Mathematical Biosciences and Engineering

MBE, 17(4): 3356–3381.
DOI: 10.3934/mbe.2020190
Received: 26 February 2020
Accepted: 20 April 2020
Published: 29 April 2020

http://www.aimspress.com/journal/MBE

Research article

R2DS: A novel hierarchical framework for driver fatigue detection in mountain freeway

Feng You[1], Yunbo Gong[1], Xiaolong Li[1] and Haiwei Wang[2,*]

[1] School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510640, China
[2] School of transportation and economic management, Guangdong Communication Polytechnic, Guangzhou 510650, China

* **Correspondence:** Email: whw2046@126.com; Tel: +8613560108489.

Abstract: Fatigue driving is one of the main factors which affect the safety of drivers and passengers in mountain freeway. To improve the driving safety, the application of fatigue driving detection system is a crucial measure. Accuracy, speed and robustness are key performances of fatigue detection system. However, most researches pay attention to one of them, instead of taking care of them all. It has limitation in practical application. This paper proposes a novel three-layered framework, named Real-time and Robust Detection System. Specifically, the framework includes three modules, called facial feature extraction, eyes regions extraction and fatigue detection. In the facial feature extraction module, the paper designs a deep cascaded convolutional neural network to detect the face and locate eye key points. Then, a face tracking sub-module is constructed to increase the speed of the algorithm, and a face validation submodule is applied to improve the stability of detection. Furthermore, to ensure the orderly operation of each sub-module, we designed a recognition loop based on the finite state machine. It can extract facial feature of the driver. In the second module, eyes regions of the driver were captured according to the geometric feature of face and eyes. In the fatigue detection module, the ellipse fitting method is applied to obtain the shape of driver's pupils. According to the relationship between the long and short axes of the ellipse, eyes state (opening or closed) can be decided. Lastly, the PERCLOS, which is defined by calculating the number of closed eyes in a period, is used to determine whether fatigue driving or not. The experimental results show that the comprehensive accuracy of fatigue detection is 95.87%. The average algorithm rate is 32.29 ms/f in an image of 640 × 480 pixels. The research results can serve the design of a new generation of driver fatigue detection system to mountain freeway.

Keywords: fatigue detection; mountain freeway; DCCNN; finite state machine; PERCLOS

1. Introduction

Every year road traffic accidents, especially mountain freeway accidents, have caused severe damage to human health. According to the statistics from the World Health Organization (WHO) in 2015, traffic accidents are one of the main reasons of human death [1].

Fatigue driving in mountain freeway is one of the leading reasons of traffic accidents. National Sleep Foundation points out about 32% of drivers have fatigue driving experience at least once every month [2]. Fatigue driving is a harmful threat to the driver and other traffic participants. Countries all over the world have made laws to tackle this problem. For example, the Chinese Road Traffic Safety Law stipulates that: "Drivers are not allowed to drive continuously for more than 4 hours, and the rest period between every two long-duration driving should be no less than 20 minutes [3]." In Europe, the law requires that: "Drivers should stop and rest for every 4.5 hours of continuous driving, and the rest period should be no less than 20 minutes [4]." In the United States, the law provides that: "The cumulative maximum daily driving time must not exceed 11 hours, and the continuous daily rest time must not be less than 10 hours [5]." As mentioned above, fatigue driving is solely associated with driving duration. Without sufficient quantified indexes and reliable data analysis, it is subjective to determine the state of the driver. Consequently, study on the intelligent recognition of fatigue detection has significant practical meanings.

Recently, driver fatigue detection methods mainly include subjective evaluation and objective detection [6]. The former uses the sleep quality of the driver as the evaluation index of the fatigue state. It mainly including Karolinska Sleepiness Scale (KSS), Visual Analog Scale (VAS) and Stanford Sleepiness Scale (SSS) [7,8]. The subjective evaluation methods are simple to operate. However, the fatigue information is difficult to quantify, and the appraise is too subjective. The latter is based on information technology to objectively judge the fatigue state of drivers. Typically, the objective identification ways are on a basis of driver's physiological information, vehicle operation state, and driver's behavior characteristics.

Table 1. Comparison of driver fatigue detection methods.

Methods		Accuracy	Stability	Real time
Subjective evaluation		medium	low	low
Objective detection	Physiological information	higher	medium	high
	Vehicle driving state	lower	lower	medium
	Driver behavior characteristics	high	high	high

The method based on driver's physiological information [9–13] need collect physiological information such as Electroencephalogram (EEG), Electrocardiogram (ECG) and Electromyogram (EMG) during driving. The method uses the physiological signal characteristics of the driver as evaluation index of the fatigue state, and it has high accuracy. However, it has a negative influence on drivers' regular operation because of the contact sensors. Besides, the results are susceptible to noise. So, the anti-interference ability is low in scenarios with high noise.

The method based on the state of the vehicle [14–18] detects the fatigue state by analyzing data. It includes the trajectory of the vehicle or the steering wheel angle. It is the advantage of the way that the data is accessible. However, the correlation between the driving state and the fatigue state is weak. So, it is difficult to judge the fatigue state of the driver accurately.

The method based on driver behavior characteristics [19–24] uses the camera to catch the picture of the driver during driving. By analyzing the head posture, blink frequency and yawning, etc., the fatigue state of the driver is judged objectively. The method does not influence the regular operation of the driver due to non-contact sensors. Also, it has high accuracy and stability in the light stabilized scene. However, this method has to face mass data processing, and faster calculation speed is required. Besides, compared to the method of physiological information, the detection accuracy needs to be further improved. Table 1 is the performance comparison of driver fatigue detection methods.

In recent years, researchers have been working on enhance the accuracy, stability and real-time capability of the fatigue detection algorithm. But, most algorithms only focus on a single performance index, instead of taking all of them into consideration. With the rapid development of artificial intelligence technology, it makes a significant impact on the performance of detection and tracking algorithms. The fatigue detection algorithm on the basis of deep learning (DL) is causing more and more concern. However, the DL method is still in the stage of exploration for driver fatigue detection. And it has not yet formed an effective and universal framework. Besides, as a fatigue state data set to drive the deep learning model, it also urgently needs to be constructed. Therefore, it is crucial to establish a data set containing the fatigue status of the driver, and construct a fatigue detection framework which takes the three indexes into account.

In this paper, according to artificial intelligence technology, we propose a new universal three-layered framework for driver fatigue detection, called time and Robust Detection System (R2DS). It is a real-time and robust detection system, which has a sensational accuracy due to the application of deep learning method. As we know, deep learning methods, especially convolutional neural network model, greatly improve the accuracy of image recognition. However, the complex network structure reduces the algorithm speed. So, in order to improve the algorithm speed while ensuring the accuracy, we design a 3-layer sub-network in this paper. It greatly deals with the combination of accuracy, speed and robustness, with certain level of competing requirement. The R2DS consists of three modules specifically. The first module, facial extraction, is to obtain a stability facial image for the next algorithms. In the module, for facial detection and eye key points location, the paper designs a deep cascaded convolutional neural network (DCCNN). Besides, a face tracking sub-module based on improved Camshift is constructed, and on the basis of Support Vector Machine (SVM) classifier, a facial validation sub-module is applied to enhance the system stability. Then the second module named eyes region extraction is carried out. The eye areas are located according to the geometric features of faces and eyes. In the last module, the paper utilizes the ellipse fitting method to obtain the shape of driver's pupils. Eyes state can be decided according to the relationship between the long and short axes of the ellipse. At last, the PERCLOS, which is defined by calculating the number of closed eyes in a period, is used to determine whether in fatigue or not.

The contributions of the paper are as follows.

1). Proposed a three-layered framework for driver fatigue detection in mountain freeway. On the basis of a finite state machine model, build a face detection-tracking-verification-(re)detection recognition loop system. It effectively enhances the accuracy capability and robustness of the driver fatigue detection algorithm.

2). DCCNN is firstly applied to face detection and eye key points location. It improves the detection accuracy and speed.

3). Established a dataset containing information on the fatigue status, which compensated for the lack of the data sets on driver fatigue detection algorithm.

The paper is organized as follows. The first part is an introduction to the paper. The second part mainly introduces the research related to this paper. The third part presents the algorithm proposed. The fourth part is the experimental analysis, and the fifth part is the conclusion.

2. Related works

As mentioned above, the method based on driver behavior characteristics becomes one of research hotspot. In terms of related technology, it mainly covers traditional methods such as skin color segmentation and template matching. The fatigue state is judged by obtaining facial feature of driver. Besides, method based on statistical learning such as SVM classifier, Adaboost classifier, artificial neural network and so on, are common algorithms for fatigue detection. Recently, with the rapid development of calculation, the detection methods based on deep learning have been significantly improved.

Yan [25] used the facial mask to locate the eyes position, then to evaluate the fatigue state with PERCLOS. This method has excellent performance on individuals with obvious features, but the mask has a large influence on the generalization performance of the model. Niu [26] divided the face image in sequence into non-overlapping blocks of the same size, extracted multi-scale features by Gabor wavelet transform, and selected the most recognizable features for driver fatigue state detection with Adaboost. The algorithm proved to have good performance under different genders, postures and illuminations. Based on adaptive attenuation quantization, Shen [27] proposed an algorithm to enhance the night facial details of the driver. It can effectively improve the lower part of the environment with weak night light intensity, and improve the stable facial image for night driver fatigue detection. Besides, the application of an active infrared light source [28–34] has also become an important branch. Bergasa [35] located eye position with active near-infrared light source equipment. They used finite-state machine to confirm whether the eye is closed. They also applied a fuzzy system to evaluate the fatigue state. However, Bergasa's algorithm depends highly on hardware level; on the other hand, the effectiveness of the "bright eye effect" strictly relies on surrounding light conditions. Based on human-vehicle characteristics, Gupta [36] proposed a fatigue driving identification method by training SVM classifier. Zheng [37] extracted eyes features from the face image. The algorithm estimates the pupil center in HSV space. Then, the radius of the pupil is evaluated and optimized. Zhao [38] proposed a driving fatigue detection algorithm on the basis on the Convolutional Neural Network (CNN). The first network was trained to determine whether the human eyes or not, and the second network was trained to identify the coordinates of the eye feature points. Finally, the fatigue state was judged by PERCLOS. Zhang [39] applied the Camshift tracking algorithm to make the targeted areas detectable, even they were under occlusion. Then, according to the specific proportion relationship of the facial organs, the eye feature points were obtained. Finally, they used PERCLOS to determine driver fatigue state.

