

## Research Article

# Driver Fatigue Detection Based on Facial Key Points and LSTM

Long Chen,<sup>1</sup> Guojiang Xin ,<sup>2</sup> Yuling Liu,<sup>1</sup> and Junwei Huang<sup>3</sup>

<sup>1</sup>College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

<sup>2</sup>School of Informatics, Hunan University of Chinese Medicine, Changsha 410208, China

<sup>3</sup>HERE North American LLC, Burlington 01803, VT, USA

Correspondence should be addressed to Guojiang Xin; lovesin\_guojiang@126.com

Received 14 April 2021; Accepted 5 June 2021; Published 14 June 2021

Academic Editor: Beijing Chen

Copyright © 2021 Long Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, fatigue driving has been a serious threat to the traffic safety, which makes the research of fatigue detection a hotspot field. Research on fatigue recognition has a great significance to improve the traffic safety. However, the existing fatigue detection methods still have room for improvement in detection accuracy and efficiency. In order to detect whether the driver has fatigue driving, this paper proposes a fatigue state recognition algorithm. The method first uses MTCNN (multitask convolutional neural network) to detect human face, and then DLIB (an open-source software library) is used to locate facial key points to extract the fatigue feature vector of each frame. The fatigue feature vectors of multiple frames are spliced into a temporal feature sequence and sent to the LSTM (long short-term memory) network to obtain a final fatigue feature value. Experiments show that compared with other methods, the fatigue state recognition algorithm proposed in this paper has achieved better results in accuracy. The average accuracy of the proposed method in detecting key points of the face is as high as 93%, and the running time is less than half of the ordinary DLIB method.

## 1. Introduction

Automobiles have become the most popular tools of transportation. As the frequency of automobile use continues to increase, traffic accidents are also increasing. In many traffic accidents, fatigue driving is one of the main reasons. Fatigue driving has caused many major traffic accidents, which caused huge losses to people's lives and properties.

Relevant Chinese traffic laws stipulate that driving for 4 hours without a break is fatigue driving. In a survey in the United States, more than half of the drivers admitted that they had fatigue driven [1]. When a driver is fatigued, his concentration, judgment ability, and reaction sensitivity are reduced [2]. These factors will make traffic accidents more likely to occur. Long-distance driving is the most prone to fatigue driving and often causes the safety accidents. Therefore, fatigue driving detection technology has become a research hotspot in the field of the traffic safety.

At present, fatigue detection methods are divided into the following categories: methods based on the physiological

information, methods based on the vehicle status, methods based on the computer vision, and methods based on the information fusion models [3].

Physiological information mainly refers to the driver's breathing rate, pulse, blood pressure, and heart rate. These parameters can quickly and accurately reflect a person's physical and mental state. The detection methods based on the physiological information not only have strong real-time performance but also have high accuracy [4]. However, the driver needs to wear related equipment during the detection process, which will affect the normal operation of the driver, so that the practical applications are limited. The status of the vehicle refers to the vehicle's trajectory, steering wheel manipulation, and lane deviation. These detection methods indirectly analyze the driver fatigue state by analyzing vehicle information [5]. The main disadvantage of these methods is low accuracy. The detection methods based on the computer vision can quickly and accurately detect the driver fatigue state by capturing and analyzing the driver's face video in real-time. These methods do not need the driver to wear the related equipment and have good

performance in the terms of detection rate and reliability. The main difficulty of these methods is face image processing. Information fusion methods are the comprehensive use of the physiological information, vehicle information, and computer vision algorithms to detect the driver's fatigue state. The advantage is that it can improve the accuracy of the detection, but the disadvantage is that it is difficult to establish an information fusion model and obtain various information.

The main contribution of this paper is to propose a new, high-precision, real-time fatigue detection method based on the computer vision. We combine MTCNN and DLIB together, which allows us to extract the facial features fast and accurately and then combine the facial features of multiple frames to make our fatigue judgment results more accurate. This method first divides the video into image frames and cuts out the facial area through MTCNN and then uses the DLIB library to extract the fatigue features of the eye and mouth for each image frame. Finally, multiple frames of the fatigue feature are input into the recognition network based on LSTM to obtain fatigue judgment results.

## 2. Related Work

In recent years, many scholars and institutions have conducted a lot of researches on the fatigue driving detection based on the computer vision.

D'orazio et al. proposed an algorithm by eye detection. The algorithm used iris geometric information to determine the entire image [6]. Sun et al. studied the relationship between the closed eyes and the fatigue, and they used PERCLOS to detect the driver's fatigue and obtained better test results [7]. Ma et al. designed a system to detect the fatigue driving state at night. They used a deep framework based on ConNN and verified it on their own dataset [8]. Zhang et al. created a model to solve the influence of the sunglasses on the fatigue detection, which used the IRF dataset [9]. Gupta et al. observed the facial features of the driver through a camera and classified the fatigue levels through principal component analysis and support vector machine (SVM) classifier [10]. Junaedi and Akbar calculated PERCLOS by detecting the eyes and used it to judge the fatigue. They used the YawDD dataset [11]. Savas and Becerikli tried to use the SVM algorithm to detect driver fatigue. In their study, they used the number of yawns, the internal area of the mouth, and the number of blinks to determine the driver fatigue level on the dataset [12]. Amodio et al. designed a driver state detection system based on pupil light reflection. They used the pupil size contour and SVM classifier to judge the driver's state [13]. Li et al. designed a human behavior recognition classification system based on ConNN. They proposed a face recognition algorithm based on LBP-EHMM [14]. Liu et al. proposed a driver fatigue detection algorithm using a two-stream network model with multiple facial features. They applied gamma correction to enhance the image contrast to obtain better results [15]. Savaş and Becerikli proposed a multitask convective neural network model to detect driver drowsiness/fatigue. The features of the eyes and mouth were used to model the behavior of the driver. The changes in these characteristics were used to monitor the

