Journal of Computing and Electronic Information Management

ISSN: 2413-1660

Classification of fatigue driving eye state based on single eye

Lian Xu [1, a], Xiaohong Ren [2, b] and Runxue Chen [3, c]

[1]School of Automation and Information Engineering, Sichuan University of Science and Engineering, Yibin, Sichuan, China

[2]Artificial Intelligence Key Laboratory of Sichuan Povince, Yibin, Sichuan, China

[3] School of Computer, Sichuan University of Science and Engineering, Yibin, Sichuan, China

[a]18381304961@163.com, [b]rx88@163.com, [c]18982976034@qq.com

Abstract: Due to factors such as illumination variation and head posture, the accuracy of existing fatigue driving eye state detection needs to be strengthened. Depending on to the consistency of the eye, an eye screening mechanism is proposed, which replaces the traditional binocular detection with monocular detection. The driver's face and eyes are detected by a multi-task cascade convolutional neural network, and the acquired monocular image is then input into a trained convolutional neural network classification model to determine the driver's eye opening and closing condition. The experimental results show that the accuracy of the classification model on self-acquired data is 98.20%, and the accuracy of the overall algorithm on the ZJU dataset is 96.70%. Compare to traditional methods, this method has high accuracy and strong generalization.

Keywords: Fatigue testing, an eye screening mechanism, face detection, multi-task cascade convolutional neural network, eye state recognition.

1. Introduction

According to the data of China Statistical Yearbooks, there were an average of 382,000 traffic accidents per year from 1995 to 2014, and 35% to 40% of traffic accidents were caused by fatigue driving. Fatigue driving detection technology can be roughly divided into three categories: behavior detection [1], physiological detection [2] and visual inspection [3]. Visual inspection has the characteristics of non-invasiveness and high accuracy, and has been widely used in the field of fatigue detection. Visual inspection mainly determines the degree of fatigue through three aspects: eye state, head

posture and facial expression. The eye includes rich information, which is not easily interfered by external interference and artificial suppression. Therefore, this paper focuses on the method of eye state recognition during fatigue.

Fatigue driving eye state recognition mainly includes face positioning, human eye key point detection and eye state classification. Yao Sheng et al. [4] used the local binary pattern texture detection operator to locate the eyes for infrared video images, and used the support vector machine (SVM) method to distinguish the opening and closing states of the eyes. Tang Yangshan et al. [5] used the Adaboost algorithm to locate the face, and positioned the eye according to the gray-value differential of the eye, and judged the state of the eye by calculating the proportion of melanin in the eyelid and pupil area. These eye state recognition methods are suitable for front face detection, and the detection accuracy is low when the face is occluded. In recent years, deep learning [6]-[7] has achieved amazing achievements in image processing, and is also involved in fatigue driving detection. Liu Junchao et al. [8] proposed a human eye detection method based on deep convolutional neural network, which uses human eye detection as a regression problem to achieve end-to-end eye state recognition process. Luo Yuan et al. [9] combined with gray scale projection and cascade convolution neural network to locate eyes, and use six feature points of human eyes to identify eye opening and closing degree. Although this method improves the detection speed, the detection accuracy is easily affected by head posture.

In order to solve the problem that eye state recognition is easily affected by head posture, this paper proposes an eye state recognition method based on eye screening mechanism, which uses deep learning to improve the accuracy of eye state recognition. First, captured video images are detected by the multi-task cascade convolution neural network (MTCNN), and the positions of the left and right eyes are located. And then obtaining an image of the left eye or the right eye to be detected through an eye screening mechanism. The trained eye state classification model (CNN) is used to judge the state of the eyes, which provide a basis for judging the fatigue state of the eye.

2. Algorithm

The basic step frame of eye state recognition in this paper is shown in Figure 1.

