Fatigue Driving Detection Based on Machine Learning and Image Processing Technology

Siquan Hu\textsuperscript{1,2} and Zhenye Lin\textsuperscript{1,3}

\textsuperscript{1}School of Computer and Communication Engineering, University of Science and Technology Beijing, 100083, China
\textsuperscript{2}Email: husiquan@ustb.edu.cn
\textsuperscript{3}Email: linzhenye117@163.com

Abstract. Fatigue driving is a major cause of traffic accidents. If fatigue driving can be detected and stopped in time, perhaps traffic accidents will be reduced. This paper proposes a fatigue driving detection algorithm using machine learning and image processing technology. The algorithm is designed to run on the vehicle’s smart terminal. In the algorithm, terminal has to analyze the status of driver’s eyes, nose and mouth using machine learning and image processing technology. When catch typical behaviors such as blink, yawn and nod, terminal will record the time and calculate the frequencies of these behaviors. If one of the frequencies is greater than the threshold, terminal will trigger the alert and take measures to make driver no longer fatigue driving. The algorithm works fine in the experiment. The experimental accuracy of the algorithm is 94%.

1. Introduction
Fatigue driving can be dangerous [1]. There are many people study how to detect and stop fatigue driving. Some people use the brain wave to study how to detect the fatigue state. The brain wave will change when the driver feels tired [2]. But the brain wave detect equipment is expensive and not so convenient. Some people want to use the image of the driver to detect the fatigue state. There will be some changes on the face when people feel tired [3]. So with good analysis, a fatigue state can be detected. There many ways to detect the differences on the face. Someone focus on the geometric characteristics of pupils [4]. Some focus on the behaviors on the face [5]. People always do some typical behaviors such as blink, yawn and nod when they are wasted [6]. Capturing these behaviors can also detect fatigue state. To analyze the image of the drivers face, the machine learning and image processing technology will be used. There are also some people using the track of the vehicle to detect fatigue driving [7].

In this paper, we focus on monitoring the typical facial behaviors when drivers are tired. And use the frequencies of typical behaviors to decide whether the driver is tired or not. And the machine learning, image processing and statistics technology will be used.

2. Capture of Driver’s Facial Features
To capture the driver’s fatigue status, a camera will be used to monitor the driver’s face. The Image of driver’s face couldn’t be used to analyze the driver’s fatigue status directly. It has to be processed by the machine learning algorithm. The machine learning algorithm we used comes from Dlib which is an open source library. Through the machine learning algorithm, the details of facial features can be captured. The facial features include the positions and outlines of eyes, nose and mouth. With these facial features, we can catch typical behaviors such as blink, yawn and nod and continue the fatigue
driving detection. As shown in figure 1, the result of capturing the driver's facial features contains 68 points. Each point has two coordinates, which indicate the location of the point in the picture.

Figure 1. The result of capturing facial features

3. Detection of Fatigue State
Fatigue state can be shown in driver’s face. When people is tired, they may blink, yawn or nod more frequently. Based on this principle, we proposed a fatigue state detection algorithm. The algorithm can be divided into four parts, parameter calibration, blink detection, yawn detection and nod detection. The parameter calibration part goes first and aims to calibrate some parameters that will be used in the other three parts.

3.1. Blink Detection
Blinking is a typical behavior people will do when they feel sleepy. Monitoring the blinking frequency can be a way to detect fatigue state. If the frequency of blinking is over a certain threshold, we believe a fatigue state is detected.

Figure 2. The distribution of points in the eyes and nose

Figure 3. The status of Eyes and nose during blinking
A blinking process can be divided into two parts, eyes closing process and eyes opening process. So after detected both eye closing progress and eye opening progress, we think a blinking process is detected. The difference between eyes closing and eyes opening is the distance between the upper and lower eyelids. When people’s eyes are opened the distances is bigger than that when people’s eyes are closed. Although the distances can tell the difference between opening or closing eyes, the distances also changed when the angle between people’s head and camera view has changed. So we propose a way to solve the problem. As shown in figure 2, we take the ratio of the distance between the upper and lower eyelids and the length of the nose as a parameter $R_{\text{eye-nose}}$. The calculation of the left eye’s parameter is shown in equation (1) and the calculation of the right eye’s parameter is shown in equation (2). During the experiment, this parameter turn out to be stable when the angle between people’s head and camera view changes.

$$R_{L\text{-eye-nose}} = \frac{|y_{44} - y_{40}| + |y_{44} - y_{36}|}{2|y_{27} - y_{33}|}$$

(1)

$$R_{R\text{-eye-nose}} = \frac{|y_{37} - y_{44}| + |y_{38} - y_{40}|}{2|y_{27} - y_{33}|}$$

(2)