Although the technology of fatigue detection has made better progress and results, it still need to be improved.

1) The detection method based on vision usually uses Adaboost Classifier Algorithm for face

localization. However, when the driver wears glasses or sunglasses, light changes, and the face is partially occluded, Adaboost cannot accurately locate the face position and promptly warn of fatigue driving.

2) Due to its intricate network structure and enormous training data, the algorithms based on artificial neural network (ANN) unusually have poor real-time performance.

Consequently, it is crucial to driver fatigue detection how to hence the accuracy, real-time capability and stability.

3. Methodology

The overall structure of R2DS is shown in Figure 1. It includes three core modules, respectively, driver facial extraction, driver eyes extraction and driver fatigue feature extraction. Precisely, the facial extraction module includes three sub-modules, namely, face detection and eye key points location, face tracking, and face validation. The first sub-module, face detection and eye key points location, is designed to achieve a facial suspected area from the images captured from CCD. The facial area and eye key points are gathered from the images using DCCNN. The second sub-module, face tracking, applies an improved Camshift algorithm to track the face and reduce redundant data in the recognition process. The last sub-module, face validation, is arranged to accomplish a face verification algorithm, which is based on theory of support vector machine (SVM) with face database. To ensure the efficient cooperation of the three sub-modules, the paper utilizes the finite state machine (FSM) to coordinate the flow among sub-modules, and provides the switching conditions of each sub-module. Based on the process above, the recognition loop, detection-tracking-validation-(re)detection, is constructed.

The driver eyes extraction module processes the output of the driver face extraction module. Firstly, according to the posture of the driver's face, the rotation correction is performed. Then the eye areas are located according to the geometric features of the driver's face and eye.

The driver fatigue feature extraction module processes the output of the eyes extraction module. It gains the parameter which can show the fatigue status of the driver. In this module, the ellipse fitting method is applied to obtain the shape of the driver's pupils. According to the relationship between the long and short axes of the ellipse, eyes state (opening or closed) can be decided. Furthermore, the fatigue status is evaluated by PERCLOS, which calculated by the number of eye-closed in a period.

3.1. Driver facial extraction

Driver facial extraction is the basis and key stage for fatigue detection based on computer vision. As described above, the paper designs three sub-modules, face detection and eyes positioning, face tracking and face validation. Among them, the face detection sub-module constructs a deep cascaded convolutional neural network, which extracts the suspected face region from the complex background quickly. The face tracking sub-module combines the improved Camshift algorithm, which tracks suspected driver face areas to reduce computational time. The face validation sub-module constructs the driver classifier based on SVM, and confirms the suspected face area during the tracking process. To ensure the efficient cooperation of the three sub-modules, the paper applies the FSM to coordinate the operation flow between the sub-modules, and gives the switching

conditions of each sub-module.

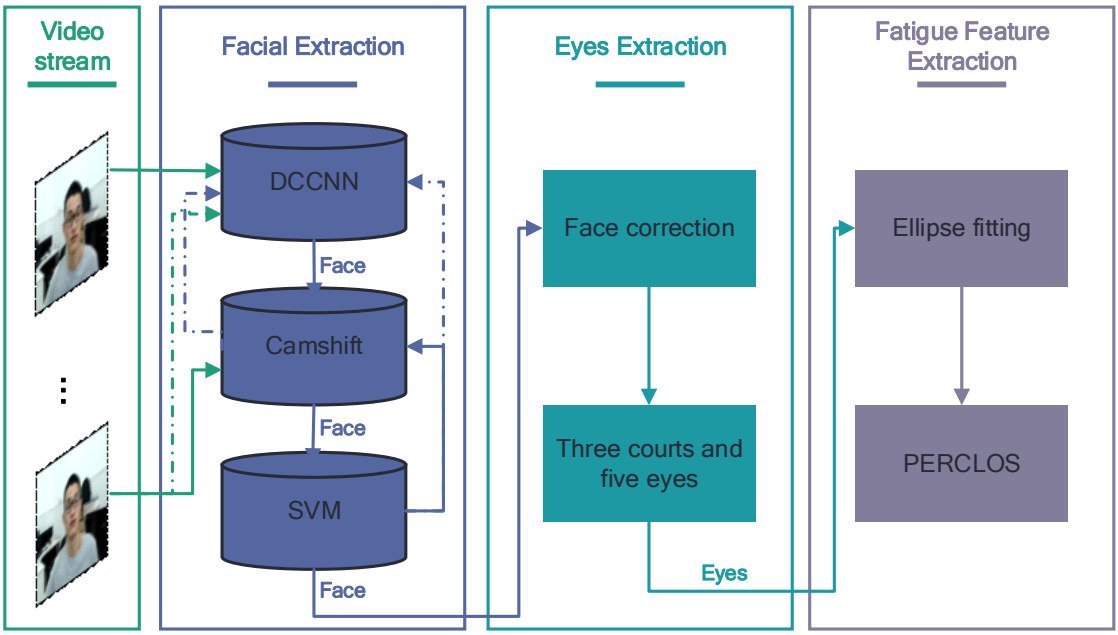


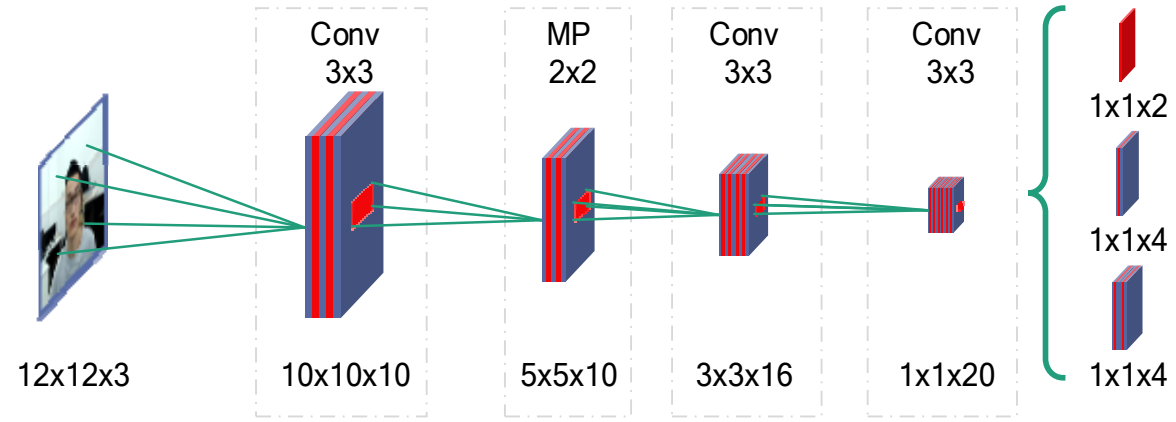
Figure 1. Driver fatigue detection algorithm structure. The solid lines in the figure show the direction of the data-flow when the module runs successfully. Conversely, the dashed lines show the direction of the data-flow when the module fails. For example, when the SVM verifies the face images, it transfers to the Camshift tracking module. And if not, it transfers to the DCCNN module. See Section 3.1 Finite State Machine for specific transfer logic.

3.1.1. Face detection and eyes location based on DCCNN

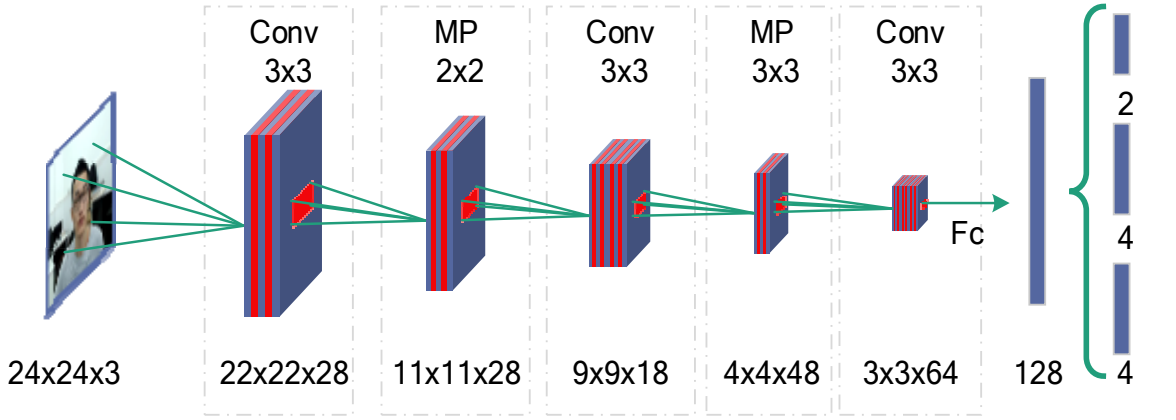
Face detection and eye key points location are the prerequisites of fatigue detection. The accuracy, real-time capability and robustness of the localization algorithm will be directly related to the result of fatigue detection. Specifically, the accuracy of the location algorithm will affect how to extract face and eyes from the complex background, which is beneficial to the identification of the subsequent fatigue state. The real-time performance of the location algorithm will determine the computational complexity of the algorithm, which is conducive to reducing the cost of hardware. The robustness of the location algorithm will improve the performance of anti-interference, which helps reduce the misjudgment rate of fatigue state.