driver's fatigue [16]. Liu et al. proposed a fatigue detection algorithm based on the deep learning facial expression analysis. They trained a facial key point detection model through multiple local binary patterns and AdaBoost classifiers [17]. Ed-Doughmi et al. proposed a method to analyze and predict driver drowsiness by applying a recurrent neural network on the driver's face in sequence frames. They used a 3D convolutional network based on a repetitive neural network architecture of a multilayer model to detect the driver's drowsiness [18].

Yawning and frequent blinking are the most obvious signs of driver fatigue. Therefore, the first task is to determine the human eyes' state and mouth's state. There are generally two ways to detect the eyes and mouth. One is to directly detect the positions of the eyes and mouth. The other is to firstly find the facial area and then detect the positions of the eyes and mouth. The human face has more information, and the features are more stable than the human eyes. Cutting out the face area can reduce the test range of the eye position and avoid the interference of the background.

The existing face detection algorithms can be divided into two categories: one is a multilevel detection algorithm based on the proposed region. The other is the target detection algorithm based on anchor frame [19]. The representative algorithms of the former are Faster-RCNN [20] and MTCNN [21]. The representative algorithms of the latter are S3FD [22] and SSH [23]. Compared with traditional learning methods, detection methods based on deep learning do not require manual feature extraction. With the support of a large amount of training data, the detection performance will be greatly improved.

Fatigue driving is a continuous behavior. Therefore, the fatigue detection method based on continuous multiple frames will definitely be better than the single frame method. Donahue proposed the LRCN framework [24], which can process continuous multiple frames of relevant information to perform behavior recognition and classification.

## 3. Methodology

The framework proposed in this paper is shown in Figure 1. We will introduce the implementation details of each part in detail.

*3.1. Face Detection.* In this task, we use MTCNN for face detection, which is based on deep learning and can quickly and efficiently complete face detection and face alignment [25]. MTCNN can detect five key points of the face: left and right corners of the mouth, nose, and left and right eyes. However, the five key points are not enough to extract facial fatigue information, so we use MTCNN just for face detection. MTCNN includes three subnets: proposal network (P-Net), refine network (R-Net), and output network (O-Net). MTCNN is composed of cascades of them [26].

*3.1.1. P-Net.* The main task of this network is to obtain the bounding box and regression vector of the candidate window. After the candidate window is calibrated, nonmaximum

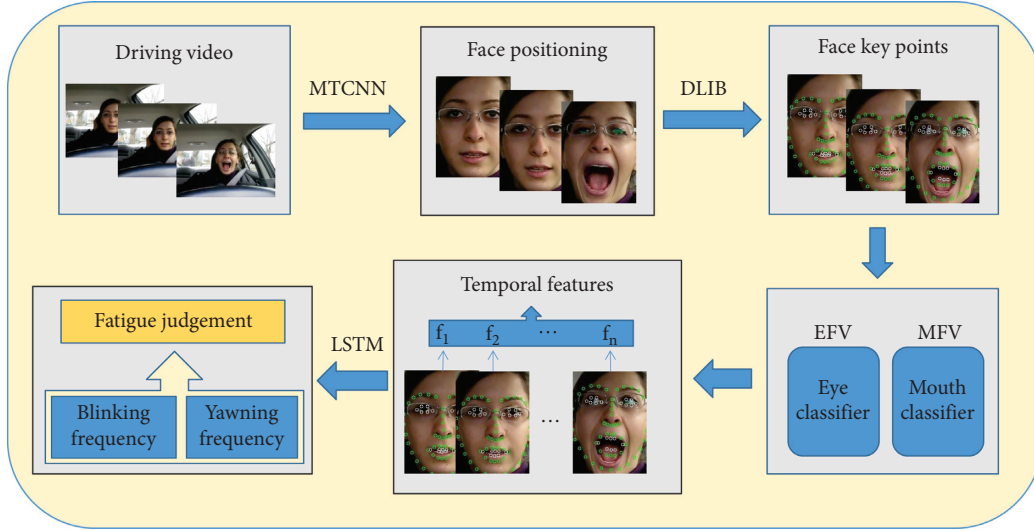


FIGURE 1: Framework of fatigue detection.

suppression is used to eliminate highly overlapping windows. P-Net is a regional proposal network for face regions. The network uses a face classifier to determine whether there is a face in the area and uses border regression and a locator of facial key points to make a preliminary proposal of the face area. This part will output many candidate windows and use these windows as the input of R-Net.