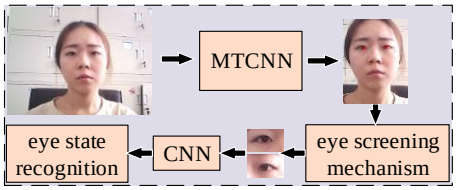


Fig. 1 Basic step frame for eye state recognition

Firstly, the captured video image is captured by MTCNN to obtain the driver's face

frame and the left and right eye pupil coordinates, and then the eye screening mechanism is used to select the eye image to be detected. Trained CNN is used to determine the closed state of the eye to be detected.

2.1 MTCNN Detection

Face detection and eye localization are the key parts of eye state recognition. Compare to traditional face detection, deep learning is more robust. Convolutional neural network (CNN) is a deep learning neural network, generally composed of an input layer. The convolutional layer, the pooling layer, the fully connected layer, and the output layer is constructed. CNN's multi-layer structure can automatically learn features and learn multiple levels of features. In the complex actual driving environment, a single CNN model can't meet the requirements of face key point positioning. This paper refers to Zhang [10]'s hierarchical design MTCNN. The three-layer network structure is simple to complex. The network structure is mainly divided into PNet and RNet. ONet three-layer network, the network structure is shown in Fig. 2.

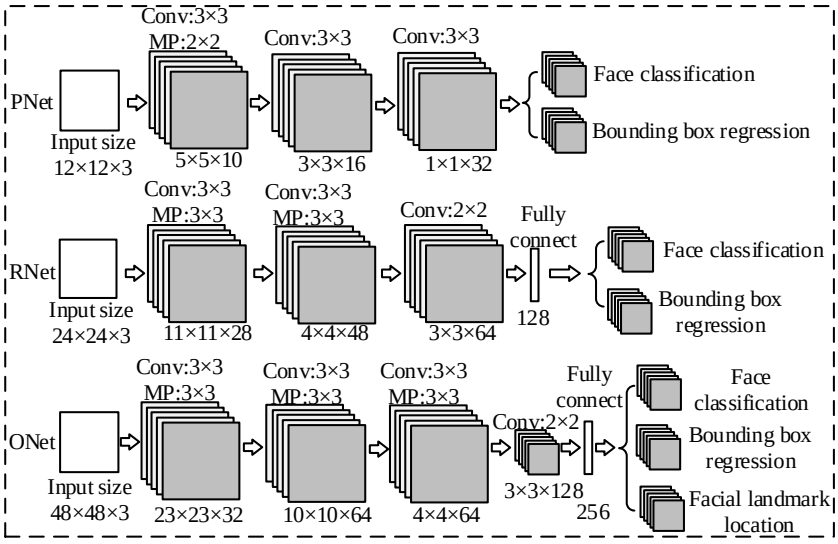


Fig. 2 MTCNN model structure

PNet uses the full convolutional network training to obtain the candidate facial windows and the bounding box vectors, and then calibrates the face candidate frame and removes it with Non Maximum Suppression (NMS). Remove the height-coincident candidate box. RNet adds a fully connected layer based on PNet, takes the candidate window of PNet prediction as input, filters out a large number of candidate blocks with poor effect, and then calibrates the candidate frame and further optimizes the prediction result by NMS. ONet has a convolutional layer relative to RNet, which uses more supervision to identify the area of the face, and finally outputs the face frame and five feature point coordinates.

The face detection dataset uses more than 20,000 images in WIDER FACE, each with a face frame label (X1, Y1, X2, Y2) in the image, where (X1, Y1) and (X2, Y2)

Represents the coordinates of the upper left and lower right corners of the face frame. The key point positioning data set uses 13466 pictures in the data set of [11], each picture gives the face frame label (X, Y, W, H) and 5 key point coordinates (including left eye, right eye, Nose tip, left mouth corner, right mouth corner), Where (X, Y) represents the coordinates of the upper left corner of the face frame, and W and H represent the width and height of the face frame, respectively.