Multi-task convolutional neural network (MTCNN) [40] has higher accuracy in face detection and key points location. The neural network structure includes three cascaded convolutional neural networks of different tasks, called Proposal Network (PNet), Refine Network (RNet), and Output Network (ONet). The algorithm principle is as follows. Firstly, by creating an image pyramid and input all images to the PNet, we obtain many bounding boxes of suspected face area in the PNet. And then, the layer network generates a large number of face suspected area bounding boxes. Then, non-maximum suppression [41–43] is used to eliminate tiny or overlapping bounding boxes. Additionally, the PNet results are exported into the RNet. Finally, the RNet outputs are input into

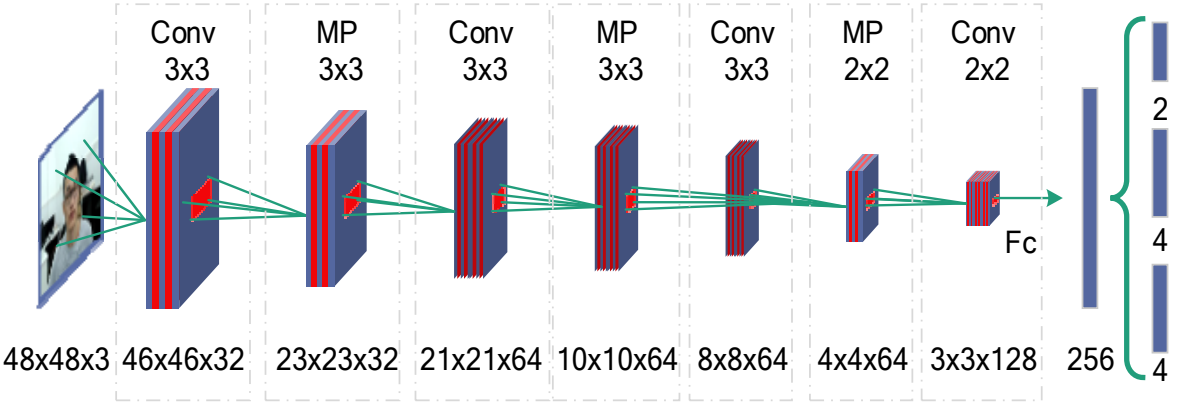
ONet, where bounding boxes and key points of the face are explored.



(a) Network-1



(b) Network-2



(c) Network-3

Figure 2. The structure of DCCNN. Where conv is the convolutional layer, MP is the largest pooled layer, and fc is the fully connected layer.

Under the framework of MTCNN, the paper designs a DCCNN to detect the area of driver's face. The network is shown in Figure 2. Similar to MTCNN, DCCNN covers three convolutional neural networks: Network-1, Network-2, and Network-3. Network-1 is a full convolutional neural network,

which will generate numerous suspected regions of the face. Network-2 processes the result of Network-1. The input image size of the Network is 24 × 24, so that it can adapt to different sizes of an image of driver's face. By further screening, the Network-2 keeps a small number of suspected face areas. Network-3 processes the result of Network-2. The network has multiple convolutional layers and pooling layers, which can extract face images with stricter facial features. And with high confidence in the output layer of Network-3, coordinates of face regions and eyes points are provided.

During training, we use the AFLW data set and the WIDER FACE data set as the driving data. Among them, The WIDER FACE dataset includes 32203 pictures and 393703 marked faces, which is one of the most common face databases. The AFLW face database contains 250 million hand-labeled face pictures, of which 59% are female, and 41% are male. On the basis of manually labeled face regions, the paper uses the WIDER_FACE and AFLW datasets to crop a number of face pictures. There are 2×10^5 face pictures, 5.9×10^5 partial face pictures and 8.9×10^5 non-face pictures.

The DCCNN training process is to continuously optimize the network parameters, until obtaining the optimal model. In order to show the impact of various parameters on network performance, a loss function is introduced during training. The loss function is an indicator, which measures the difference between the predicted output and the actual result. There are three tasks during the training process of DCCNN, severally, face classification task, face region fitting task, and eye key points locating task.

For the first task, the cross entropy [44] loss function is applied during training. For any sample x_i, the cross-entropy loss function is:

$$L_i^1 = -(y_i^1 \log(p_i^1) + (1 - y_i^1)(1 - \log(p_i^1))) \tag{1}$$

Where p_i^1 is the output of network, y_i^1 is the real label of (face/non-face) x_i.

The second and third tasks are used to forecast the coordinates of the face region, which is a regression problem. Consequently, we apply the Euclidean loss functions during training. The loss function of face region fitting task is shown in Eq (2). And the loss function of eye key points locating task can be expressed by Eq (3).

$$L_i^2 = \left\| p_i^2 - y_i^2 \right\|_2^2 \tag{2}$$

Where p_i^2 is the predicted coordinates of face region, and y_i^2 is the real coordinate of the face region.

$$L_i^2 = \left\| p_i^2 - y_i^2 \right\|_2^2 \tag{3}$$

Where p_i^3 is the predicted coordinates of the key points of eyes, and y_i^3 is the real coordinates of the key points.

3.1.2. Face tracking based on improved camshift

So far, the DCCNN has achieved the extraction of driver facial features. However, the driving fatigue detection system usually requires a fast speed in practical applications. The deep learning

network requires a large amount of calculation time, resulting in that the real-time capability cannot meet the demands of the driving fatigue detection system.

Target tracking aims to reposition the target quickly in the subsequent image, which is based on target detection. The application of target tracking can improve system speed significantly because of less unnecessary counting. To improve the speed of facial extraction, it is essential to design a tracking algorithm to enhance the real-time capability of the framework. Presently, face tracking algorithms mainly include Kalman filtering [45,46], particle filtering [47], meanshift [48], and Camshift [49–51]. Among them, the Camshift face tracking algorithm is simple and fewer calculations, which is suitable for face tracking of the driver. However, the Camshift tracking algorithm is sensitive to the color like skin, which leads to misjudgment of the face region. Therefore, we propose an improved Camshift face tracking algorithm. Based on Camshift algorithm, the paper introduces the variable weight histogram method. To avoid the problem of the tracking window flowing caused by background noise interference, it uses the detected face window as the initial window to enhance the weight of the face target area.

Assuming that the DCCNN detect that the center of the face window is (x_0, y_0), the image size is $m * n$ pixels, and the weight function is introduced as Eq (4).

$$\omega_c(x,y) = \sum_{x-x_0}^{m}\sum_{y-y_0}^{n}\sqrt{\frac{q_u(x_c,y_c)}{p_u(x,y)}}f(b(x,y),u) \tag{4}$$

Where $q_u(x_c, y_c)$ is the probability density function of the target window, $p_u(x_c, y_c)$ is the probability density function of the candidate window; $b(x, y)$ is the Eigen function; $f(b(x, y), u)$ is a function for determining whether $b(x, y)$ is an eigenvector.

After the weight function is introduced, the new search window center can be expressed by Eqs (5) and (6):

$$x_c = \frac{\sum_{x-x_0}^{m}\sum_{y-y_0}^{n}\omega_c(x,y)xI_c(x,y)}{\sum_{x-x_0}^{m}\sum_{y-y_0}^{n}\omega_c(x,y)I_c(x,y)} \tag{5}$$

$$y_c = \frac{\sum_{x-x_0}^{m}\sum_{y-y_0}^{n}\omega_c(x,y)yI_c(x,y)}{\sum_{x-x_0}^{m}\sum_{y-y_0}^{n}\omega_c(x,y)I_c(x,y)} \tag{6}$$

Where $I_c(x, y)$ is the gray value of the pixel points (x, y) position.

3.1.3. Face validation based on SVM

As mentioned above, the paper uses the face detection sub-module to mark the suspected region

of driver face. And then, the improved Camshift algorithm is applied to track the suspected area of the face in the image stream. However, the initial window of the mark is inevitably subject to small errors. After operating the tracking mode for a certain period, the face window may be shifted to the non-face area due to the accumulation of errors.

Thus, the paper designs a face validation sub-module to confirm the suspected face of the marked face, and judges whether there is a face in the marked box. When the tracking window drifts to the non-face window, the face validation sub-module provides identification feedback, thereby restarting the face detection sub-module.

To verify the suspected region of the face, there are two problems to be solved: The description of the face features and the discrimination of face. For the face feature description, the Histograms of Oriented Gradient (HOG) [52] maintains good invariance to image geometry and optical deformation. Besides, the HOG feature allows for subtle changes in the behavior of target. Thus, the paper selects HOG as the description method of facial features. For the discrimination of face, Support Vector Machine (SVM) [53] is suitable for binary classification problems with obvious machine learning characteristics, for instance, HOG. It can effectively avoid dimensionality disasters and simplify classification and regression problems. The paper chooses SVM as the face of the suspected area of the face and non-face validation algorithm. Figure 3 shows part of the positive and negative samples of the SVM classifier training.

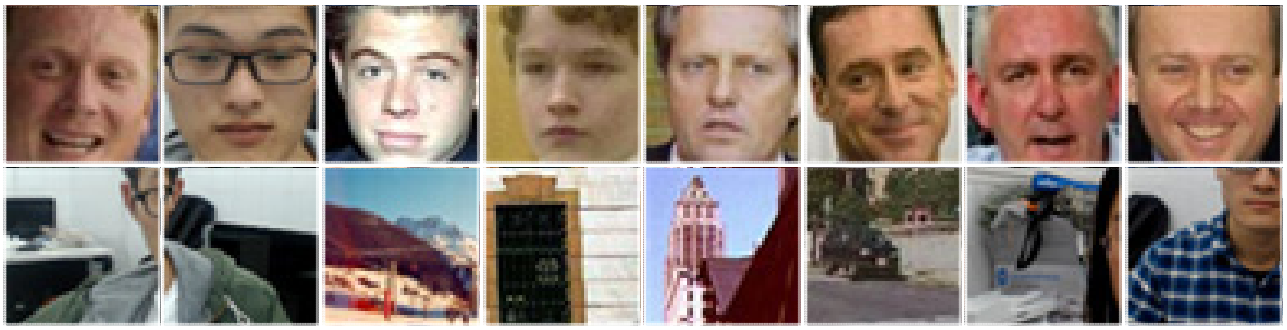


Figure 3. SVM face classifier training sample.