**3.1.2. R-Net.** The main task of this network is to eliminate false samples and continue to obtain bounding boxes and regression vectors. Unlike the previous network, R-Net has a more complete connection layer. When the test sample passes the P-Net layer, many candidate windows are gotten. The network will filter out a large number of wrong candidate windows. Finally, bounding-box regression and non-maximum suppression (NMS) were performed on the selected candidate boxes to further optimize the prediction results.

**3.1.3. O-Net.** This network is more complicated than the first two networks. O-Net has a 256 fully connected layer. After further filtering the candidate window of R-Net, this layer of network will also calculate the position of the facial feature points. In addition, this operation can eliminate the influence of some obstructions, such as sunglasses, hats, and ordinary glasses.

**3.2. Facial Key Point Detection.** In this phase of the task, we use DLIB to label the key points of the face. DLIB can be regarded as a machine learning toolbox, which is designed to solve the extraction of key points of human faces. DLIB has received widespread attention once it is launched, and it can be applied to mobile devices or large-scale high-performance computing environments. Like many open-source libraries, DLIB can be used by researchers for free. We choose the DLIB library because it can provide training and extraction tools for 68 facial key points. We can use it to obtain 68 facial key points and use these key points to extract fatigue features [27].

**3.2.1. Closed-Eye Detection.** Obviously, when people's eyes are open, the distance between the upper and lower feature points of the eyes will be relatively large. When the eyes are closed, the distance becomes smaller. The EYE value is calculated by using the distance of the eye feature points. Among the 68 feature points on the face, the eye points correspond to 37–42 and 43–48, respectively. Figures 2 and 3, respectively, show the state of open and closed eyes.

The calculation formula of EYE values is as follows:

$$\text{EYE} = \frac{\|P_{38} - P_{42}\| + \|P_{39} - P_{41}\|}{2\|P_{37} - P_{40}\|} \quad (1)$$

The numerator represents the Euclidean distance between the vertical feature points of the eyes, and the denominator is the Euclidean distance between the horizontal feature points of the eyes. The Euclidean distance between two points is calculated as follows:

$$\text{Dis}(a, b) = \sqrt{(P_a \cdot x - P_b \cdot x)^2 + (P_a \cdot y - P_b \cdot y)^2}, \quad (2)$$

where  $P_a \cdot x$  and  $P_a \cdot y$  represent the coordinates  $x$  and  $y$  of point  $a$ , respectively, and the horizontal and vertical Euclidean distances of the eye can be expressed as follows:

$$\text{Eye}_h = \text{Dis}(P_{37}, P_{40}), \quad (3)$$

$$\text{Eye}_v = \text{Mean}(\text{Dis}(P_{38}, P_{42}), \text{Dis}(P_{39}, P_{41})), \quad (4)$$

where  $\text{Mean}(A, B)$  means the average of  $A$  and  $B$ , and then the aspect ratio of the eye can be expressed as follows:

$$\text{EYE}_{\text{left}} = \frac{\text{Eye}_v}{\text{Eye}_h} \quad (5)$$

Since the value calculation process of the left and right eyes is the same, the calculation process of the right eye will not be repeated. The eye feature vector (EFV) is composed of  $\text{EYE}_{\text{left}}$  and  $\text{EYE}_{\text{right}}$ .

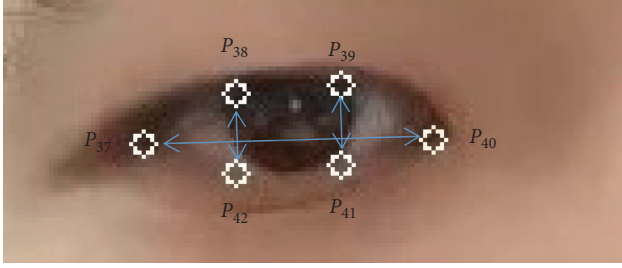


FIGURE 2: Eye open state diagram.

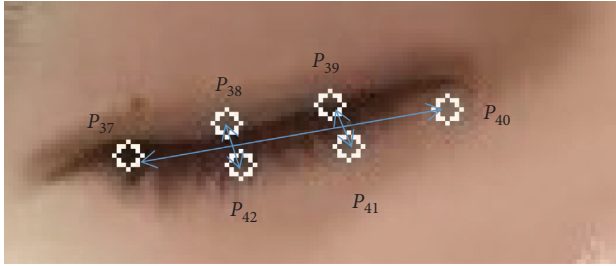


FIGURE 3: Eye-closed state diagram.

**3.2.2. Yawn Detection.** Yawn detection is similar to closed eye detection. The key points of the mouth are 61–68. These points make up the key points of our inner lips. Some scholars use the key points of the outer lips. However, due to the individual lip differences, the calculated value will not be accurate enough. The MOUTH value is calculated by using the distance of the mouth feature points, which can judge the state of the mouth. The mouth feature vector (MFV) only consists of MOUTH. Figures 4 and 5 show the open and closed states of the mouth, respectively.

The horizontal and vertical Euclidean distances of the mouth can be expressed as follows:

$$\text{Mouth}_v = \text{Mean}(\text{Dis}(P_{62}, P_{68}), \text{Dis}(P_{63}, P_{67}), \text{Dis}(P_{64}, P_{66})), \quad (6)$$

$$\text{Mouth}_h = \text{Dis}(P_{61}, P_{65}), \quad (7)$$

where  $\text{Mean}(A, B, C)$  means the average of  $A$ ,  $B$ , and  $C$ .