Each network of MTCNN has three loss functions, including face classification, face frame regression, key point regression, and face classification using cross entropy loss function as shown in equation (2.1).

$$Loss_{class} = -[\mathrm{y}_i \cdot \log(\mathrm{p}_i) + (1-\mathrm{y}_i) \cdot \log(1-\mathrm{p}_i)] \tag{2.1}$$

Where $\mathrm{y}_i \in (0, 1)$ represents the label of the sample, the face is 1 and the non-face is 0, and p_i represents the probability that the sample is predicted to be a face. Face box regression uses the mean square error loss function as shown in equation (2.2).

$$Loss_{box} = -\frac{1}{N}\sum_{i=1}^{N}\left\|b_i - b_i\right\|_2^2 \tag{2.2}$$

Where N is the number of training samples, b_i is the detected face frame coordinates, and b_i is the coordinate of the face frame. The key point regression uses the mean square error loss function. The same formula is as follows.

$$Loss_{landmark} = -\frac{1}{N}\sum_{i=1}^{N}\left\|c_i - c_i\right\|_2^2 \tag{2.3}$$

c_i is the key point coordinates of the detection, and c_i is the key point coordinates of the label. For MTCNN, different tasks are performed in each CNN network, and there are different types of training images in different CNN networks. By integrating all tasks, the overall goal is set to the following formula.

$$Loss_{all} = \min\sum_{i=1}^{N}\sum_{j\in\{\text{class,box,landmark}\}}\alpha_j\beta_j^i Loss_j^i \tag{2.4}$$

α_j indicates the weight of different network structures, defined $\alpha_{class}=1.0$, $\alpha_{box}=0.5$, $\alpha_{lardmark}=0$ in PNet and RNet networks, defined $\alpha_{class}=0.5$, $\alpha_{box}=0.5$, $\alpha_{lardmark}=1.0$ in the ONet network. Since there is no training key point positioning in the first two layers of the network, the weight of the key point training is added in the third layer network, and the face key point positioning can be performed more accurately. $\beta_j^i \in \{0,1\}$ is the label indication.

2.2 Eye Status Recognition

Methods of eye state recognition are mainly based on feature analysis [12] and pattern classification [13]. These two methods have their advantages, but in the actual environment, the ideal effect can't be achieved in the case of light change and low image resolution. In this paper, an eye classification model CNN, combined with eye

screening mechanism is aimed at improving the detection range and accuracy.

2.2.1 Eye classification model

This paper constructs a convolutional neural network (CNN) to achieve eye classification. The network structure is shown in Fig. 3. The network layer includes four convolution layers, two pooling layers, three dropout layers, and two fully connected layers. The relu activation function is used after each convolutional layer, and the dropout layer with a ratio of 0.25 and 0.5 after the fourth convolutional layer and the first fully connected layer respectively prevents overfitting.

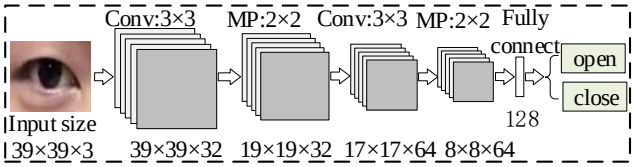


Fig. 3 eye state classification network structure

After two fully connected layers, the final classification is done through a softmax layer, and the softmax function is defined as shown in equation (2.5).

$$p_m = \frac{exp(x_m)}{\sum_{k=0}^{1} exp(x_K)}, m = 0,1 \tag{2.5}$$

Where P_m denotes the probability of classifying m, m is 1 for blinking, m is 0 for closed eyes, and x_m is the output of the last layer of the fully connected layer. The calculation formula of the fully connected layer is as shown in equation (2.6).

$$x_m = \sum_n z_n w_{n,m} + b_m \tag{2.6}$$

Where z_n represents the output of the previous layer, and $w_{n,m}$ and b_m represent the weight and offset of the last layer of the fully connected, respectively.

This network uses a binary cross entropy loss function, the formula is as follows,

$$L_n = -[y*log(P)+(1-y)*log(1-P)] \tag{2.7}$$

Where L_n represents the cross entropy of the mth sample, y represents the label of the eye sample, the blink is 1 and the closed eye is 0, and p represents the probability that the sample is predicted to be blinked.