3.1.4. Facial collaborative recognition based on finite state machine

As mentioned above, the paper constructs the DCCNN to complete the driver face detection. Then, to improve the system calculation speed, an improved Camshift algorithm is designed for tracking the face that the DCCNN has detected. With the SVM classifier, it is verified that the result of the tracking is a face. To ensure that the three sub-modules work efficiently and orderly, the paper combines the finite state machine to control the operation flow between each sub-module, which form the detection-tracking-validation-(re)detection recognition loop. The Finite State Machine (FSM) can represent a limited number of states and behaviors, such as transitions and actions between states. Also, it is widely used in computers, communications, digital logic design, and software engineering [54].

As shown in Figure 4, when the face detection and eye key points location network are ready, the FSM starts from the initial state (INIT_STATE) and transitions to the detection state (DETECTION_STATE). Under DETECTION_STATE, the DCCNN detects the face of the driver

and locates eye key points from the video stream image. If the driver's face is detected, the running state is switched to the tracking window initialization state (TRACK_INIT). Under TRACK_INIT, the face suspected region obtained by the DCCNN is used as the initial window of the improved Camshift tracking algorithm. After the initialization is successful, the running state is transferred to the tracking state (TRACKING_STATE). In the TRACKING_STATE, Camshift initializes the window to detect the face of the subsequent image stream. After the tracking algorithm acquires the suspected area of the face in the subsequent video image, the running state is transferred to the face validation state (VALIDATION_STATE). Under VALIDATION_STATE, if the face is verified, it will move to the tracking state (TRACKING_STATE), and if it does not pass, it will be transferred to the face detection state (DETECTION_STATE) again.

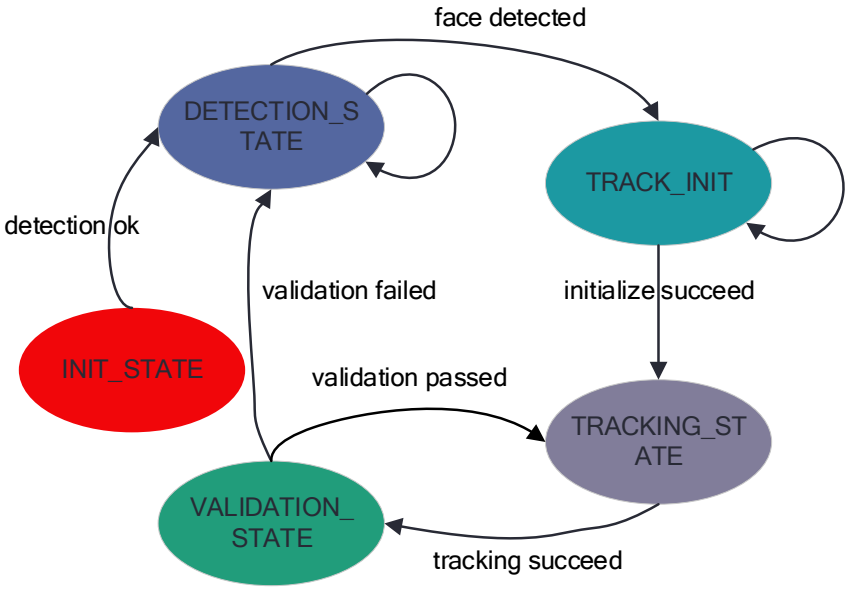
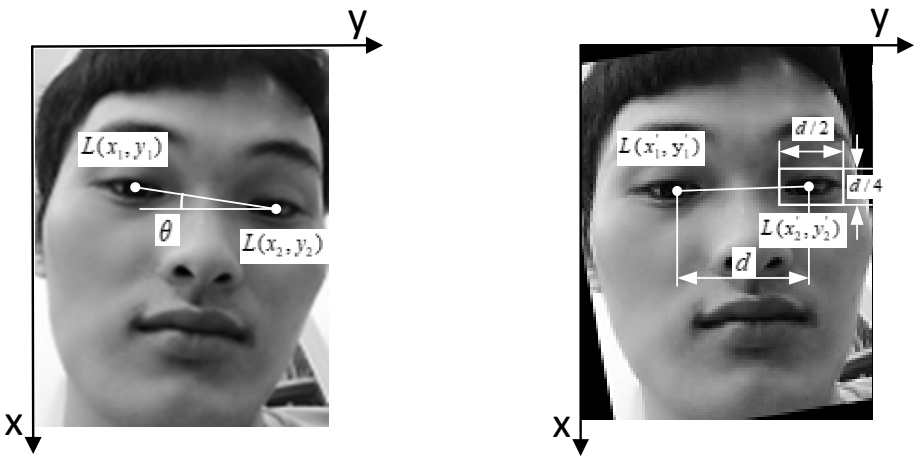


Figure 4. Driver facial extraction finite state machine model.



(a) original image (b) image after rotation operation

Figure 5. Rotating operation diagram.

3.2. Driver eyes regions extraction

In Section 3.1, the closed-loop is identified to extract the facial image from a complex background. The key steps above lay the foundation for the positioning of the eye area and the analysis of the eye state. According to the output of the DCCNN, the key points of the eyes are located, and the geometric features of faces and eyes is used to reposition the eyes area of driver. Considering that the head of the driver may have a certain degree of rotation during driving, the paper corrects the rotation of the face image, as shown in Figure 5.

The image is rotated by θ degrees around the center points according to the left eye key points coordinates $L(x_1, y_1)$, right eyes $L(x_2, y_2)$ and the image center points coordinate $O(H/2, W/2)$. The rotation calculation formula is as shown in Eqs (7–9).

$$\theta=\arcsin\left(\frac{|y_1-y_2|}{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}}\right) \tag{7}$$

$$T=\begin{bmatrix}\cos\theta & 0 & -\frac{H}{2}\cos\theta \\ \sin\theta & \cos\theta & -\frac{H}{2}(\sin\theta+\cos\theta) \\ 0 & 0 & 0\end{bmatrix} \tag{8}$$

$$P=TP_1 \tag{9}$$

Where T is the rotation matrix, H is the image height, W is the image width, $P_1=(x_0, y_0, 1)^T$ is the pixel coordinates before rotation, and $P=(x, y, 1)^T$ is the pixel coordinates after rotation.

After the rotation correction, a frontal image of the face can be obtained, as shown in Figure 5(b). The left and right eyes coordinates are $L(x_1^{'}, y_1^{'})$ and $L(x_2^{'}, y_2^{'})$. The center distance of the two eyes is calculated from the centers coordinates of the left and right eyes after the rotation correction, as shown in the Eq (10), and the eyes region is positioned by Eqs (10–12).

$$d=x_1^{'}-x_2^{'} \tag{10}$$

The coordinates (x, y) of the upper left corner of the left eyes area are:

$$x=x_1^{'}-d/8 \tag{11}$$

$$y=y_1^{'}-d/4 \tag{12}$$

The width and high of driver's eyes area are $d/2$ and $d/4$.

Based on the driver facial extraction, the paper extracts and verifies the suspected area of the eyes in the image. Then, by identifying the movement characteristics of the driver eyes, the fatigue

state of the driver is analyzed. Therefore, the correct positioning of the eyes is related to the result of the fatigue state discrimination.

To enhance the accuracy of the eyes area extraction and avoid the problem of poor image quality due to errors, the eyes validation module is introduced in this paper, and the HOG feature is combined with the SVM classifier method. The eyes validation module is constructed similarly to face validation, and will not be described here.

3.3. Parameter extraction and fatigue detection

After the eyes position is obtained, the driver fatigue feature extraction and the driver fatigue state determination are performed. Driver fatigue is a description of the state, and its corresponding fatigue level is a dynamic process. Drivers' facial changes are abundant, and with the occurrence of fatigue, drivers often show sleepiness and yawning. Therefore, the characteristic index of fatigue can be extracted from the eyes movement state, the head posture and the mouth state, thereby recognizing the fatigue state. Among them, when the driver is fatigued, the eyes of the driver movement state changes significantly, providing an observation dimension for fatigue detection. Wierwille [55] proposed the Percentage of Eyes Closure (PERCLOS) as a valid indicator for fatigue driving, which was widely accepted and adopted by a large number of researchers.

Given this, the paper chooses PERCLOS as an essential parameter for the fatigue state of the driver. PERCLOS [56] is a physical quantity that measures the state of human fatigue (sleepiness), defined as the time taken by the eyes to close in unit time. The US Federal Highway Administration and the US National Highway Traffic Safety Administration simulated driving in the laboratory, verifying the effectiveness of PERCLOS in characterizing the fatigue state. As defined by PERCLOS, it is important to conclude whether the eyes are opening or closed. Presently, PERCLOS usually has three criteria: 1) P80, namely, if the eyelid covers the pupil area more than 80%, it will be classified as closed; 2) P70, namely , if the eyelid covers the pupil area more than 70%, it will be classified as closed; 3) EM, namely, if the eyelid covers the pupil area more than 50%, it will be classified as closed. After experimental validation [57], the P80 standard has the best accuracy. Therefore, the paper uses the P80 standard as the fatigue state judgment index, that is, when the eyelid of driver covers the pupil area over 80%, it is determined that the eyes are closed in the present image, and the proportion of the eyes closure is counted in the specified time. The state of human fatigue is assessed. The formula for PERCLOS is:

$$PERCLOS = \frac{N_{close}}{N_{total}} \times 100\% \tag{13}$$

Where N_{close} is the number of closed eye images and N_{total} is the total number of images.

After selecting the fatigue state evaluation parameters, it is crucial to obtain the proportion of the eyelid covering pupil from the eye image. As shown in Figure 6, the pixel distribution of opening eyes images has a significant difference from closed. In the closed eyes, the eyelids completely cover the pupil, and the proportion of skin pixels in the image is large. Except for the eyelash area, the pixels in other areas are more evenly distributed. In the opening state, the proportion of black pupil pixels is large, and compared to the closed eyes state, the opening state image has a larger pixel distribution difference in the vertical direction. According to this, in conjunction with the P80

standard, the state of the eyes in the image can be determined.