Although we can get rid of the angle problem by using both $R_{R\text{-eye-nose}}$ and $R_{L\text{-eye-nose}}$. We still have to deal with the problem that the parameters $R_{R\text{-eye-nose}}$ and $R_{L\text{-eye-nose}}$ are different for different person. In order to deal with the problem, we decided to calibrate these two parameters during the beginning stage of the algorithm. As shown in equation (3) and equation (4), we record the first 50 values of both parameters and get their own average value as calibrated value. We just need to take the ratio of the value and its own calibrated value as a new parameter to solve the problem. For left eye, we take the new parameter as $R_{L\text{-eye-cali-rate}}$ and the calculation is shown in equation (5). For right eye, we take the new parameter as $R_{R\text{-eye-cali-rate}}$ and the calculation is shown in equation (6). When both $R_{L\text{-eye-cali-rate}}$ and $R_{R\text{-eye-cali-rate}}$ are smaller than a certain threshold, we believe that an eye closing process is captured. After that, if both $R_{L\text{-eye-cali-rate}}$ and $R_{R\text{-eye-cali-rate}}$ are bigger than a certain threshold, we think an eye opening process is captured. With both eye closing process and eye opening process were captured, a blinking process is completed. We can record the time when it completed. If more than one time was captured, we can use the last two time to calculate the frequency and compare the frequency with a certain threshold to decide whether to trigger the alert or not. Also the algorithm will trigger the alert if the driver closes his eye for a long time.

$$R_{\text{cali-L\text{-eye-nose}}} = \frac{\sum_{1}^{50} R_{L\text{-eye-nose}}}{50}$$

(3)

$$R_{\text{cali-R\text{-eye-nose}}} = \frac{\sum_{1}^{50} R_{R\text{-eye-nose}}}{50}$$

(4)

$$R_{L\text{-eye-cali-rate}} = \frac{R_{L\text{-eye-nose}}}{R_{\text{cali-L\text{-eye-nose}}}}$$

(5)

$$R_{R\text{-eye-cali-rate}} = \frac{R_{R\text{-eye-nose}}}{R_{\text{cali-R\text{-eye-nose}}}}$$

(6)
3.2. Yawn Detection
The same as the blinking detection, if people yawn frequently in a short time, we believe they are tired.

![Image of mouth and nose distribution with annotations.]

**Figure 4.** The distribution of points in the mouth and nose

Yawn can also be divided into two parts, mouth opening process and mouth closing process. As shown in figure 4, we take the ratio of the distance between the upper and lower lips and the length of the nose as a parameter $R_{\text{mouth-nose}}$. The calculation of $R_{\text{mouth-nose}}$ is shown in equation (7). When $R_{\text{mouth-nose}}$ is over a certain threshold, it seems that the mouth is opened. It takes about 6 second to complete a yawn behavior. That means if someone open his mouth over 6 second, it seems that this person is yawning. After that, if the mouth is closed, we believe a yawn process is completed. If the algorithm capture a yawn behavior is completed, it will record the time. If there are more than one time recorded, algorithm will use the last two times to get the yawn frequency and compare the frequency to a certain threshold. If the frequency is over the threshold, the alert will be triggered.

$$R_{\text{mouth-nose}} = \frac{|y_{62} - y_{66}|}{|y_{27} - y_{33}|}$$

(7)

3.3. Nod Detection
The same as the blinking detection and the yawning detection, if the frequency of nodding reaches a certain level, we believe a fatigue state is detected.
Figure 6. The normal status of mouth and nose

Figure 7. The status of mouth and nose during nodding.

Nodding detection is different from blinking detection and yawning detection. Because when someone nods, it will make a difference to the angle between the head and the camera. In order to catch the change, we take the ratio of upside nose length and downside nose length as a parameter $R_{\text{nose-down-up}}$. As shown in figure 6 and figure 7, the red line represents upside nose length and the yellow line represents downside nose length. During nodding head, $R_{\text{nose-down-up}}$ will become smaller. That is away to quantify the angle between the head and the camera. The calculation of $R_{\text{nose-down-up}}$ is shown in equation (8).

$$R_{\text{nose-down-up}} = \frac{|y_{22} - y_{30}|}{|y_{30} - y_{33}|}$$

(8)

$$R_{\text{cali-nose-down-up}} = \frac{\sum_{i=1}^{50} R_{\text{nose-down-up}}}{50}$$

(9)

$$R_{\text{nose-cal rate}} = \frac{R_{\text{nose-down-up}}}{R_{\text{cali-nose-down-up}}}$$

(10)