2.2.2 Eye screening mechanism

The eye screening mechanism proposed in this paper was established with the consistency of the default binocular blink. When the face is tilted to an extreme situation, replacing the occluded eye picture with an unoccluded eye picture can improve the recognition accuracy of the eye in extreme situations without expanding the data set. When the head is in a non-extreme situation, the single eye has the same effect as both eyes. Taking the center point of the face frame as a reference point, When the horizontal coordinate of the midpoint of the positioning binocular pupil coordinate line is greater than the abscissa of the reference point, Then the picture left eye (the actual driver's right eye) is detected; otherwise the picture right eye (the

actual driver's left eye) is detected.

For the detected face size, this paper uses an adaptive method to cut the eye image. The eye is cut according to the eye position distribution and face size detected by MTCNN, as shown in the following formula.

$$\begin{cases} w = 8/25 * fw \\ h = 1/5 * fh \end{cases} \tag{2.8}$$

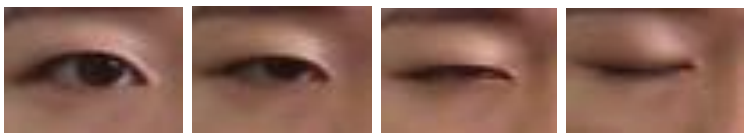
Where w and h represent the width and height of the cut eye, respectively, and fw and fh respectively indicate the width and height of the face frame detected by the MTCNN.

3. Experimental Analysis

The data collected, training, and testing covered in this paper is based on Python 3.6 and Tensorflow 1.2.1. The camera is a normal camera (640×480), the graphics card is Tesla T4, and the memory is 15G. MTCNN has a high accuracy for deflecting a certain angle of the face, and has good robustness, which lays a foundation for eye state recognition. This paper will analyze the experimental results from two aspects: a single eye and binocular detection and ZJU data set verification.

3.1 Data Collections

In this article, the default eye is only open and closed. Define a standard: if the iris and white part of the eye are defined as blinking, otherwise it is defined as a closed eye. Different eye states are shown in fig. 4, and as shown in figure 2(c), the eyes are actually open, but the iris is invisible. This situation of approximate eye closure is defined in this paper as closed eyes. This standard can be used in many practical application scenarios.



(a)open (b)open (c)close (d)close

Fig. 4 Different eye states

CNN training and test data are self-collected, taking into account the complexity of the actual driving environment. Self-collected eye images include wearing glasses, not wearing glasses, front, side and uneven illumination. Filter the dataset (without making a distinction between left and right eyes), select 12000 eyes images.

3.2 Model Training

The model training in this paper includes two parts: MTCNN training and CNN training.

3.2.1 MTCNN training

MTCNN uses Stochastic Gradient Descent (SGD) to train the model. During the training process, PNet iterates 30 times, RNet iteration 20 times, ONet iteration 18 times, and

the three-layer network training loss function curve is shown in Fig. 5. The ONet converges to 0.2 after a total loss rate of 40,000 steps, and the key point location loss function converges to 0.02, which can respond to experimental requirements.