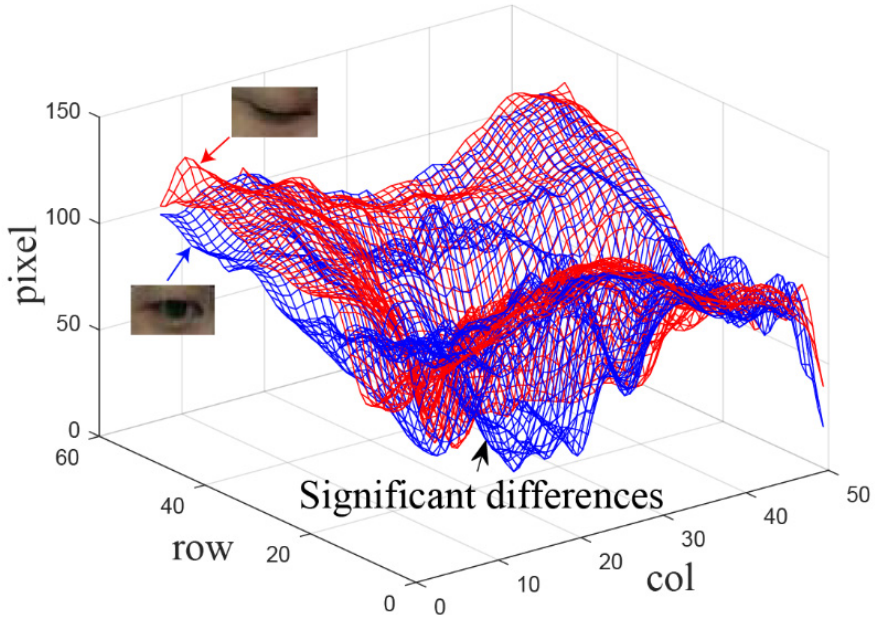


Figure 6. Pixel distribution in opening and closed eyes.

As can be seen from Figure 6, when the driver's eyes are completely open, the pupil is elliptical. Therefore, the paper first splits the eyes from the image and then performs ellipse fitting on the black pupil. The ratio of the long and short axes of the obtained ellipse is defined as the fatigue characteristic parameter. To simplify the calculation, $rate$ is used as the ratio of the pupil area to the pupil area. As shown in the Eq (14), when $rate < Th_{rate}$, it is determined that the eyes are closed.

$$rate = a / b \tag{14}$$

Where a is the elliptical minor axis length. b is the length of major axis and Th_{rate} is the threshold. According to P80 standard, Th_{rate} is set to 0.2

Based on extracting the driver fatigue characteristic parameter $rate$, the proportion of the eyes closed in a predetermined time is analyzed, and the $PERCLOS$ value is calculated. When $PERCLOS > Th_{PERCLOS}$, the driver is determined to be in a fatigue driving state.

4. Experiments

To certify the effectiveness of the framework, the paper designs a series of experiments. Firstly, using public dataset FDDB and the self-built dataset FSD as the test data, the performance of the DCCNN is evaluated. Secondly, a contrast experiment is arranged to corroborate the reasonability of detection-tracking-validation-(re)detection method. Finally, we construct an experiment to test the accuracy and speed of R2DS. And then, it is compared with relative algorithms.

4.1. Experimental environment and data sets

The experimental platform is the Intel Core i5-8400 with x86 architecture, and the CPU clock speed is 2.80 GHz. Graphics card is GTX1060 with Pascal architecture (CUDA: 9.2; CUDNN: 7.2), The RAM is 8 G DDR4, and the opencv3.4.6 image library is used. The deep learning computing framework is TensorFlow1.7. The environment of the program is in python 3.6.

As mentioned before, during training, we use the AFLW data set and the WIDER FACE data set as the driving data. But, in the phase of the experiment, different data is applied in the experiments, including public data sets and self-built data sets. The public data set is Face Detection Data Set and Benchmark (FDDB) [58]. FDDB is one of the prevalent evaluation datasets for face recognition algorithms. It includes 2845 images with 5171 faces, and the coordinates of each face region are marked already. However, FDDB data sets lack the information of the driver's eyes state, resulting in inability to evaluate the accuracy of fatigue detection algorithms. Consequently, we establish a Fatigue State Dataset (FSD), which contains 1,560 images of the driver. It adds the information of eyes state by manual marking. Among them, male drivers and female drivers accounted for 60 and 40%, with 835 eyes-opening images and 725 eyes-closing images. The experimental data sets information is shown in Table 2.

Table 2. Data sets.

data sets	Total images	Face number	Eyes-opening images	Eyes-closing images	Face border	Eyes coordinates	Eyes state
WIDER FACE	32203	393703	0	0	√	×	×
AFLW	250 million	250 million	0	0	√	×	×
FDDB	2845	5171	0	0	√	√	×
FSD	1560	1560	835	725	√	√	√

4.2. DCCNN performance evaluation

The DCCNN is a deep learning network, which identifies the driver's face and locates the key points of eyes. The performance of the DCCNN is directly related to the accuracy of the driver fatigue detection algorithm. Therefore, this paper evaluated the performance of DCCNN on the FDDB data sets and the FSD data sets firstly.

The paper selected accuracy (ACR) and ROC curve [59] as performance evaluation indexes. The accuracy is the ratio of the number of correctly detected samples to the number of all samples, which is an intuitionistic index to evaluate the performance of the network. However, the accuracy index is difficult to express the performance of the network, when the number of positive and negative samples is unbalanced. In the pattern recognition task, Sensitivity is the proportion of all positive samples are detected correctly. And specificity indicates the proportion of all negative samples are detected correctly. ROC curve is a comprehensive evaluation index that combines sensitivity and specificity.

4.2.1. Accuracy

In the tasks of face detection and eye key points location, the ACR is the ratio of the number of

detected images to the number of all images, as shown in Eq (15)

$$ACR = \frac{N_{\det ected}}{N_{total}} \tag{15}$$

Where $N_{\det ected}$ is the number of detected images and N_{total} is the total number of images

In the stages of training and validation of DCCNN, to measure the similarity between the suspected face area and the actual area of face, the Intersection over Union (IoU) parameter is introduced. IoU [60], a parameter, is used to evaluate the accuracy of detecting relevant objects in the peculiar data sets, as shown in Figure 7, where is the face area detected by the DCCNN, and is the marked area of face. The IoU can be computed by Eq (16).

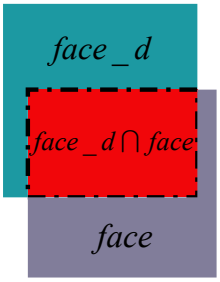


Figure 7. Diagram of IoU.

$$IoU = \frac{Area(face\_d \cap face)}{Area(face\_d \cup face)} \tag{16}$$

Where $Area(face\_d \cap face)$ is the area of $face\_d \cap face$ and $Area(face\_d \cup face)$ is the area of $face\_d \cup face$.

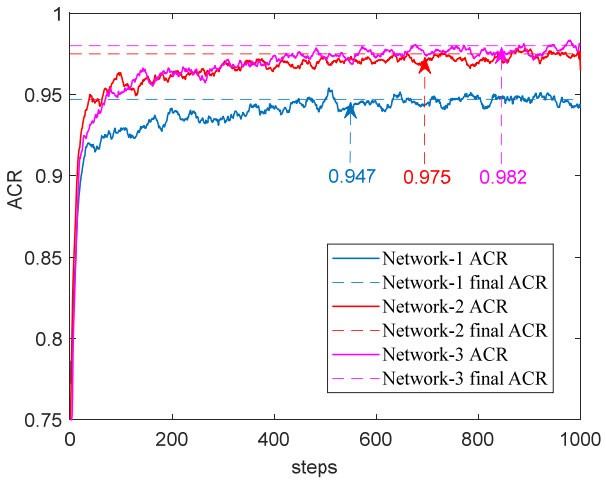


Figure 8. Accuracy of driver face detection and eye key points location.

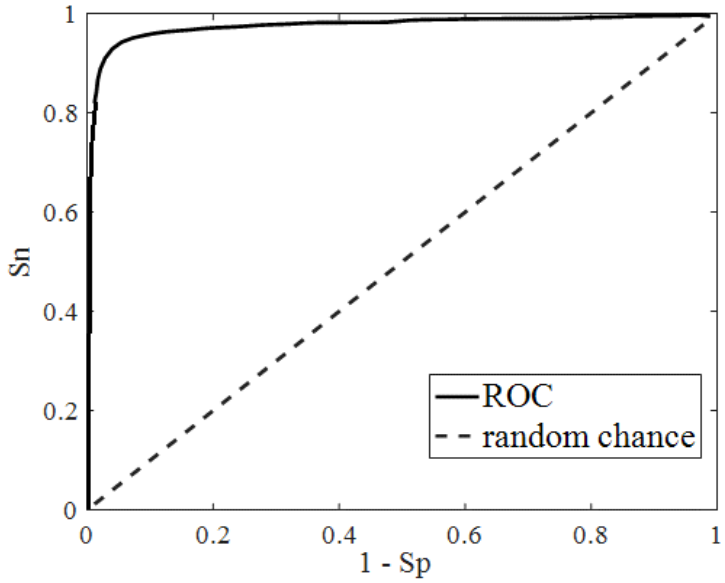


Figure 9. ROC curve in face detection and eye key points location.

As show in Figure 7 and Eq (16), it is indicating that the higher of IoU, the larger of overlap between prediction face area by DCCNN and the actual face area. Ideally, if $IoU = 1$, it means that the prediction area overlaps with the actual area. Generally speaking, it is considered that the accuracy of target detection network is excellent if $IoU > 0.5$. In [40], the author think the face is detected when the IOU above 0.65. In this paper, considering that the accuracy of DCCNN will affect the performance of follow-up algorithms, we affirm that face of driver is detected correctly when $IoU > 0.75$.

As mentioned above, the driver face detection model based on the DCCNN include three cascaded sub-convolution neural networks. Each sub-network are trained and validated independently.

