Real Time Driver Drowsiness Detection

Anjana Vasudevan, Vilas H Gaidhane,

ABSTRACT - Driver fatigue is one of the causes of road accidents and fatalities as it results in lack of attention to the road conditions. This paper presents a drowsiness detection system which will alert the driver about their exhaustion. It will focus on the algorithm and the implementation of the system, its scope and limitations. The system first recognizes the drivers' face, tracks their eye and mouth movement to check if the eye is closed and (or) yawning occurs for a prolonged period of time, in terms of frames, and raises an alarm.

Keywords – Drowsiness, detection, image processing, tracking

INTRODUCTION

Drowsiness or fatigue is the state of feeling lethargic or tired. Driver fatigue is among the top four causes of accidents worldwide, causing 7% of crashes in the United States alone [1]. Reducing the number of fatalities due to driver fatigue is a key area of concern and research. Automobile manufacturers such as Volkswagen [2] and Nissan [3] have fatigue detection systems based on the speed of the vehicle, tracking steering wheel movements and braking mechanisms applied by the driver. The traditional approaches to detect drowsiness include:

1. Intrusive detection – which consists of monitoring the drivers' pulse by means of EEG signal extraction and analysis. [4]
2. Non-intrusive detection, which uses image processing and (or) machine learning techniques to monitor monitoring the driver's facial features.

Out of these, non – intrusive fatigue detection methods are highly preferred as they do not hamper the concentration of driver.

There are different algorithms available for monitoring facial features using image processing techniques. Viola Jones algorithm uses Haar features to map out the regions, and uses a classifier (machine learning – Adaboost classifier) to classify the regions [5]. These features can be isolated. The Haar feature selection uses the wavelet transform to identify the features based on skin texture and color. Another approach is to evaluate the face regions by using image processing techniques such as Hough transform for mapping the eye regions [6]. A novel technique based on brightness mapping of facial features [7] is also used to monitor drowsiness.

The system proposed in this paper uses a combination of image processing and machine learning techniques to accurately map out the facial regions. The rest of the paper will discuss the system design, the algorithm implemented, the results and the scope of the fatigue system.

SYSTEM DESIGN

The system design is done as shown:

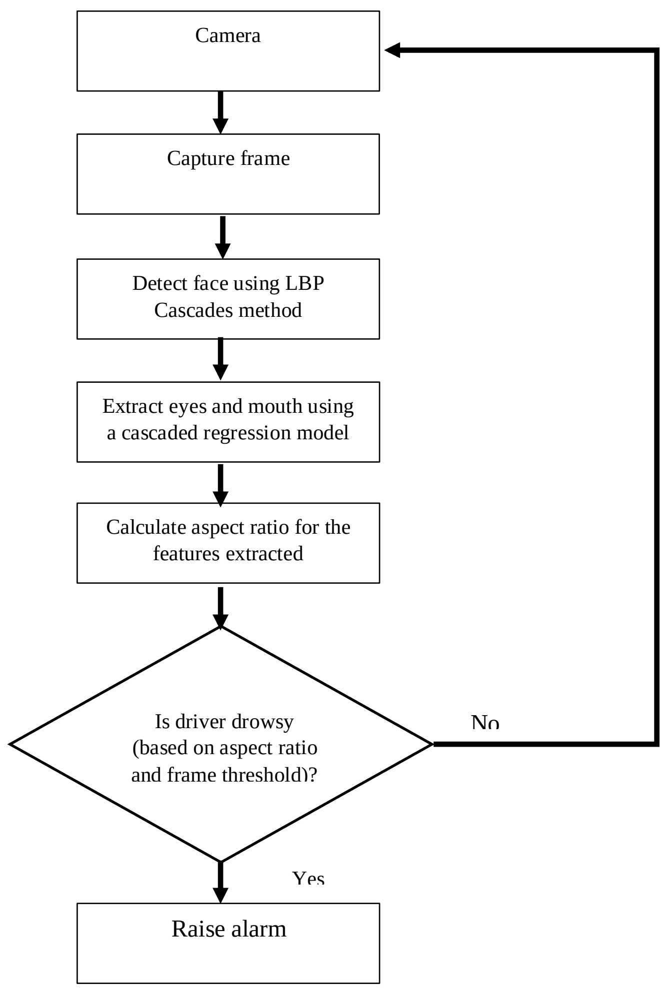

Figure 1- Flow diagram of the system

The algorithm is implemented using Python and consists of the following stages:

1. Face detection
2. Extraction of features – eyes and mouth
3. Calculating aspect ratio

FACE DETECTION

The face detection is done using Local Binary Patterns Method. The algorithm uses a neighborhood of 8 × 8 and computes pixel value based on value of the neighborhood pixels. A histogram is then computed over the neighborhood and normalized, and a feature vector is generated. The output vector gives information about the number of features and type of features (magnitude of the vector) [8]. The ease of implementation of the algorithm and its high processing speed in limited resources (computing speed and power) [9] makes it a preferred candidate to implement face detection.