Then, the aspect ratio of the mouth can be expressed as follows:

$$\text{MOUTH} = \frac{\text{Mouth}_v}{\text{Mouth}_h}. \quad (8)$$

**3.3. Fatigue Recognition Network.** Many existing fatigue identification methods only use a single fatigue feature, which will lead to many misjudgments. Assuming that only the mouth information is used to determine whether you are tired, it is likely to misjudge your speech as fatigue [28]. Therefore, the fatigue detection results obtained by analyzing one single frame are not accurate. Inspired by LRCN [24], a two-stage fatigue identification method is designed in this

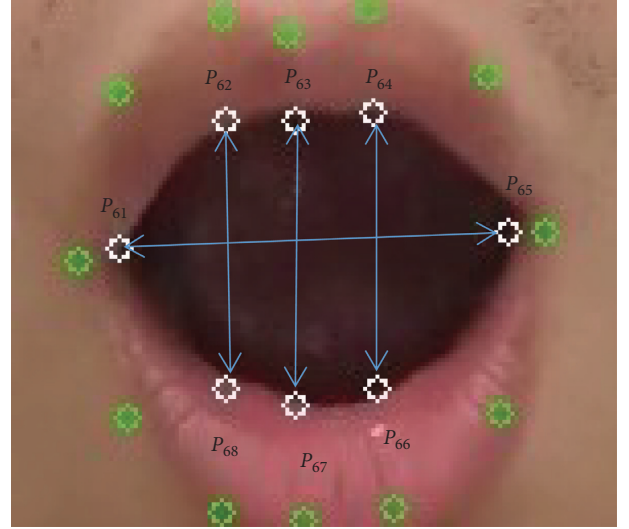


FIGURE 4: Mouth open-state diagram.

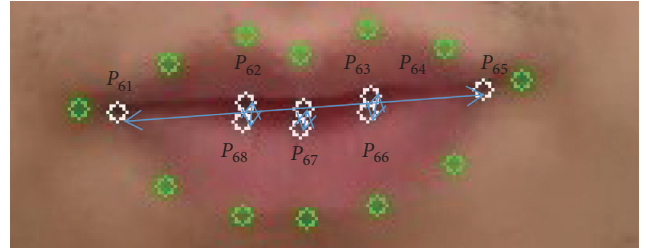


FIGURE 5: Mouth closed-state diagram.

paper. The first stage is splitting the input video into frames of pictures. The fatigue vector of a single frame is extracted through MTCNN and DLIB, and the information of multiple consecutive frames is combined to form a temporal feature vector. The second stage is as follows: these fatigue feature sequences are input into the LSTM-based network to identify the fatigue state.

**3.3.1. Temporal Fatigue Characteristic Sequence.** The feature extraction task needs to extract the eyes and mouth state values of each frame. Therefore, we set the single frame feature vector length to 3. The fatigue feature vector of a single frame image is as follows:

$$f_n = \{\text{Value}_{\text{leye}}, \text{Value}_{\text{reye}}, \text{Value}_{\text{mouth}}\}, \quad (9)$$

where  $\text{Value}_{\text{leye}}$  and  $\text{Value}_{\text{reye}}$  represent the state of the left eye and the right eye and  $\text{Value}_{\text{mouth}}$  represents the state of the mouth.

The feature vector of each frame is  $1 \times 3$ . So, we splice the feature vectors of multiple frames, and a temporal feature sequence of  $n \times 3$  will be formed. The vector length is 3, and the number of spliced frames is  $n$ . The splicing process is shown in Figure 6.



As shown in Figure 6, the length of the time window is a key parameter to construct the temporal fatigue characteristic sequence. If the length is too short, the obtained sequence may not be able to completely cover the fatigue state, and the excessively long time window will cause the sequence to contain too much redundant information. Another key parameter is the number of the skipped frames. Since the information of adjacent frames will be almost the same, it is not necessary to extract the information of each frame, which will cause a lot of waste of calculation and greatly reduce the efficiency. We split each video sample at a rate of two frames per second. Since the fatigue process usually does not exceed three seconds, we chose a time window length of 6 and skipped frames number of 2.

**3.3.2. Fatigue Recognition Network Based on LSTM.** LSTM is carefully designed to avoid the problem of long dependencies. Remembering long historical information is actually a default behavior. LSTM works very well on various problems and is now widely used in pattern recognition. Based on this idea, a fatigue identification network based on LSTM is applied in this paper. Its structure is shown in Figure 7.

As shown in Figure 7, the input of the LSTM network is a sequence of time features. The time feature sequence is composed of six single frame feature vectors. Therefore, the length of LSTM is also 6. LSTM will return a probability value, which represents the probability of driver fatigue in the current time window. When the probability value is more than 0.5 or equal to 0.5, we set this value to 1 and indicate that the driver is in a state of fatigue during the current period. When the probability value is less than 0.5, we set this value to 0, which indicates that the driver is awake during the current period. As long as a period is judged to be fatigued, we will treat the video as a fatigue sample.