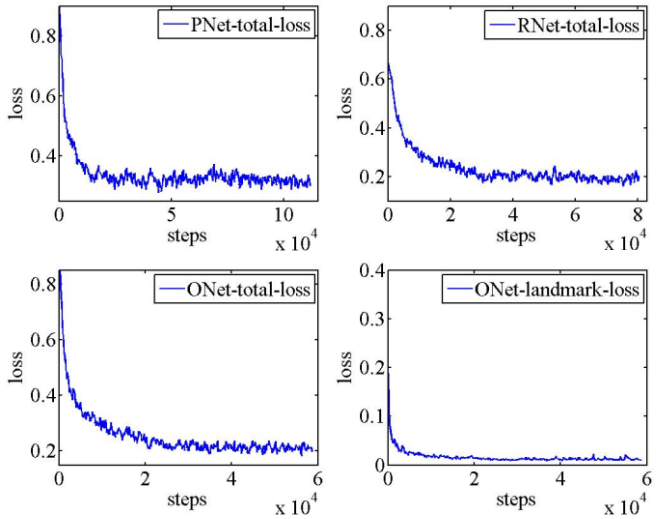


Fig. 5 MTCNN network loss function curve

The face detection and key point positioning process are shown in Fig. 6. The original picture is pyramided into different size pictures. Three kinds of training data of positive, negative and intermediate samples are generated by a random sampling method. The picture pyramid is input into the PNet network to obtain a large number of candidate frames, and then these pictures are finely adjusted through the RNet network to remove the candidate frame with poor effect. Finally, the coordinates of the five key points of the accurate face frame are obtained through the ONet network.

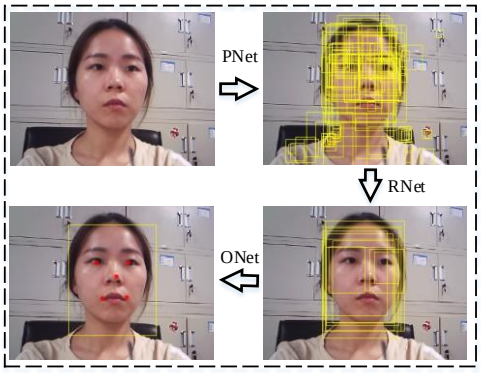


Fig. 6 Flow chart of face detection and key point positioning

3.2.2 CNN training

The network is trained by Stochastic Gradient Descent (SGD) and momentum (momentum) optimizers. In order to better evaluate the performance of the model and avoid over-fitting, the network uses the ten-fold cross-validation principle to divide the data set. For self-collecting eye data, 70% of the images were used for model training and 30% of the images were used for verification. It can be seen from Fig. 7 that the loss function curve basically converges to 0.05 after 200 iterations.

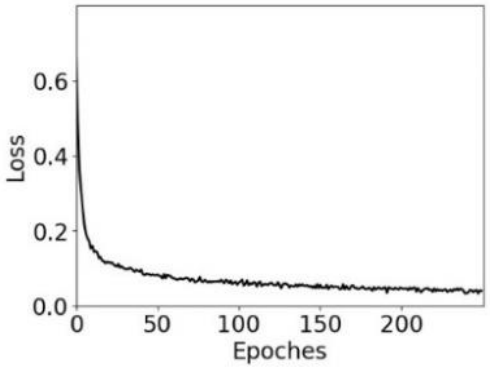


Fig. 7 eye classification network loss function curve

Table. 1 Accuracy of different algorithms in self-collected data sets

Algorithm	Accuracy(%)	time consuming (ms/frame)
LBP+SVM	94.36	80
LBP+AdaBoost	88.12	300
HOG+ AdaBoost	85.64	280
VGG-16[14]	96.81	12
ours	98.20	6

The trained eye state classification model was accurate to 98.2% in the self-collected data set. Table 1 shows the classification accuracy and algorithm time consumption of different algorithms on the self-acquisition data test set. It can be seen from Table 1 that the algorithm have the highest classification accuracy and the least time.

3.3 Contrast Binocular Detection And monocular Detection

3.3.1 Binocular detection

With the front of the driver as the central axis and the head deflected by 30 degrees in all directions, binocular detection can achieve better detection results. Figure 8 shows a partial representative eight head state experiment diagrams. The green box shows the detected eye area. Each box is labeled with the eye. Green "open" means open eyes, red "close" means closed eyes. As shown in Fig. 8(d) and (h), when the driver is tired, the spirit will be paralyzed, the head will deflect or frequent nodding will occur. In this way, the face collected by the camera is blocked by a large area, thereby causing a misdetection of the state of the eye.