Step 1: Train the Network-1 with 12×12 pixel image as the input.

Step 2: Use the candidate face area output of Network-1 as the input of Network-2, train the Network-2.

Step 3: Train the Network-3 with the candidate image output of Network-2.

As show in Figure 8, with the increase of the training steps, the ACR is gradually enhanced. The final ACR of the Network-1 is 94.7%, the final ACR of the Network-2 is 97.5%, and the final ACR of the Network-3 is 98.2%.

4.2.2. ROC curve

Sensitivity and specificity are important evaluation indexes of the pattern recognition model [61]. The TP, TN, FP and FN represent the amount of true positive, true negative, false positive and false negative data. The sensitivity and specificity can be expressed by Eqs (17) and (18).

$$S_n = \frac{TP}{TP + FN} \tag{17}$$

$$S_p = \frac{TN}{TN + FP} \tag{18}$$

The ROC curve can be drawn with S_n and $1 - S_p$. In the pattern recognition tasks, the closer the ROC curve is to the upper left corner, the higher the accuracy of the test is. Figure 9 shows the ROC curve in the face detection and eye key points location tasks. It indicates that the DCCNN has excellent performance in face detection and key points location.

In summary, the accuracy of face detection and eye key points location, on FDDB datasets and FSD datasets, shows that the DCCNN proposed in this paper has outstanding efficiency. Besides, ROC curve indicates that the DCCNN can avoid two kinds of errors in pattern recognition, that is, to confirm that the face can be detected correctly in the image while avoiding the wrong judgment.

4.3. Driver facial extraction

Except for considering accuracy, the real-time capability and stability of driver fatigue detection framework are also critical. In this section, based on finite state machine, three experiments are designed to compare the real-time capability and stability of the detection-tracking-validation-(re)detection model.

The paper selected 10 test drivers of different ages, genders and driving ages. For safety reasons, the fatigue driving experiments videos were taken from the simulator in mountain freeway. The test driver participated in three comparative tests: Test A, Test B, Test C, and their personal information is shown in Table 3.

Test A (Detecting): Only use the DCCNN to identify the face and locate eye key points. No tracking and validation module are applied.

Test B (Detecting + Tracking): Run DCCNN to identify the face and locate the eye key points. Then, apply the improved Camshift algorithm to track the face area in video stream.

Test C (Detecting + Tracking + Validating): Take the detection-tracking-validation-(re)detection framework, the specific steps are as follows:

Step 1: Use DCCNN to detect the suspected area of face and locate the eye key points. based on the suspected area, establish an initial window to for the improved Camshift tracking module.

Step 2: Apply the improved Camshift algorithm to track the face area of driver in video stream.

Step 3: Determine whether the tracking area is the face of driver with the SVM classifier. If yes, repeat step 2. Otherwise, transform to step 1.

Figure 10 is the algorithm speed of 10 subjects under different tests. It can be seen that the face detection sub-module based on the DCCNN has a slow speed, and the average speed is 46.34 ms/f. After applying the face tracking sub-module, the speed of driver facial extraction is greatly improved. The average speed is 19.30 ms/f, and the average speed of the detection-tracking-validation-(re)detection model is 23.67 ms/f.

Furthermore, to illustrate the importance of the face validation sub-module, the paper uses DCCNN to identify the face region as the ground truth bound, improved Camshift tracking suspected face area as candidate bound. Compare the difference between IoU with face validation sub-module and without face validation sub-module.

Table 3. Personal information of test drivers

	Gender	Age	Driving age	Resolution	Total frames
1	Male	23	2	480p	23280
2	Male	26	3	480p	21120
3	Male	41	7	480p	18300
4	Male	22	1	480p	15890
5	Male	37	6	480p	12876
6	Female	21	1	480p	20482
7	Female	26	4	480p	30144
8	Female	32	8	480p	25680
9	Female	40	10	480p	19630
10	Female	24	4	480p	32018

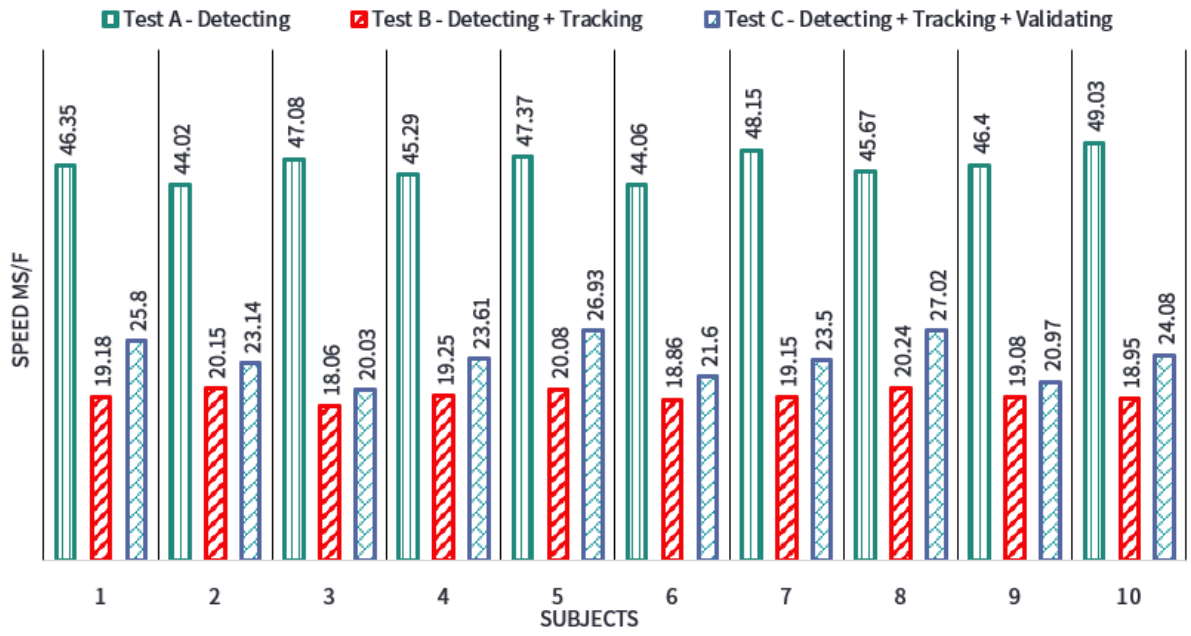


Figure 10. Algorithm speed of different tests.

Figure 11 shows the self-adaptive adjustment procedure with the face validation sub-module, which includes 4 stages. Stage 1 shows the process of face detection and eye key points location using DCCNN, and then put the detected face as the initial window for tracking. Stage 2 is the face tracking process. In this process, the improved Camshift algorithm is used to detect the suspected area of the face in the subsequent video stream. In stage 3, the face validation sub-module is not applied. The tracking window drifts to non-face area due to external interference. The IoU drops from 0.7 to 0 qiuckly and remains at 0 for a long time. The result indicates that the system does not have the capabilities of self-adaptive adjustment. The face validation sub-module is applied in stage 4. Therefore, when the tracking windows drifts to non-face area, the system will restart the face detection and eye key points location sub-module. The IoU rises from 0 to 0.7 quickly, which indicates that the system is robust with applying the face validation sub-module.

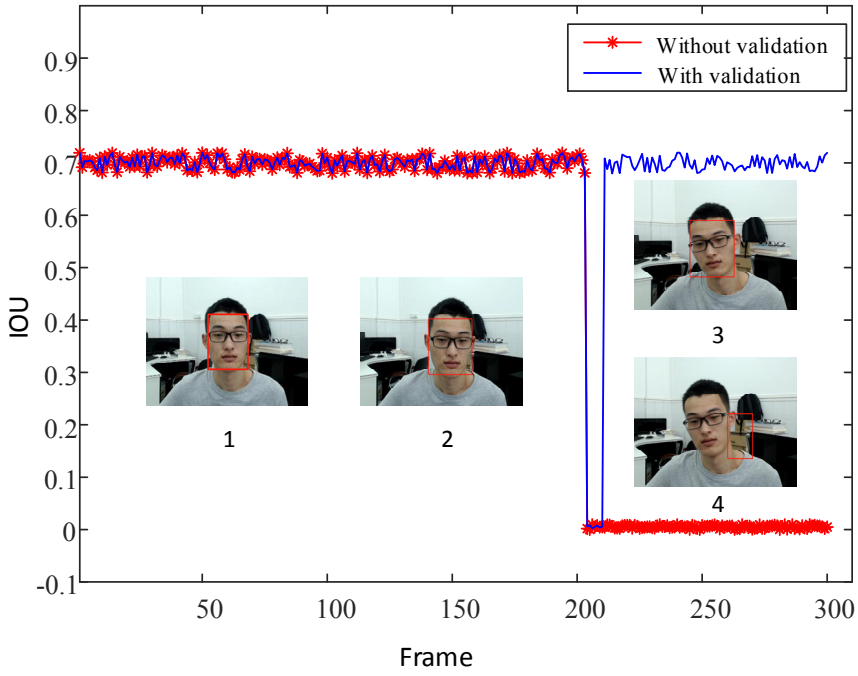


Figure 11. Self-adaptive adjustment procedure with face validation sub-module.

In summary, the recognition loop we proposed, detection-tracking-validation-(re)detection, considers both accuracy and real-time capability. In addition, through the introduction of the face validation sub-module, the framework has strong adaptive adjustment ability when encountering external interference.

4.4. Driver facial extraction

Face detection and eye key location are the foundation of fatigue detection. As described above, after segmenting the eye image from the original image, the ellipse fitting algorithm is used to fit the boundary of the pupil. Then, the ellipse length and short axis ratio is used as the fatigue state evaluation index. The driver fatigue evaluation test is divided into two steps.

Table 4. Fatigue detection accuracy and algorithm speed.

	accuracy/%		speed / $ms \cdot f^{-1}$
	Face detection and eye key points location AC_1	Fatigue detection AC_2	
1	98.01	96.32	31.83
2	97.85	95.82	32.76
3	98.24	96.46	32.02
4	96.97	94.79	29.65
5	97.04	94.96	35.18
average	97.62	95.87	32.29

Table 4 indicates the driving fatigue detection accuracy and algorithm speed. As we can see, the average accuracy of face detection and eye key points location is 97.62%. The average accuracy of fatigue detection is 95.87%, and the average algorithm speed of the framework is 32.29 ms/f.