## 4. Experiment

**4.1. Dataset.** In the experimental part, we selected the YawDD dataset and self-built dataset to verify the performance of the method.

**4.1.1. YawDD Video Dataset.** The dataset is collected by Abtahi et al. [29], which was captured in a static environment. The collectors gathered a large number of volunteers. The volunteer group was composed of drivers of different skin colors, sexes, and ages. They did different actions according to the instructions as normal driving, talking, and yawning. Each volunteer was shot multiple videos. When the driver wears pure black sunglasses, the human eye cannot recognize the eye condition of driver. Therefore, we selected 100 videos where volunteers were not wearing pure black sunglasses, including 50 men and 50 women for testing. A part of the dataset is shown in Figure 8.

**4.1.2. Self-Built Dataset.** In the YawDD dataset, some drivers in the video do not yawn naturally, but just open their mouths to make a yawning action. In order to capture the most natural fatigue state as much as possible, our fatigue video samples are all taken after the volunteers get off work. After working for a long time, most people are more prone to fatigue. We cannot guarantee that every sample captures the natural yawning action, but we filmed the behavior that fits the most natural fatigue. Our algorithm was tested on behaviors often associated with fatigue versus actual fatigue. Second, the proportion of yellow people in the YawDD dataset is low, mostly whites and Indians. Adding a self-built dataset can help reduce the difference in experimental results caused by races of different skin colors.

Self-built dataset was collected by our experimental team. We gathered 10 volunteers and each was shot two videos: one is a normal video, and the other is a fatigue video; they included closing eyes, talking, laughing, and yawning. These videos had slightly different face orientations, mouth shapes, and whether they wear glasses, and they were collected under different lighting conditions. Part of the dataset is shown in Figure 9.

**4.2. Experimental Results and Analysis.** The platform of this experiment is Windows 10, the processor is Inter(R) Core™ i7-9700k, the main frequency of the CPU is 3.6 GHZ, and the memory is 8 GB. The programming language is Python. In the experiment, we split the video dataset into images and use MTCNN to detect and crop the face images. After cropping the face image, the DLIB library is used to mark the key points of the face to calculate the state value of the eye and mouth. By calculating the aspect ratio of the eyes and the mouth, we can perform closed eye detection and yawn detection. In order to verify the performance of the proposed algorithm, we compare our algorithm with the key point detection algorithms proposed in recent years. The experimental results are shown in Tables 1 and 2.

Tables 1 and 2, respectively, show the detection accuracy of our model and other methods in the YawDD dataset and the self-built dataset. It can be seen that our model is significantly better than other algorithms. The method proposed in this paper has a higher eye-mouth marking rate than other methods. Compared with the Viola-Jones algorithm, our method has significantly better results in the detection of faces, eyes, and mouths. Second, the detection results on the YawDD dataset are slightly lower than the self-built dataset. This may be due to the small number of videos in our self-built dataset. There is not much difference in actual detection results. Next, we compare the detection time between different methods.

Tables 3 and 4, respectively, show the detection time of our model and other methods in the YawDD dataset and self-built dataset. The Viola-Jones algorithm uses integral images to calculate its Haar-like features, which greatly reduces the amount of calculation. However, this algorithm was originally used to detect frontal face images, and it is not very robust to the detection of side face images. Therefore, its detection accuracy is low. The head pose estimation

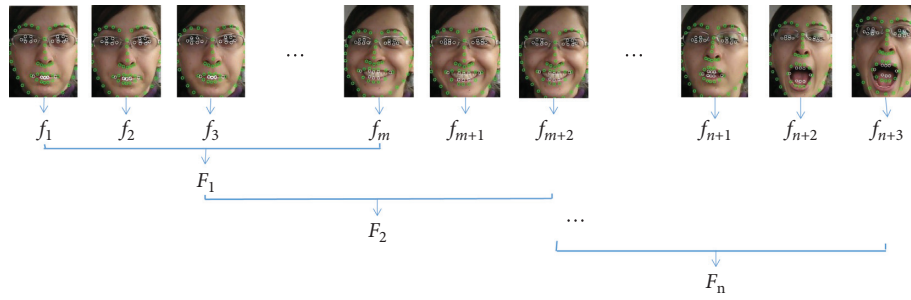


FIGURE 6: Temporal feature sequence composition diagram.

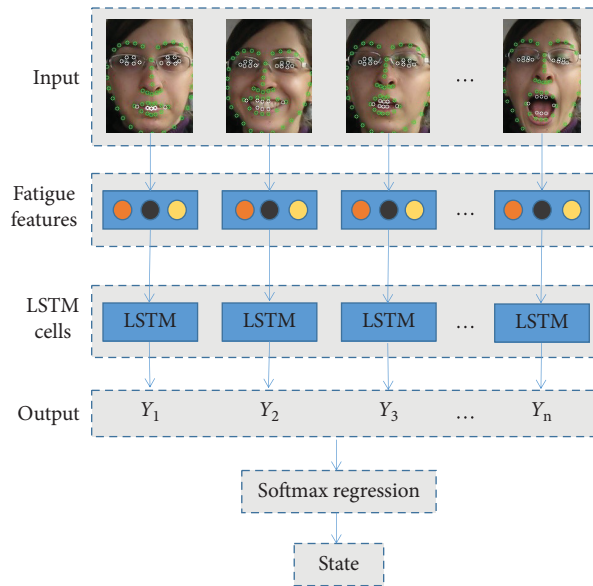


FIGURE 7: Fatigue recognition network diagram.