In order to get rid of the problem that different person may has different $R_{\text{nose-down-up}}$, a calibration process is also required for nodding detection. As shown in equation (9), we record the first 50 values of $R_{\text{nose-down-up}}$ and get average value as calibrated value $R_{\text{cali-nose-down-up}}$. As shown in equation (10) we take the ratio of $R_{\text{nose-down-up}}$ and $R_{\text{cali-nose-down-up}}$ as a new parameter $R_{\text{nose-cal rate}}$. If $R_{\text{nose-cal rate}}$ is over a certain threshold, it means a nod behaviour is captured and the time will be recorded. When there are more than one time been recorded, the nodding frequency will be calculated by using the last two recorded time. If the frequency is bigger than a certain threshold, the alert will be triggered.
4. Experiments
The fatigue state detection algorithm we proposed contains a lot of parameters. The algorithm won’t work well if these parameters are not suitable. In order to optimize the algorithm, we have to do lots of experiments to get the best values of the parameters. We tried many groups of parameters. Each group of parameters has been used to do experiment for over 50 times. At the end, we calculated the accuracy of each group. Table 1 shows some groups of parameters in the experiments. The parameter $T_{\text{eye-rate}}$ represents the threshold of $R_{\text{L-eye-cali-rate}}$ and $R_{\text{R-eye-cali-rate}}$. The parameter $T_{\text{f-blink}}$ represents the threshold of blinking frequency. The parameter $T_{\text{c-eye-time}}$ represents the threshold of the time eye has been closed. The parameter $T_{\text{mouth-rate}}$ represents the threshold of $R_{\text{mouth-cali-rate}}$. The parameter $T_{\text{f-yawn}}$ represents the threshold of yawning frequency. The parameter $T_{\text{head-rate}}$ represents the threshold of $R_{\text{nose-cali-rate}}$. The parameter $T_{\text{f-nod}}$ represents the threshold of nodding frequency.

![Table 1. Some groups of parameters in the experiments.](image)

| $T_{\text{eye-rate}}$ | $T_{\text{f-blink}}$ | $T_{\text{c-eye-time}}$ | $T_{\text{mouth-rate}}$ | $T_{\text{a-mouth-time}}$ | $T_{\text{f-yawn}}$ | $T_{\text{head-rate}}$ | $T_{\text{f-nod}}$ | Accuracy |
|-----------------------|----------------------|------------------------|-------------------------|--------------------------|---------------------|---------------------|---------------------|----------|
| 0.3                   | 0.4                  | 2                      | 0.5                     | 1                        | 0.1                 | 0.2                 | 0.1                 | 66.7%    |
| 0.25                  | 0.3                  | 1                      | 0.6                     | 2                        | 0.2                 | 0.25                | 0.2                 | 45%      |
| 0.3                   | 0.4                  | 2                      | 0.5                     | 1                        | 0.1                 | 0.3                 | 0.1                 | 72.4%    |
| 0.3                   | 0.4                  | 3                      | 0.5                     | 1                        | 0.33                | 0.3                 | 0.1                 | 78.5%    |
| 0.5                   | 0.4                  | 3                      | 0.5                     | 1                        | 0.33                | 0.3                 | 0.1                 | 84%      |
| 0.5                   | 0.67                 | 3                      | 0.7                     | 6                        | 0.33                | 0.3                 | 0.1                 | 94%      |
| 0.5                   | 0.8                  | 4                      | 0.7                     | 5                        | 0.33                | 0.3                 | 0.1                 | 90%      |

As shown in table 1, when the group of parameters are 0.5, 0.67, 3, 0.7, 6, 0.33, 0.3 and 0.1, the accuracy will reach 94%. And this group of parameters has been used in experiment for 100 times. During the 100 time experiments, 94 times was right.

5. Conclusion
In this paper, a fatigue driving detection algorithm based on machine learning and image processing technology is proposed. The algorithm focuses on the eyes, nose and head’s state and analyses the frequencies of blinking, yawning and nodding. If the algorithm found out the driver is tired, it will trigger the alert and take measures to stop the fatigue driving. The accuracy of the algorithm is up to 94%. It seems the algorithm is reliable.

6. References
[1] Cao G Z, Peng H and Tan W 2016 Electronic Design Engineering 24 (12) pp 165 – 171
[2] Huang X, Wu Y J, Wen Z X, Liao M H 2016 Journal of Foshan University 34 (2) pp 54 - 59
[3] Sun W, Zhang W G, Zhang X R, Chen G and Lv C X 2009 Modern Transportation Technology 6(1) pp 73 - 76
[4] Xu S F, Li K T, Di H W and Liu Z J 2009 Optical Technique 35(1) pp 148 - 151
[5] Pierre P, Patricia S, Nicholas M, Jacques T, André C, Christian G and Bernard B 2005 Accid Anal Prev 37(3) pp 473 - 478
[6] Gerald M and Paula A D 2002 Quarterly Journal of Experimental Psychology A Human Experimental Psychology 55 (2) p 659
[7] Lu Y and Gao C 2013 Applied Mechanics & Materials 333-335 pp 1060 -1064