This algorithm is implemented using OpenCV's LBP cascade classifier. The LBP coefficients are obtained, following which Adaboost classifier is used to map the face regions. Adaboost classifier is a boosting algorithm that strengthens a set of weak boosters or classifiers.

FEATURE EXTRACTION

The features – eyes and mouth are extracted using the Histogram of Gradients – HOG technique. It involves the following steps:

- Preprocessing of the test image / frame by equalizing its color components, such power law transform (contrast stretching techniques)
- Calculating the gradient of the preprocessed image using a suitable operator (e.g. Sobel, Canny edge operators)
- Dividing the resultant image from above into regions called cells (usually 8 × 8 or 16 × 16 regions – pixels)
- Construct a histogram of pixel magnitude – vs – angle of orientation for the gradient image
- Contrast normalizing the above histogram over blocks (2 – 3 cells at a time). [10]

The resultant image from the last step gives the outline of the shape to be found. Since the shape outline may not be precise, its accuracy is improved by machine learning techniques. In this system, the accuracy obtained from the HOG technique is further improved by using a cascade of regression trees using the Gradient Boosting tree algorithm. This requires the following input parameters:

1. The input image
2. The rough shape estimate (obtained by the HOG technique) in the previous iteration.

A weight matrix, learning rate ($0 < \alpha < 1$) are supplied in addition to the inputs. With every iteration, the weights and the learning rates are updated. As there are many features to be extracted, there are a cascade of regressors. The process is repeated until all the regressors reach the target error goal. The error is computed using least squares method. [11] The step by step procedure is shown as below:

1. Initialise

$$F_0\big(I, S(t)\big) = \sum_{i=1}^{n} (\Delta S_i(t) - \gamma)^2$$

where γ is the error correction term and ΔS is the error from the previous iteration.

2. For k epochs, the regressor r is given by:

$$r_{ik} = \Delta S_i(t) - f_{k-1}(I_i, S_i(t))$$

3. Fit a regression tree to the regressor r, and update f(I, S) with the help of gamma and learning rate.

The HOG and its improvisation come prebuilt in the dlib library of Python. Dlib is a C++ library with python bindings to simplify facial recognition. The above implementation of cascade of regressors comes as a 68-point human face model, which is used to extract the features.

ASPECT RATIO AND EVALUATING DROWSINESS

Once the features are identified using the shape predictor model, the marker points that identify the parts are extracted. The aspect ratio is then computed using the traditional Euclidean formula (for 2D space):$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

For the eyes, the aspect ratio is calculated based on the formula:

$EAR = \frac{V1+V2}{2H}$, where:

- V1 and V2 are the Euclidean distances of p2 and p6, p3 and p5 respectively.
- H is the Euclidean distance between p1 and p4 [11]. The points are located as shown in the figure:

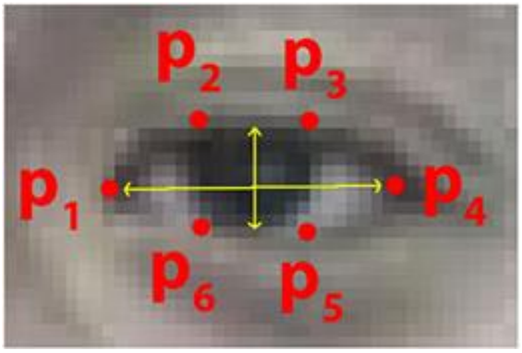

Figure 2: Eye marker points diagram

The above formula is simply the ratio of horizontal distance to the vertical distance. If the ratio goes below 1 (precisely 0.5), a warning is raised. If the ratio continues to be less than the threshold value, an alarm is raised. The formula is used for the left and right eyes separately, and the average of the aspect ratios is used for evaluating drowsiness.