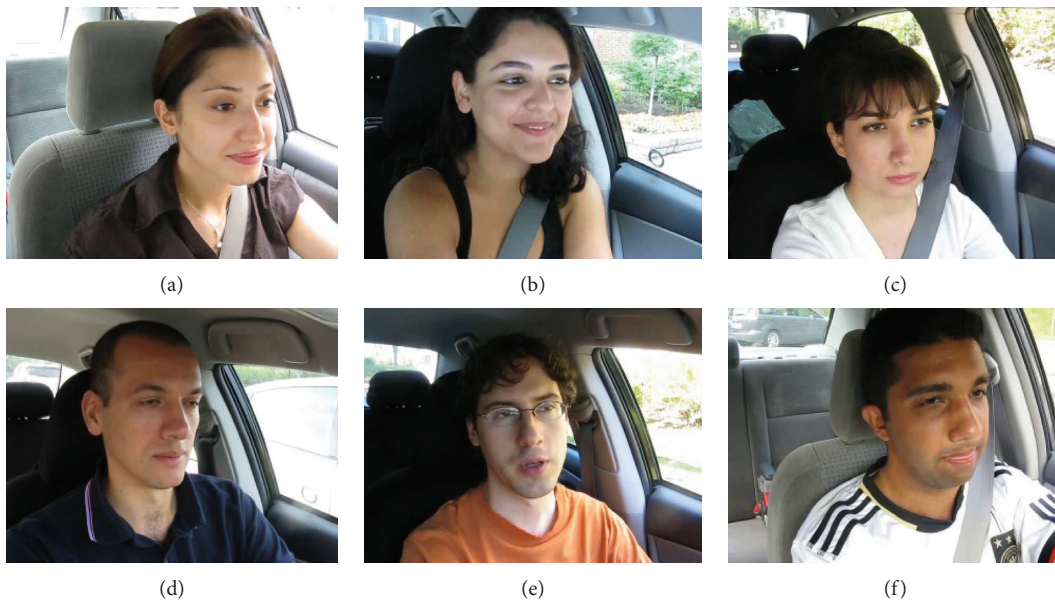


FIGURE 8: YawDD video dataset.

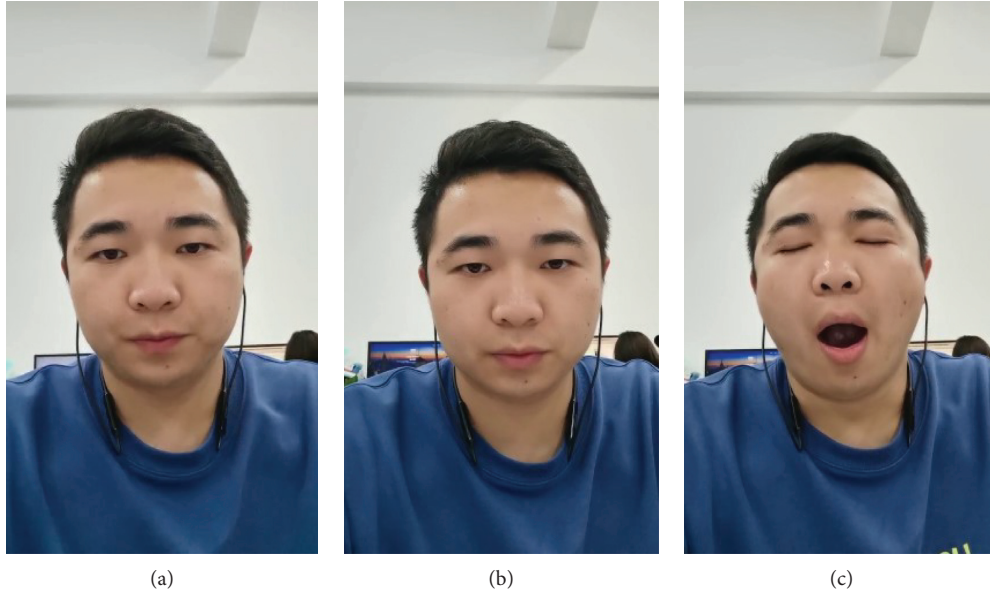


FIGURE 9: Self-built dataset.

TABLE 1: Detection accuracy of different areas in the YawDD dataset.

Algorithm	Face detection accuracy (%)	Eye detection accuracy (%)	Mouth detection accuracy (%)	Average detection accuracy (%)
Head pose estimation [27]	83	82	83	82.7
Viola–Jones [30]	73	79	81	77.7
Proposed	98	91	89	92.7

TABLE 2: Detection accuracy of different areas in the self-built dataset.