Step 1: Randomly obtain 1000 images from the FSD data set as a group of samples for 5 times, then calculate the accuracy of face detection and eye key points location and fatigue detection. The accuracy of face detection and eye key points location can be obtained by Eq (15). If the number of images judged correctly in the sample is $R_{fatigue}$, the accuracy of the driving fatigue detection can be expressed by Eq (19):

$$AC_2 = \frac{R_{fatigue}}{AC_1 * N} \tag{19}$$

Where, AC_1 is the accuracy of face detection and eye key points location. N is number of samples.

Step 2: Select 5 driving videos to evaluate the real-time capability of the driving fatigue detection framework.

To prove the effectiveness of PERCLOS in fatigue detection, we select two videos with typical state (fatigue and non-fatigue) to calculate the PERCLOS within 30 seconds. Figure 12 shows the PERCLOS curve of the two typical states. It can be seen from the figure that the PERCLOS is always kept below 0.4 in non-fatigue state. In the fatigue state, the PERCLOS is mostly greater than 0.4.

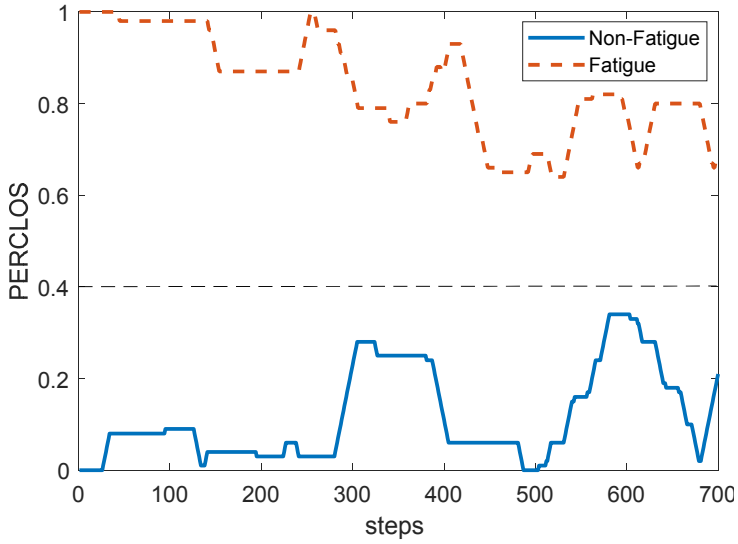


Figure 12. PERCLOS curve of fatigue and non-fatigue.

Table 5. Fatigue detect algorithm comparison.

Algorithms	accuracy/ %	speed/$ms{\cdot}f^{1}$
MTCNN + LRCN [62]	91.00	52.97
MTCNN + LSTM [63]	95.83	132.34
Ours	95.87	32.29

As illustrated in Table 5, the paper compared the framework proposed with the related algorithms. It indicates that compared with the MTCNN + LRCN [62] and MTCNN + LSTM [63] algorithms, the proposed method has higher accuracy of fatigue detection. Moreover, because of the application of detection-tracking-validation-(re)detection, the speed of fatigue detection algorithm has been significantly improved.

5. Experiments

In the mountain freeway, the traffic accidents caused by fatigue driving will bring severe losses. The driver fatigue detection system is an objective supplement to the laws and regulations. Accuracy, real-time capability and robustness are essential evaluation indexes for driver fatigue detection algorithm.

This paper constructs a new real-time and robust framework for fatigue detection named R2DS. To avoid extracting feature manually and enhance the accuracy of face detection algorithm, a deep convolutional neural network is designed to identify the face of driver and locate eye key points. The performance of the deep learning network is evaluated on the public dataset FDDB and the self-built dataset FSD, indicating that the proposed network has higher accuracy. On this basis, a recognition loop, detection-tracking-validation-(re)detection, is constructed. We designed a face detection sub-module using DCCNN, then applied an improved Camshift for face tracking sub-module. Finally, a trained SVM classifier was used for face validation sub-module. Experiments show that with the recognition loop proposed in the paper, the speed of face detection and eye key points location was enhanced significantly. Besides, the robustness of the system was significantly enhanced. Based on face detection and eye key points location, we perform the face rotation correction according to the coordinates of the key points firstly. Then, the eyes areas are located according to the geometric features of faces and eyes. Lastly, the ellipse fitting algorithm is used to fit the boundary of the pupil, and evaluating the driver fatigue state by calculating PERCLOS. Experiments show that the comprehensive accuracy of the driver fatigue detection framework proposed in the paper can reach 95.87%, and the algorithm speed is 32.29 ms/f in the 640 × 480 resolution image.

In the future, we will concentrate on the following research.

1). Upload the results of the fatigue detection to the cloud platform, and combine the big data analysis techniques to analyze the driver's fatigue period.

2). Integrate the fatigue driving detection algorithm into ADAS (Advanced Driving Assistant System).

3). Expand the applicable environment of the algorithm and explore the driver fatigue detection algorithm in night environment.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Granted Nos.: 51808151), Guangdong Provincial Natural Science Foundation (The Granted No.: 2020A1515010842), Guangzhou 2020 R&D plan for key areas, Guangdong Provincial Science and Technology Plan Project (2017A040405021), and the Fundamental Research Funds for Guangdong Communication Polytechnic (20181014).

Conflict of Interests

The authors declare there is no conflict of interest.

References

1. A. Amodio, M. Ermidoro, D. Maggi, S. Formentin S. M. Savaresi, Automatic Detection of Driver Impairment Based on Pupillary Light Reflex, *IEEE Trans. Intell. Transp. Syst.*, **20** (2019), 3038–3048.
2. X. Li, X. Lian, F. Liu, *Rear-End Road Crash Characteristics Analysis Based on Chinese In-Depth Crash Study Data*, 16th Cota International Conference of Transportation Professionals, 2016, 1536–1545. Available from: https://ascelibrary.org/doi/abs/10.1061/9780784479896.140.
3. G. Zhang, K. K. W. Yau, X. Zhang, Y. Li, Traffic accidents involving fatigue driving and their extent of casualties, *Accid. Anal. Prev.*, **87** (2016), 34–42.
4. D. Mollicone, K. Kan, C. Mott, R. Bartels, S. Bruneau, M. Wollen, et al., Predicting performance and safety based on driver fatigue, *Accid. Anal. Prev.*, **126** (2019), 142–145.
5. P. Chen, *Research on driver fatigue detection strategy based on human eye state*, 2017 Chinese Automation Congress (CAC), 2017, 619–623. Available from: https://ieeexplore.ieee.org/abstract/document/8242842.
6. G. Sikander, S. Anwar, Driver Fatigue Detection Systems: A Review, *IEEE Trans. Intell. Transp. Syst.*, **20** (2019), 2339–2352.
7. A. Anund, C. Fors, C. Ahlstrom, The severity of driver fatigue in terms of line crossing: A pilot study comparing day- and night time driving in simulator, *Eur. Transp. Res. Rev.*, **9** (2017), 31.
8. Y. Chang, Y. Feng, O. T. Chen, *Real-time physiological and facial monitoring for safe driving*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2016, 4849–4852. Available from: https://ieeexplore.ieee.org/abstract/document/7591813.
9. Z. Zhou, Y. Zhou, Z. Pu, Y. Xu, Simulation of pedestrian behavior during the flashing green signal using a modified social force model, *Transportmetrica A Transp. Sci.*, **15** (2019), 1019–1040.
10. Z. You, Y. Gao, J. Zhang, H. Zhang, M. Zhou, C. Wu, *A study on driver fatigue recognition based on SVM method*, International Conference on Transportation Information & Safety, 2017, 693–697. Available from: https://ieeexplore.ieee.org/abstract/document/8047842.
11. J. Hu, Automated Detection of Driver Fatigue Based on AdaBoost Classifier with EEG Signals, *Front. Comput. Neurosci.,* **11** (2017), 72.
12. W. Kong, W. Lin, F. Babiloni, S. Hu, G. Borghini, Investigating Driver Fatigue versus Alertness Using the Granger Causality Network, *Sensors*, **15** (2015), 19181–19198.
13. Z. Zhou, Y. Cai, R. Ke, J. Yang, A collision avoidance model for two-pedestrian groups: Considering random avoidance patterns, *Phys. A,* **475** (2017), 142–154.
14. D. Ma, X. Luo, S. Jin, D. Wang, W. Guo, F. Wang, Lane-based saturation degree estimation for signalized intersections using travel time data, *IEEE Intell. Transp. Syst. Mag.*, **9** (2017), 136–148.
15. D. M. Morris, J. J. Pilcher, F. S. Switzer III, Lane heading difference: An innovative model for drowsy driving detection using retrospective analysis around curves, *Accid. Anal. Prev.*, **80** (2015), 117–124.

