Real-time fatigue features detection

A I Gaidar¹, P Y Yakimov¹,²

¹Samara National Research University, Moskovskoe Shosse, 34A, Samara, Russia, 443086
²Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

e-mail: gajdar97artem@gmail.com, pavel.y.yakimov@gmail.com

Abstract. Fatigue detection is a very important goal because often tired people lose control of a certain task. So the driver falls asleep during a long trip. Driver’s state is very important because one of the main reasons for motor vehicular accidents is related to driver’s fatigue. To prevent accidents a driver fatigue monitoring and control system that works in real time is required. The main purpose of this study to build a base for developing the drowsiness control system. The article presents drowsiness features study, such as the closed eye, the yawn are required for building the system.

1. Introduction

Road safety is certainly one of the most important goals. The reasons for the high accident rate are not only ineffective functioning of the system for ensuring safe road traffic but also in the low discipline of drivers. One of the solutions is to provide a driver with more information about the road conditions [1-2], i.e. creating systems for automatic traffic sign recognition, lane detection, collision warning systems, etc. Currently, driver fatigue is one of the main factors. The tired driver is a reason for 20% of all traffic accidents and 25% of fatal accidents [3].

There are various systems for monitoring driver drowsiness. One type of such systems is based on processing biological signals body driver. The most common approaches are electroencephalography, electrocardiogram, electrooculography, surface electromyogram, galvanic skin reaction, and respiration. [4-5]. These methods create discomfort for a driver.

Another type of system is based on the fatigue measurement scale. The most popular and effective scale is “Karolinska Sleepiness Scale” [6]. Systems with a scale ask the driver about his fatigue status at every interval. Such systems are very intrusive.

The following approach is based on signals that reflect the direct actions of the driver. The driver’s state can be analyzed using signals such as force pressing the pedals, changing the speed of the vehicle, moving the steering wheel, changing the position or lane [7]. A system based on this approach requires an individual approach to each person, so it will not be applicable for a new driver.

The last system is based on monitoring the driver’s face of using a camera and image processing methods to obtain some physical indicators, such as closing the eyes, blinking speed and PERCLOS [8], and detection of yawning [9], gaze direction [10]. This approach is effective. Firstly, the driver’s drowsiness is manifested as a result of features of fatigue on the driver’s face. Secondly, this approach
is unobtrusive. There are commercial products, such as faceLAB and DSS [11], and DFM [12], Viulib [13].

This article is aimed at studying the methods of fatigue needed to build a system based on the approach described in the previous paragraph.

2. Face, eyes and mouth detection

Recently, research to detect a face, as well as eye and mouth are very great importance and are considered one of the most promising areas in object recognition detection.

Each of the detection methods is unstable due to the person’s different pose, facial expressions, face position and orientation, skin color, glasses or long bangs, lighting conditions and image resolution.

2.1. Detection via OpenCV

The detection face is based on the method of Viola-Jones (proposed by P. Viola and M. Jones in 2001) [14].

This method builds a cascade classifier; each of its levels has a huge number of parameters to be checked. This means that the region that didn’t pass the test at the first level of the cascade of classifiers is not checked at the next levels but is discarded as a region that doesn’t contain a face. The classification parameters are image features – functions that calculate the difference in intensities of adjacent image areas. The example of cascade features is shown in Figure 1.

![Cascade features](image1.png)

Fig. 1. Cascade features.

OpenCV already contains a set of trained classifiers to recognize individual parts of the face (eyebrows, nose, lips, eyes, mouth) [15]. But these classifiers are inaccurate. An example of inaccurate detection is presented in Figure 2.

![Face, mouth and eyes detection](image2.png)

Fig. 2. Face, mouth and eyes detection.

A good way to detect areas of small parts of the face is to detect face landmarks.

2.2. Detection via Dlib

Dlib is a less well-known cross-platform library of general-purpose software, also written in the C++ language [16]. Dlib contains software components for working with artificial neural networks, streams, graphical user interfaces, data structures, machine learning, image processing, etc.

Image processing in Dlib is built on the HOG (Histogram of oriented gradients) and SVM (Support vector machine) algorithms.

Dlib doesn’t have detectors for detecting mouth, nose or eyes. There is an only good implementation of the algorithm from the article “One Millisecond Face Alignment with an Ensemble of Regression Trees”.

The model of 68 face landmarks is shown in Figure 3.

![68 face landmarks](image3.png)

Fig. 3. Face landmarks.

2.3. Algorithm comparison

For comparison, we used the image size of 300 × 300, CPU such as Intel Core i7 3632QM, Intel Core i5 8250 and Intel Core i7 9700K.
For testing, the face detection algorithms were run on 10,000 pictures and the processing speed was average.

**Figure 3.** Face landmarks.

Processing speed is shown in Table 1.

|                | Dlib HOG | OpenCV Haar | OpenCV DNN |
|----------------|----------|-------------|------------|
| Intel Core i5 8250 | 57       | 32          | 22         |
| Intel Core i7 3632 QM | 63       | 37          | 16         |
| Intel Core i7 9200K | 148      | 102         | 64         |

**Table 1.** Processing speed in fps.

Comparison of accuracy is shown in Figure 4.

**Figure 4.** Detection accuracy.

Detection of eyes and mouth is performed using a model of 68 face landmarks. Let’s compare existing trained OpenCV and Dlib models. Figure 5 shows the speed of the predictors.