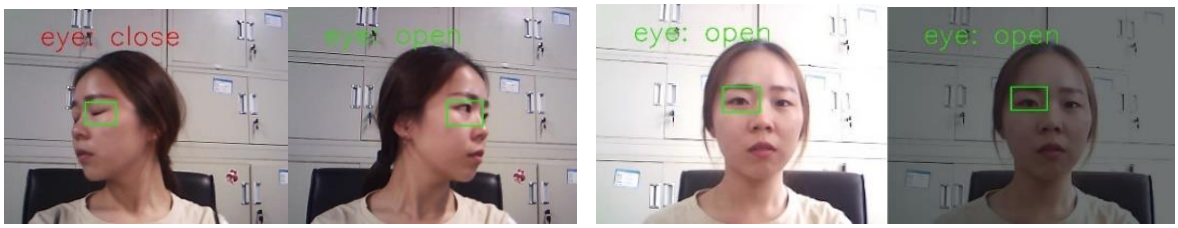
(a)close (b)close (c)close (d)close

(e)open (f)open (g)open (h)open

Fig. 8 Example of binocular test results

3.3.2 Single eye detection

As shown in Figs. 8(d) and (h), when the head is deflected to the extreme of the left and right, the occluded eye detects an error, but the eye that is not blocked is detected correctly. Therefore, this paper proposes an eye screening mechanism. The video streaming detection results are shown in Figure 9. Monocular detection not only has a good detection effect in the case of large head deflection, but also can adapt to a certain range of illumination changes.



(a) Large deflection of the head (b) changes in light intensity

Fig. 9 Example of monocular test results based on eye screening mechanism

3.3.3 Experimental comparison

This article uses the same data to test monocular detection and binocular detection. Yawning Detection Dataset(YawDD)[15] is a video data taken in a real environment, including speech, yawning, etc. Among them, 20 people do not wear glass and 10 people wear glass. The test results are given in table 2. Accuracy of single eye detection is higher than that of both eyes. The accuracy of this algorithm on the YawDD data set is 91.28%. The accuracy of wearing glasses is lower than that of glass. Since colored glasses frames have different degrees of occlusion on the eye area, it is more difficult to detect eye conditions. Among them, the error detection pictures include face positioning and eye positioning failure, or eye state detection error, etc.

Table 2 Compare test results for monocular detection on the YawDD data set

\	Glasses		No glasses		All	
	Eyes	Monocular	Eyes	Monocular	Eyes	Monocular
Picture(frame)	6916	6916	13089	13089	20005	20005
Error(frame)	1345	1019	1064	726	2418	1745
Accuracy(%)	80.24	85.27	91.87	94.50	87.91	91.28

and some detection pictures are shown in Fig. 10.

Fig. 10 Example of the algorithm in the YawDD dataset test image

3.4 ZJU Data Set Verification

ZJU [16] is a collection of blinking videos of 13 males and 7 females collected by Zhejiang University. Each person is photographed with thin-rimmed glasses, black-rimmed glass, no glass, and no glasses. The ZJU database is used for verification. The algorithm is for single-person detection, and the image of multi-person scene is removed. The detection results are presented in table 3. The overall accuracy of the algorithm in the ZJU data set is 97.51%.

Literature [17] proposes a new feature descriptor, the main direction gradient multi-scale histogram (MultiHPOG), which combines MultiHPOG, LTP, Gabor, and SVM to detect the state of the eye. Although the method has high accuracy, it takes a long time. It can be seen from Table 3 that the algorithm of this paper consume less time than the literature [17] under the premise of ensuring accuracy.

Table. 3 compares the detection effects of different algorithms on the ZJU dataset

\	Literature [17]	ours
Accuracy(%)	96.83%	97.51%
Face alignment and eye positioning time consuming	204	90
Eye state recognition time consuming	122	6

In general, the driver's blink frequency is 15-30 times per minute. For 400×300 images, the method proposed in this paper can detect about 700 frames of video images per minute running on the CPU. Normal people spend an average of 0.25-0.3s per blink. The algorithm in this paper only needs 0.096s to run each image on the CPU. Therefore, the algorithm in this paper can satisfy the driver's real-time eye fatigue state judgment.