16. Z. Li, S. Li, R. Li, B. Cheng, J. Shi, Online Detection of Driver Fatigue Using Steering Wheel Angles for Real Driving Conditions, *Sensors*, **17** (2017), 495.
17. A. D. McDonald, C. Schwarz, J. D. Lee, T. L. Brown, *Real-Time Detection of Drowsiness Related Lane Departures Using Steering Wheel Angle*, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, **56** (2016), 2201-2205. Available from: https://journals.sagepub.com/doi/abs/10.1177/1071181312561464.
18. D. Ma, X. Luo, W. Li, S. Jin, W. Guo, D. Wang, Traffic demand estimation for lane groups at signal-controlled intersections using travel times from video-imaging detectors, *IET Intell. Transp. Syst.*, **11** (2017), 222–229.
19. D. Ma, X. Luo, S. Jin, W. Guo, D. Wang, Estimating maximum queue length for traffic lane groups using travel times from video-imaging data, *IEEE Intell. Transp. Syst. Mag.*, **10** (2018), 123–134.
20. B. Mandal, L. Li, G. S. Wang, J. Lin, Towards Detection of Bus Driver Fatigue Based on Robust Visual Analysis of Eye State, *IEEE Trans. Intell. Transp.*, **18** (2017), 545–557.
21. N. Alioua, A. Amine, M. Rziza, Driver's Fatigue Detection Based on Yawning Extraction, *Int. J. Veh. Technol.*, **2014** (2014), 1–7.
22. K. J. Raman, A. Azman, V. Arumugam, S. Z. Ibrahim, S. Yogarayan, M. F. A. Abdullah, et al., *Fatigue Monitoring Based on Yawning and Head Movement*, International Conference on Information Communication Technology, 2018, 343–347. Available from: https://ieeexplore.ieee.org/abstract/document/8528759/.
23. X. Sun, H. Zhang, W. Meng, R. Zhang, K. Li, T. Peng, Primary resonance analysis and vibration suppression for the harmonically excited nonlinear suspension system using a pair of symmetric viscoelastic buffers, *Nonlinear Dyn.*, **94** (2018), 1243–1265.
24. C. Xu, Y. Yang, S. Jin, Z. Qu, L. Hou, Potential risk and its influencing factors for separated bicycle paths, *Accid. Anal. Prev.*, **87** (2016), 59–67.
25. J. J. Yan, H. H. Kuo, Y. F. Lin, T. L. Liao, Real-Time Driver Drowsiness Detection System Based on PERCLOS and Grayscale Image Processing, *2016 International Symposium on Computer, Consumer and Control (IS3C) IEEE*, 2016, 243–246. Available from: https://ieeexplore.ieee.org/abstract/document/7545182.
26. G. Niu, C. Wang, Driver Fatigue Features Extraction, *Math. Probl. Eng.*, **2014** (2014), 860517.
27. J. Shen, G. Li, W. Yan, W. Tao, G. Xu, D. Diao, et al., Nighttime Driving Safety Improvement via Image Enhancement for Driver Face Detection, *IEEE Access*, **6** (2018), 45625–45634.
28. A. Liu, Z. Li, L. Wang, Y. Zhao, *A practical driver fatigue detection algorithm based on eye state*, 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), 2010, 235–238. Available from: https://ieeexplore.ieee.org/abstract/document/5604919.
29. X. Ma, L. Chau, K. Yap, *Depth video-based two-stream convolutional neural networks for driver fatigue detection*, 2017 International Conference on Orange Technologies, 2017, 155–158. Available from: https://ieeexplore.ieee.org/abstract/document/8336111.
30. W. Di, R. Wang, P. Ge, F. Cao, *Driver eye feature extraction based on infrared illuminator*, 2009 IEEE Intelligent Vehicles Symposium, 2009, 330–334. Available from: https://ieeexplore.ieee.org/abstract/document/5164299.
31. B. Cyganek, S. Gruszczyński, Hybrid computer vision system for drivers' eye recognition and fatigue monitoring, *Neurocomputing*, **126** (2014), 78–94.

32. F. You, Y. Li, L. Huang, K. Chen, R. Zhang, J. Xu, Monitoring drivers' sleepy status at night based on machine vision, *Multimedia Tools Appl*., **76** (2017), 14869–14886.
33. L. Geng, X. Liang, Z. Xiao, Y. Li, Real-time driver fatigue detection based on polymorphic infrared features and depth learning, *Infrared Laser Eng.*, **47** (2018), 69–77.
34. H. Xiong, X. Zhu, R. Zhang, Energy Recovery Strategy Numerical Simulation for Dual Axle Drive Pure Electric Vehicle Based on Motor Loss Model and Big Data Calculation, *Complexity*, **2018** (2018), 4071743.
35. L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, M. E. Lopez, Real-Time System for Monitoring Driver Vigilance, *IEEE Trans. Intell. Transp*., **7** (2006), 63–77.
36. R. Gupta, K. Aman, N. Shiva, Y. Singh, *An improved fatigue detection system based on behavioral characteristics of driver*, 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE), 2017, 227–230. Available from: https://ieeexplore.ieee.org/abstract/document/8056914.
37. Z. Zheng, J. Yang, L. Yang, A robust method for eye features extraction on color image, *Pattern Recognit. Lett*., **26** (2005), 2252–2261.
38. X. P. Zhao, C. M. Meng, M. K. Feng, S. J. Chang, Fatigue detection based on cascade convolutional neural network, *J. Optoelectron.·Laser*, **28** (2017), 497–502.
39. F. Zhang, J. Su, L. Geng, Z. Xiao, *Driver Fatigue Detection Based on Eye State Recognition*, International Conference on Machine Vision & Information Technology, 2017, 105–100. Available from: https://ieeexplore.ieee.org/abstract/document/7878723.
40. K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks, *IEEE Signal Proc. Lett*., **23** (2016), 1499–1503.
41. X. Qu, Y. Yu, M. Zhou, C. T. Lin, X. Wang, Jointly Dampening Traffic Oscillations and Improving Energy Consumption with Electric, Connected and Automated Vehicles: A Reinforcement Learning Based Approach, *Appl. Energy*, **257** (2020), 114030.
42. S. Qiu, G. Wen, Z. Deng, J. Liu, Y. Fan, Accurate non-maximum suppression for object detection in high-resolution remote sensing images, *Remote Sens. Lett.*, **9** (2018), 237–246.
43. M. Zhou, Y. Yu, X. Qu, Development of an Efficient Driving Strategy for Connected and Automated Vehicles at Signalized Intersections: A Reinforcement Learning Approach, *IEEE Trans. Intell. Transp. Syst.,* **21** (2020), 433–443.
44. J. Tabor, P. Spurek, Cross-entropy clustering, *Pattern Recognit.*, **47** (2014), 3046–3059.
45. H. Lin, X. Tang, G. Ou, An Open Loop with Kalman Filter for Intermittent GNSS Signal Tracking, *IEEE Commun. Lett.*, **21** (2017), 2634–2637.
46. Z. Y. Lu, D. M. Wang, J. Wang, Y. Wang, A tracking algorithm based on orthogonal cubature Kalman filter with TDOA and FDOA, *Acta Phys. Sin.*, **64** (2015), 150502.
47. K.Y. Liu, Y. Li, S. Li, L. Tang, L. Wang, A new parallel particle filter face tracking method based on heterogeneous system, *J. Real Time Image Process.*, **7** (2012), 153–163.
48. W. Pairo, P. Loncomilla, J. R. Del Solar, A Delay-Free and Robust Object Tracking Approach for Robotics Applications, *J. Intell. Robot. Syst.*, **95** (2018), 99–117.
49. K. Chen, C. L. Liu, Y. Xu, *Face Detection and Tracking Based on Adaboost CamShift and Kalman Filter Algorithm*, Communications in Computer and Information Science, 2014. Available from: https://link.springer.com/chapter/10.1007/978-3-662-45261-5_16.
50. J. Lee, H. Jung, J. Yoo, A Real-time Face Tracking Algorithm using Improved CamShift with Depth Information, *J. Electr. Eng. Technol.*, **12** (2017), 2067–2078.

51. R. Zhang, Z. He, H. Wang, F. You, K. Li, Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system, *IEEE Access*, **5** (2017), 6649–6660.
52. S. Wei, S. Lai, Fast Template Matching Based on Normalized Cross Correlation with Adaptive Multilevel Winner Update, *IEEE Transp. Image Process.*, **17** (2018), 2227–2235.
53. X. LI, N. Tan, T. Wang, S. Su, *Detecting driver fatigue based on nonlinear speech processing and fuzzy SVM*, International Conference on Signal Processing, 2014, 510–515. Available from: https://ieeexplore.ieee.org/abstract/document/7015057.
54. F. J. R. Ruiz, I. Valera, L. Svensson F. Perez-Cruz, Infinite Factorial Finite State Machine for Blind Multiuser Channel Estimation, *IEEE Trans. Cognit. Communi. Networking*, **4** (2018), 177–191.
55. L. Lin, C. Huang, X. Ni, J. Wang, H. Zhang, X. Li, et al., Driver fatigue detection based on eye state, *Technol. Health Care*, **23** (2015), S453–S463.
56. S. Junaedi and H. Akbar, Driver Drowsiness Detection Based on Face Feature and PERCLOS, *J. Phys.*, **1090** (2018), 12037.
57. P. Zheng, Z. Song, Y. Zhou, PERCLOS-Based Recognition Algorithms of Motor Driver Fatigue, *J. China Agric. Univ.*, **7** (2002), 104–109.
58. V. Jain, E. Learned-Miller, *FDDB: A benchmark for face detection in unconstrained settings*, UMass Amherst Technical Report January, 2010. Available from: https://people.cs.umass.edu/~elm/papers/fddb.pdf.
59. J. Hernándezorallo, ROC curves for regression, *Pattern Recognit.*, **46** (2013), 3395–3411.
60. L. Tychsen-Smith, L. Petersson, *Improving Object Localization with Fitness NMS and Bounded IoU Loss*, IEEE Conference on Computer Vision and Pattern Recognition, 2017, 6877–6885. Available from: http://openaccess.thecvf.com/content_cvpr_2018/html/Tychsen-Smith_Improving_Object_Localization_CVPR_2018_paper.html.
61. R. S. Salcido, Diabetic Diagnostic Dilemmas: Sensitivity and Specificity, *Adv. Skin Wound Care*, **31** (2018), 149.
62. C. Zhang, X. Lu, Z. Huang, S. Xia C. Fu, *A Driver Fatigue Recognition Algorithm Based on Spatio-Temporal Feature Sequence*, 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, 1–6. Available from: https://ieeexplore.ieee.org/abstract/document/8965990.
63. Z. Xiao, Z. Hu, L. Geng, F. Zhang, J. Wu, Y. Li, Fatigue driving recognition network: Fatigue driving recognition via convolutional neural network and long short-term memory units, *IET Intell. Transp. Syst.*, **13** (2019), 1410–1416.

AIMS Press