Figure 6 shows the mean square error as a parameter of accuracy for predicting face landmarks. In most applications, it is impossible to know the size of the face in the image in advance. For such cases, it is better to use OpenCV. The most accurately is OpenCV DNN method. If speed is critical, it is better to use a haar-based implementation. Dlib HOG is the fastest method on the processor. But he does not detect faces of small size. A fast and accurate predictor for determining points from Dlib was used with the Dlib HOG + SVM face detector. Figure 7 presents a comparison of the accuracy of the predictor with the two best detectors. As can be seen from the
graph, in almost all videos the predictor works better with the detector from Dlib. The reason is that the predictor works very poorly when the face is not frontal. Dlib HOG does not allow detecting faces with large occlusion. In real application, we did not get frames where the person was not detected by Dlib HOG.

![Figure 5. Predictors processing speed.](image1)

![Figure 6. Predictors accuracy.](image2)

![Figure 7. Comparison predictor’s accuracy.](image3)

3. Fatigue features detection

3.1. Eye closure detection

The whole aspect ratio approach is based on the use of the ear threshold value. This value averaged among all people and for some specific individuals with a different eye structure it can be incorrect.

To detect the state of openness of the eyes, we used the machine learning technique. One of the machine learning algorithms is a convolutional neural network. The neural network was built and trained using the Keras library. The neural network architecture is shown in Figure 8.

For training and testing of the neural network was used dataset [13], which contains 2,423 images, in size 24 × 24.

This architecture is optimal on the processor and will work in real time. Also, this architecture allows detecting the state of the eye with high accuracy. The neural network was trained and predicts with 0.96 accuracies.

3.2. Yawning detection

To detect the yawn, we will use an adjacent algorithm with a neural network. Let’s transform data from the dataset [18] for a neural network. The original data is stored as video recordings of 1 minute with an AVI file extension and with a frame rate of 30 fps. In the video, the driver talks and periodically yawns. On average, a person yawns 2-3 times on a video. We calculate the mar value for
each frame and transform the signal using a median filter. The original signal and the signal processed by the median filter are shown in Figure 9.

![Image of neural network architecture](image)

**Figure 8.** Neural network architecture.

![Image of yawn signals](image)

**Figure 9.** Yawn signals.

Next, we transform into a signal depending on four classes, which is shown in Figure 10. We can see 4 classes of values from 0 to 3. Below in Figure 11 is a classification of these values. Images were cut using the face alignment library.

The neural network with the architecture shown in Figure 12 receives 54,381 images. The accuracy of the trained neural network is 0.9389.

**4. Experiment researches**

Detection fatigue featured implement in close relation with Dlib face landmark algorithm. So, these methods don’t work with face occlusion, lack of light, sun glare, glasses and other obstacles to detect via face landmark algorithm. The yawn detection results are shown in Figure 13.

Dlib face landmarks a lot of error points with obstacles. Figure 14 shows error predictions.
Figure 11. Yawn classes.

Figure 12. Neural network architecture.

Figure 13. Yawn.
Figure 14. Error predictions.

The processing speed of algorithms is shown in Table 2.

|                | Left eye state | Right eye state | Yawn  |
|----------------|----------------|-----------------|-------|
| Intel Core i7 3632 QM | 0.00124        | 0.00119         | 0.00194 |

Therefore, when using only CPU, FPS is 228, which is suitable for real-time processing.

5. Conclusion
The developed algorithms work in real time with fps 228 on the CPU Intel Core i7 3632 QM but have limitations. The imperfection of the algorithm for detecting face landmark gives a lot of false predictions. It is possible to use a more accurate library, such as face-alignment, but these methods don’t work in real time.

In future research, the described method will be implemented on mobile processors like Nvidia Jetson Nano.

6. References
[1] Shustanov A, Yakimov P 2017 CNN design for real-time traffic sign recognition Procedia Engineering 201 718-725
[2] Shakhuro V I, Konushin A S 2018 Image synthesis with neural networks for traffic sign classification Computer Optics 42(1) 105-112 DOI: 10.18287/2412-6179-2018-42-1-105-112
[3] Driver fatigue and road accidents 2011 The Royal Society for the Prevention of Accidents p 4
[4] Berka C, Levendowski D J, Lumicao M N, Yau A, Davis G, Zivkovic V T, Olmstead R E, Tremoulet P D and Craven P L 2007 EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks Aviat. Space Environ. Med. 78 231-244
[5] De Rosario H, Solaz J, Rodriguez N and Bergasa L 2010 Controlled inducement and measurement of drowsiness in a driving simulator Intell. Transp. Syst. IET 4 280-288
[6] Akerstedt T, Gillberg M 1990 Subjective and objective sleepiness in the active individual Int. J. Neurosci. 52 29-37
[7] Wakita T, Ozawa K, Miyajima C, Igarashi K, Itou K, Takeda K and Itakura F 2005 Driver Identification Using Driving Behavior Signals Proceedings of the IEEE Intelligent Transportation Systems 396-401
[8] Dinges D 1998 PERCLOS: A Valid Psychophysiology Measure of Alertness as Assessed by Psychomotor Vigilance Technical Report; Federal Highway Administration (Washington, DC, USA)
[9] Omideyeganeh M, Shirmohammadi S, Abtahi S, Khurshid A, Farhan M, Scharcanski J, Hariri B, Laroche D and Martel L 2016 Yawning detection using embedded smart cameras IEEE Transactions on Instrumentation and Measurement 65(3) 570-582
[10] Vicente F, Huang Z, Xiong X, De la Torre F, Zhang W and Levi D 2015 Driver gaze tracking and eyes off the road detection system IEEE Transactions on Intelligent Transportation Systems 16(4) 2014-2027
Acknowledgements

This work was partly funded by the Russian Foundation for Basic Research – Project # 17-29-03112 ofi_m and the Russian Federation Ministry of Science and Higher Education within a state contract with the "Crystallography and Photonics” Research Center of the RAS under agreement 007-ГЗ/43363/26.