Algorithm	Face detection accuracy (%)	Eye detection accuracy (%)	Mouth detection accuracy (%)	Average detection accuracy (%)
Head pose estimation [27]	85	84	85	84.7
Viola–Jones [30]	77	83	84	81.3
Proposed	97	90	92	93

algorithm mainly uses the DLIB library to detect facial key points. The method proposed in this paper first uses MTCNN to extract the face and then uses the DLIB library to detect the key points of the face. In the process of detecting the key points of the eyes and mouth, the head pose estimation algorithm uses DLIB to detect the entire picture, which increases the amount of calculation and the detection rate is low. It can be seen from the data in the two tables that our method has a longer detection time than the Viola–Jones algorithm, but our average detection accuracy is 11%–15% higher than the Viola–Jones algorithm. Compared with the head pose estimation algorithm, the detection time is reduced by half, and the accuracy is increased by 8%–10%. Finally, we compared the accuracy of fatigue detection.

Tables 5 and 6, respectively, show the fatigue detection accuracy of our model and other methods in the YawDD dataset and self-built dataset. This study selected videos of drivers driving normally, talking, laughing, and yawning from the dataset and analyzed the results of driver fatigue through the state of the eyes and mouth. We use MTCNN + DLIB, DLIB + LSTM, head pose estimation method, and Viola–Jones method to compare the results with the method in this paper. When using DLIB + LSTM to detect the fatigue state, DLIB directly detects the entire picture, which not only takes a long time to detect but also has lower accuracy. The facial key points' detection accuracy directly affects the judgment of the fatigue state. When we use MTCNN + DLIB to detect the fatigue state, we only rely on the fatigue feature value of a

TABLE 3: Detection time of different areas in the YawDD dataset.

Algorithm	Face detection time (s)	Eye detection time (s)	Mouth detection time (s)	Average detection time (s)
Head pose estimation [27]	0.1675	0.1273	0.1328	0.1425
Viola-Jones [30]	0.033	0.0237	0.0319	0.0295
Proposed	0.1064	0.0447	0.0415	0.0642

TABLE 4: Detection time of different areas in the self-built dataset.

Algorithm	Face detection time (s)	Eye detection time (s)	Mouth detection time (s)	Average detection time (s)
Head pose estimation [27]	0.1655	0.1250	0.1323	0.1409
Viola-Jones [30]	0.0319	0.0245	0.0295	0.0286
Proposed	0.1082	0.0455	0.0413	0.065

TABLE 5: Fatigue detection accuracy in the YawDD dataset.

Method	Fatigue detection accuracy (%)
MTCNN + DLIB	79
Head pose estimation [27]	77
Viola-Jones [30]	82
DLIB + LSTM	74
Proposed	88

TABLE 6: Fatigue detection accuracy in the self-built dataset.

Method	Fatigue detection accuracy (%)
MTCNN + DLIB	85
Head pose estimation [27]	80
Viola-Jones [30]	85
DLIB + LSTM	75
Proposed	90

single frame to determine the fatigue state, but fatigue is a continuous time behavior. So, the accuracy of this detection method is significantly lower than our method. In addition to these two methods, we also select two methods with superior performance to compare with our method. It can be seen from the result in Tables 5 and 6 that the accuracy rate of our method has reached 88%–90%.

## 5. Conclusion

We proposed a fatigue detection algorithm based on facial key points and long short-term memory. Since the face contains more features than the eyes and mouth, it is easier to be detected. So, we first obtained the face image and marked the key points of the eyes and mouth in the face image. This can reduce the scope of the eyes and mouth test and also avoid the interference of the background area in the image. Fatigue is a continuous behavior. It is easy to make misjudgments if the result only relies on the eye and mouth features of a single frame, so we split the fatigue feature values of a single frame into a temporal fatigue feature sequence and sent it to LSTM network. Although our method is superior to other methods in the extraction accuracy of facial key points and the final fatigue

determination accuracy, the detection performance under insufficient light still needs to be improved. Our next step is to study fatigue driving detection in complex lighting environments and focus on the challenge of fatigue testing under poor light conditions, such as strong light and weak light. These application scenes are more practical and more difficult. When an automobile enters a tunnel or runs at night, how can we recognize the driver's fatigue driving behavior in time? This direction is also one of the current researches focuses in the field of fatigue driving detection.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interests regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant nos. 61872134 and



61672222, in part by Science and Technology Project of Transport Department of Hunan Province under grant no. 201935, in part by Science and Technology Program of Changsha City under grant nos. kh200519 and kq2004021, in part by National Key Research & Development Plan under grant no. 2017YFC1703306, and in part by School Level Project of Hunan University of Chinese Medicine under grant no. 2018GL01.