For yawning, the region extracted will contain marker points for both upper and lower part of lips (refer figure). The Euclidean distance between the topmost and the most bottom marker points decides if the person yawns. If the no. of yawns exceeds a certain value, an alarm is raised.

SYSTEM IMPLEMENTATION

The algorithm, as discussed is implemented using Python. It is developed in a 64-bit Windows environment and is tested out using the built-in web camera of Dell Inspiron Laptop. It is divided into three sections – initializing the classifiers (face recognition and the shape predictors), extracting the eyes and mouth region points, and evaluating drowsiness. The output of the system is shown as follows:

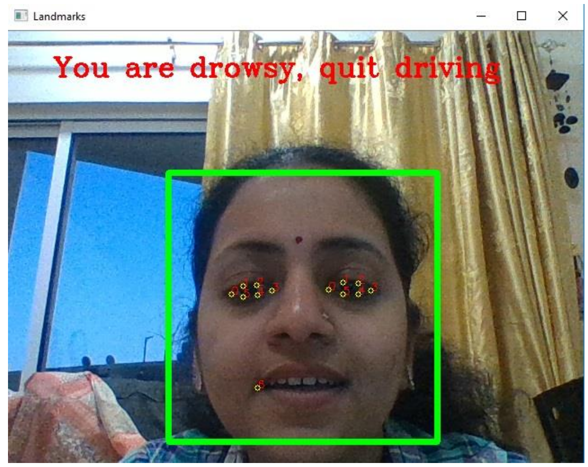

Figure 3: Sample Output of the system

The system was also tested on the ULg Multimodality database, also popularly known as the Drozy database. The database contains videos of 14 subjects, who were subjected to 10-minute psychomotor vigilance tests (that includes sleep deprivation). The results obtained on testing the system with 5 subject videos are shown:

TABLE 1 TEST RESULTS ON DROZY DATABASE

Subject Type	Detecting Landmarks (Y / N)	Detecting Drowsiness (Y / N)	Accuracy = $\frac{No\ of\ frames\ detected}{No\ of\ frames\ actually\ drowsy} \times 100$
1	Y	Y	95%
3	N	Y	92%
7	N	Y	93.83%
9	Y	N	89%
14	N	Y	90%

It was observed that the detection system performed better in excellent illumination – such as sunlight, well lit areas. With spectacles, it performed better with those spectacles that were rimless (no frame for the lens).

References

[1] Governor's Highway safety Asscociation, "Wake Up Call - Understanding Drowsy driving and what States can do".

[2] Volkswagen, "Driver Alert System," [Online]. Available: https://www.volkswagen-newsroom.com/en/driver-alert-system-3932. [Accessed 20 April 2019].

[3] Nissan, "Nissan's "Driver Attention Alert" helps detect erratic driving caused by drowsiness and inattention," 1 April 2015. [Online]. Available: https://nissannews.com/en-US/nissan/usa/releases/nissan-s-driver-attention-alert-helps-detect-erratic-driving-caused-by-drowsiness-and-inattention.

[4] C.-Y. C. L.-W. K. C.-F. C. T.-P. J. S.-F. L. a. C.-T. L. Nikhil R. Pal, "EEG-Based Subject- and Session-independent Drowsiness," *EURASIP Journal on Advances in Signal Processing archive,* vol. 2008, January 2008.

[5] M. J. Paul Viola, "Robust Real-time Object Detection," in *ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001*, 2001.

[6] M. L. C. G. A. D. T. D'Orazio, "A visual approach for driver inattention detection," *Pattern Recognition,* vol. 40, no. 8, pp. 2341-2355 , August 2007.

[7] R. A. Z. Pooneh. R. Tabrizi, "Drowsiness Detection Based on Brightness and Numeral Features of Eye Image," in *IIH-MSP '09 Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing,* 2009.

[8] Y. V. H. a. V. S. VILAS H GAIDHANE, "Emotion recognition using eigenvalues and Levenberg–Marquardt algorithm-based classifier," *Sa¯dhana¯,* vol. 41, no. 4, pp. 415-423, April 2016.

[9] "A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Local Binary Patterns in Several Platforms," *Modelling and Simulation in Engineering*, vol. vol 2015, 2015.

[10] N. D. a. B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* San Diego, CA, USA, USA, 22-25 June 2005.

[11] J. S. Vahid Kazemi, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 23-28 June 2014.