4. Conclusion

This paper proposes a method for eye state recognition of fatigue driving with single eye detection. The eye screening mechanism is used to detect the single eye and output the driver's eye state in real time. The experimental results show that the

proposed algorithm has a high accuracy for the driver's left and right large head in the actual driving environment, and can adapt to certain lighting changes.

Acknowledgements

This paper was supported by Sichuan Provincial Department of Education Fund Project (No.17ZB0302)

References

[1] Huang Hao, "Fatigue driving detection based on driver behavior and vehicle stste", Southeast University, 2016

[2] Faramarz G, Gebraeil N S, Adel M, et al, "Detecting driver mental fatigue based on EEG alpha power changes during simulated driving", Iranian Journal of Public Health, 2015, Vol. 44(12), p1693-1700

[3] Cyganek B, Gruszczyński S, "Hybrid computer vision system for drivers' eye recognition and fatigue monitoring", Neurocomputing, 2014, Vol. 126, p78-94

[4] Yao Sheng, Li Xiaohua, Zhang Weihua, et al. "Eyes state detection method based on LBP", Application Research of Computers, 2015, Vol. 32(06), p1897-1901

[5] Tang Yangshan, Xu Zhongshuai, Yang Yuxiao, et al. "Extraction of drivers' fatigue feature fased on facial fxpression", Journal of Liaoning University of Technology(Natural Science Edition), 2018, Vol. 38(06), p57-60+65

[6] Lu Wei, Hu Haiyang, Wang Jiapeng, et al. "Tractor driver fatigue detection based on convolution neural network and facial image recognition", Transactions of the Chinese Society of Agricultural Engineering, 2018, Vol. 34(7), p192－199

[7] Wang Y, Huang R, Guo L, "Eye gaze pattern analysis for fatigue detection based on GP-BCNN with ESM", Pattern Recognition Letters, 2019, Vol. 123, p61-74

[8] Liu Junchao, Chen Zhijun, Fan Xiaochao, et al. "Eye detection based on deep convolutional neural networks", Modern Electronics Technique. 2018, Vol. 41(18), p72-75+79

[9] Luo Yuan, Yun Mingjing, Wang Yi1, et al. "Human fatigue detection based on eye information characteristics", Journal of Computer Applications. 2019, Vol 39(07), p2098-2102

[10] Zhang Kaipeng, Zhang Zhangpeng, Li Zhifeng, et al. "Joint face detection and alignment using multitask cascade convolutional networks", IEEE Signal Processing Letters, 2016, Vol. 23(10), p1499-1503

[11] Sun Y, Wang X, Tang X, "Deep convolutional network cascade for facial point detection", Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, p3476-3483

[12] Kim H, Jo J, Toh K A, et al. "Eye detection in a facial image under pose variation based on multi-scale iris shape feature", Image and Vision Computing, 2017, Vol. 57, p147-164

[13] Zhou Yunpeng, Zhu Qing, Wang Yaonan, et al. "Method of driver drowsiness detection based on fusion of multi-face clues", Journal of electronic measurement and instrumentation. 2014, Vol. 28(10), p1140-1148

[14] Qassim H, Verma A, Feinzimer D. "Compressed residual-VGG16 CNN model for big data places image recognition", 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2018, p169-175

[15] Abtahi S, Omidyeganeh M, Shirmohammadi S, et al. "YawDD: A yawning detection dataset", Proceedings of the 5th ACM Multimedia Systems Conference.ACM, 2014, p24-28

[16] G. Pan. ZJU eyeblink database [2019-5-26], http://www.cs.zju.edu.cn/ gpan/database/db _ blink.html

[17] Song F, Tan X, Liu X, et al. "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients", Pattern Recognition, 2014, Vol. 47(9), p2825-2838.