## References

- [1] S. Nordbakke, "Driver fatigue and falling asleep-experience, knowledge and action among private drivers and professional drivers," *Fatigue*, 2004.
- [2] S. Nordbakke and F. Sagberg, "Sleepy at the wheel: Knowledge, symptoms and behaviour among car drivers," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 10, no. 1, pp. 1-10, 2007.
- [3] P. Chen, "Research on driver fatigue detection strategy based on human eye state," in *Proceedings of the CAC*, Jinan, China, October 2017.
- [4] G. Sikander and S. Anwar, "Driver fatigue detection systems: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339-2352, 2019.
- [5] D. Ma, X. Luo, S. Jin, W. Guo, and D. Wang, "Estimating maximum queue length for traffic lane groups using travel times from video-imaging data," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 3, pp. 123-134, 2018.
- [6] T. D'orazio, M. Leo, and A. Distanto, "Eye detection in face images for a driver vigilance system," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 95-98, Parma, Italy, June 2004.
- [7] X. Sun, C. Lan, and X. Mao, "Eye locating arithmetic in fatigue detection based on image processing," in *Proceedings of the CISP-BMEI*, pp. 1-5, Shanghai, China, October 2017.
- [8] X. Ma, L. P. Chau, and K. H. Yap, "Depth video-based two-stream convolutional neural networks for driver fatigue detection," in *Proceedings of the ICOT*, pp. 155-158, Singapore, December 2017.
- [9] F. Zhang, J. Su, L. Geng, and Z. Xiao, "Driver fatigue detection based on eye state recognition," in *Proceedings of the CMVIT*, pp. 105-110, Singapore, February 2017.
- [10] R. Gupta, K. Aman, N. Shiva, and Y. Singh, "An improved fatigue detection system based on behavioral characteristics of driver," in *Proceedings of the ICITE*, Singapore, September 2017.
- [11] S. Junaedi and H. Akbar, "Driver drowsiness detection based on face feature and PERCLOS," in *Proceedings of the Journal of Physics: Conference Series 1090*, pp. 1-6, Ancona, Italy, June 2018.
- [12] B. K. Savaş and Y. Becerikli, "Real time driver fatigue detection based on SVM algorithm," in *Proceedings of the CEIT*, pp. 1-4, Bergen, Norway, October 2018.
- [13] A. Amodio, M. Ermidoro, D. Maggi, S. Formentin, and S.M. Savaresi, "Automatic detection of driver impairment based on pupillary light reflex," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 3038-3048, 2018.
- [14] T. Li, L. Wang, Y. Chen, Y. Ren, L. Wang, and J. Xia, "A face recognition algorithm based on LBP-EHMM," *Journal on Artificial Intelligence*, vol. 1, no. 2, pp. 61-68, 2019.
- [15] W. Liu, J. Qian, Z. Yao, X. Jiao, and J. Pan, "Convolutional two-stream network using multi-facial feature fusion for driver fatigue detection," *Future Internet*, vol. 11, no. 5, p. 115, 2019.
- [16] B. K. Savaş and Y. Becerikli, "Real time driver fatigue detection system based on multi-task ConNN," *IEEE Access*, vol. 8, pp. 12491-12498, 2020.
- [17] Z. Liu, Y. Peng, and W. Hu, "Driver fatigue detection based on deeply-learned facial expression representation," *Journal of Visual Communication and Image Representation*, vol. 29, no. 2, pp. 87-91, 2020.
- [18] Y. Ed-Doughmi, N. Idrissi, and Y. Hbali, "Real-time system for driver fatigue detection based on a recurrent neuronal network," *Journal of Imaging*, vol. 6, no. 3, pp. 1-14, 2020.
- [19] S. Jida, B. Aksasse, and M. Ouanan, "Face segmentation and detection using Voronoi diagram and 2D histogram," in *Proceedings of the ISCV*, pp. 1-5, Fez, Morocco, April 2017.
- [20] J. Zou and R. Song, "Microarray camera image segmentation with faster-RCNN," in *Proceedings of the ICASI*, pp. 86-89, Taiwan, China, April 2018.
- [21] X. Chen, X. Luo, X. Liu, and J. Fang, "Eyes localization algorithm based on prior MTCNN face detection," in *Proceedings of the ITAIC*, pp. 1763-1767, Chongqing, China, May 2019.
- [22] N. L. Arifin, H. Widiastuti, and A. Wibowo, "Study on effect of source to film distance (SFD) on the radiographic images," in *Proceedings of the ICAE*, pp. 1-4, Hongkong, China, August 2018.
- [23] P. Samangouei, R. Chellappa, M. Najibi, and R. Chellappa, "Face-magnet: magnifying feature maps to detect small faces," in *Proceedings of WACV*, pp. 122-130, Lake Tahoe, NV, USA, March 2018.
- [24] J. Donahue, L.A. Hendricks, M. Rohrbach, S. Venugopalan, and S. Guadarrama, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the CVPR*, pp. 2625-2634, Boston, MA, USA, June 2015.
- [25] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.
- [26] Y. Ji, S. Wang, Y. Zhao, J. Wei, and Y. Lu, "Fatigue state detection based on multi-index fusion and state recognition network," *IEEE Access*, vol. 7, pp. 64136-64147, 2019.
- [27] N. Zhang, H. Zhang, and J. Huang, "Driver fatigue state detection based on facial key points," in *Proceedings of the ICSAI*, pp. 144-149, Shanghai, China, September 2019.
- [28] C. Zhang, X. Lu, and Z. Huang, "A driver fatigue recognition algorithm based on spatio-temporal feature sequence," in *Proceedings of the CISP-BMEI*, pp. 1-6, Shanghai, China, October 2019.
- [29] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A yawning detection dataset," in *Proceedings of the 5th ACM Multimedia Systems Conference*, pp. 24-28, Singapore, March 2014.
- [30] M. Omidyeganeh, S. Shirmohammadi, S. Abtahi et al., "Yawning detection using embedded smart cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 570-582, 2016.