS

To obtain the result a large number of videos were taken and their accuracy in determining eye blinks and drowsiness was tested.

For this project we used a 1.3 megapixel webcam connected to the computer. The webcam had inbuilt white LEDs attached to it for providing better illumination. In real time scenario, infrared LEDs should be used instead of white LEDs so that the system is non-intrusive. An external speaker is used to produce alert sound output in order to wake up the driver when drowsiness exceeds a certain threshold.

The system was tested for different people in different ambient lighting conditions(daytime and nighttime). When the webcam backlight was turned ON and the face is kept at an optimum distance, then the system is able to detect blinks as well as drowsiness with more than 95%

accuracy. This is a good result and can be implemented in real-time systems as well. Sample outputs for various conditions in various images is given below. Two videos were taken; one in which only the eyes were detected and the other in which both face and eyes were detected. Though the two processes have relatively equal accuracy, the computational requirements of the former and lesser than that of the latter

Results with open and closed eyes

Fig: Open eye1

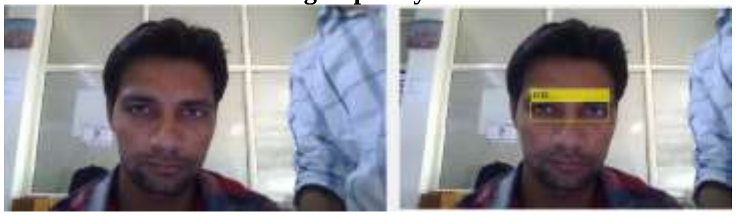


Fig: Open eye2

Fig: Close eye 1

Fig: Close eye 2

Graph When Eye Blinking

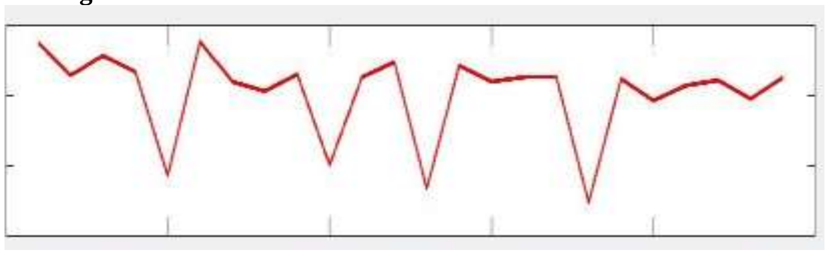


Fig: Blinking test1

Fig: Blinking test2

II. CONCLUSION

(i) The findings carried out by us suggest that is very much possible to detect drowsiness in drivers by analyzing their blink pattern but works on an assumption that all individual develop drowsiness in the same way.

(ii) It was proved that while the system was found in effective in darkness or lack of lighting causing errors due to non detection of eyes it performed impeccably in normal light giving up to 94% accuracy.

(iii) In the real time drowsy driver identification using eye blink detection if the parameters exceed a certain limit warning signals can be mounted on the vehicle to warn the driver of drowsiness.

(iv) Further it is a viable option to design a continuous scale of drowsiness and on crossings a certain threshold value level the systems could generate a signal which would automatically slow down or switch off the motor.This idea presents a system for drowsiness detection in automotive drivers based on image and voice cues. Onboard monitoring of the alertness level of an automotive driver has been a challenging research in transportation safety and management.

(v) In this paper, we have developed a robust real-time embedded platform to monitor the loss of attention of the driver based on image processing.

(vi) The image based algorithm relies on the computation of PERCLOS.

III. REFERENCES

1. Brown I. Driver fatigue. Human Factors, 1994, 36(2):298-314.
2. Dinges D, Mallis M, Maislin G, et al. Final report: Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management. NHTSA,Washington DC, Tech. Rep. DOT HS 808762, 1998.
3. Davies, E.R. "Machine Vision: theory, algorithms, and practicalities", Academic Press: San Diego, 1997.
4. Eriksson, M and Papanikolopoulos, N.P. "Eye-tracking for Detection of Driver Fatigue", IEEE Intelligent Transport System Proceedings (1997), pp 314-319.
5. S. J. Julier, J. K. Uhlmann "Unscented filtering and nonlinear estimation" at Proceedings of the IEEE, 2004, 92(3): 401 – 422.
6. H. Gu, Q. Ji, Z. Zhu "Active facial tracking for fatigue detection" at Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision. Piscataway: IEEE, 2002: 137 – 142.
7. [6] Rainer Lienhart and Jochen Maydt Intel Labs. "An Extended Set of Haar-like Features for Rapid Object Detection" . Intel Corporation, Santa Clara, CA 95052, USA.
8. Atish Udayashankar, Amit R Kowshik, Chandramouli S 2012 „Assistance for the Paralyzed using Eye blink Detection" Fourth International Conference on Digital Home
9. Abdolhossein Fathi Fardin, Abdali-Mohammadi "Camera-based eye blinks pattern detection for intelligent mouse" SIViP DOI 10.1007/s11760-014-0680-1, Received: 5 November 2013 / Revised: 8 July 2014 / Accepted: 11 July 2014 © Springer-Verlag London 2014
10. http://www.ijarse.com/images/fullpdf/1495707088_ijarse661.pdf
11. Suman Deb, Sujay Deb, "Designing an intelligent blink analyzer tool for effective human computer interaction through eye", IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction, Kharagpur, India, December 27-29, 2012
12. http://www.mathworks.in/help/vision/ref/vision.cascadeobjectdetector.step.html

CITE AN ARTICLE

Seemar , P., & Chandna, A. (2017). DROWSY DRIVER DETECTION USING IMAGE PROCESSING. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*, *6*(7), 234-241. doi:10.5281/zenodo.823118