IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 5, 2021, accepted April 25, 2021, date of publication April 28, 2021, date of current version May 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3076365

Driver Eye Location and State Estimation Based on a Robust Model and Data Augmentation

YANCHENG LING[1], RUIFA LUO[2], XIAOXIAN DONG[1], AND XIAOXIONG WENG[1]
[1]School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510640, China
[2]College of Transportation Engineering, Tongji University, Shanghai 200082, China

Corresponding author: Xiaoxiong Weng (ctxxweng@scut.edu.cn)

ABSTRACT Eye state evaluation is crucial for vision-based driver fatigue detection. With the outbreak of COVID-19, many proposed models for eye location and state evaluation based on facial landmarks are unreliable due to mask coverings. In this paper, we proposed a robust facial landmark location model for eye location and state evaluation. First, we develop an existing lightweight face alignment model for eye key point locations that is robust in large poses. Then, to develop the performance of our model in a complex driving environment such as an environment with mask coverings, changing illumination, etc., we design a method to augment the training data set based on the original landmark data set without any extra cost. Finally, some facial landmarks around the eyes are extracted, and the eye aspect ratio (EAR) is introduced to evaluate the eye state based on eye key points. The experiment shows that our model achieves significantly improved landmark location performance on a driving simulation data set due to data augmentation. We tested our model on the BioID data set to measure the eye state evaluation performance, and the results showed that our model obtained satisfactory performance with an accuracy of approximately 97.7%. Further testing on the driving simulation data set shows that our model is robust in different driving scenarios with an average accuracy of approximately 93.9%.

INDEX TERMS Eyes state evaluation, eye location, driver fatigue detection, data augmentation, EAR.

I. INTRODUCTION

Eye detection and state estimation are important in our daily lives for their wide use in eye gaze estimation, driver fatigue detection, human-robot interaction, and other applications [1], [2]. Research studies show that approximately 1/5 of traffic accidents in China occur due to fatigue [3], and more than 30% of divers experience fatigued driving each month [4]. Traffic accidents have a high correlation with fatigued driving in our daily lives [5]–[7]; and many methods have been developed to detect fatigued driving, mainly including physiological features, vehicle running characteristics, and facial features [8]–[10]. With the development of computer technology, driver fatigue detection based on vision has become increasingly popular due to its real-time performance and reliable detection results [11], [12]. Drowsy drivers often have intrinsic visual characteristics across their faces, especially related to eye states, blinking, and yawning. Therefore, it is crucial to locate eye and analyze eye states for fatigue detection. With the outbreak of COVID-19, there are many new challenges for all walks of life. many proposed models for eye location and state evaluation based on facial landmarks are unreliable due to mask coverings. Robust model structure and enough data are important to develop the performance on complex conditions. However, though lots of public datasets for face alignment, few of them specialize in this scenario. Thus, it is significant to develop suitable model and proposed an effective method to augment existing dataset without extra costs.

A. METHODS FOR EYES LOCATION

Many research studies on eye location have been conducted. Traditional techniques for eye localization can be categorized as eye characteristic models, statistical appearance models, and structural information.

An eye characteristic model mainly exploits the shape feature or intensity contrast between the eyeball and eyes white to locate the eyes. Yuille *et al.* [13] proposed a method to detect faces and features of faces by using deformable templates. They use a parameterized template to describe

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

different features of interest and incorporate edges, peaks, and valleys in their model. They have to use a large continuous parameter space to fit the model to a testing image. The Hough transform technique was used to detect the circular shape of the iris in [14], and O. Jesorsky *et al.* [15] use the Hausdorff distance to fit the test image to the general model. Dynamic template [16], radial symmetry [17], and other developed characteristic methods have been proposed to locate eyes, and most of them perform well in controlled environments. However, their robustness is poor in complex and uncontrolled environments due to a lack of information on eye appearance.

Eye appearance-based approaches obtain proper appearance features of cropped eye patches using different methods and build a statistical model to locate eyes. Many popular feature sets, such as Haar-like features [18], Harr wavelet features [19], Gabor features [20], and gradient-based features [21], are used. Based on these feature sets, the Support Vector Machine (SVM) [18], [22], [23], Principal Component Analysis (PCA) [24], AdaBoost [18], [25] and neural networks [26] are most commonly used to build classification models. In [18], a hybrid eye location model was proposed. First, a couple of AdaBoost classifiers trained with Haar-like features were used to select possible eye locations, and then an SVM was used to select the best pair of eyes among all candidate locations. Though these approaches are more robust than an eye characteristics model, they cannot perform well in complicated environments such as those with poor illumination and low-resolution images.

The structural information among eyelids, pupils, irises, and these components is less affected by the environment, and many research studies based on structural information have been proposed. In [27], the Active Shape Model (ASM) was proposed based on the structural information of objects. In [28], Hough-transform technique was used to locate the eyes. In [29], an enhanced pictorial structure model was proposed. A discriminative pictorial structure model and a series of global constraints were introduced to develop the performance of proposed model in eye location. Compared to eye appearance-based approaches, the structural information is usually integrated into a statistical eye model and has more reliable detection results in complicated uncontrolled conditions.

Compared to traditional techniques, models based on convolutional neural networks are becoming increasingly more popular due to their powerful feature extraction abilities. Sun *et al.* [30] design a three-level convolutional network (DCNN) to detect 5 facial landmarks. Zhou *et al.* [31] proposed a four-level convolutional network cascade based on a DCNN to detect 68 facial landmarks. Zhang *et al.* [32] proposed a multitask cascaded convolutional network (MTCNN) to locate the face and 5 facial landmarks at the same time. In [33], a deep alignment network (DAN) was proposed, and landmark heatmaps were used to increase the accuracy of the model. To improve the detection speed and reduce the storage size of the model, Liu *et al.* [34] propose a weight binarization cascade convolution neural network. Compared to conventional models for eye location, eye location methods based on deep learning are more robust in complex environments.

B. METHODS FOR STATE ESTIMATION AND FATIGUE DETECTION BASED ON EYE STATES

Eye state estimation can be achieved based on precise eye locations, and traditional eye estimation methods can be classified as (1) shape-based, (2) template-based, and (3) learning-based. Regarding shape-based models, in [35], the shape of the eye edge was used to detect the eye state; and in [36], the eye contour information was used to evaluate the eye state. An eyes open template and an eyes close template are used to detect eye states in [37], and rich eye patch information is used in [38]. Regarding learning-based methods, the neural network, Support Vector Machine (SVM) [39], and AdaBoost were used to learn eye state features in the training phase and classify eye states in the testing phase.

For driver fatigue detection based on eye states, increasingly more eye location and eye state evaluations based on facial landmarks have been proposed. In [40], the MTCNN was used to detect the face, and a Convolutional Experts Constrained Local Model (CE-CLM) [41] was used to locate 68 facial landmarks and obtain eye states and fatigue parameters. In [42], the AdaBoost algorithm was used to detect the face, and facial landmarks were detected by a cascade regression. A CNN model was used to classify eye states based on extracted eye regions. In [43], the MTCNN was used to locate the face and eyes, and a fatigue detection convolutional network (FDCN) was designed to detect fatigue. In [12], [44]–[47], the OpenCV Dlib toolkit [48] was used to detect 68 facial landmarks, and eye states were evaluated based on it.

C. METHODS FOR DATA AUGMENTATION

Data augmentation has been proven to be an effective method to improve the performance of deep learning models in many research studies [49], [50]. Many technologies such as flipping, rotation, scaling, cropping, translation, and deep learning technology [51], [52] are used to expand datasets to improve the performance of models in complex environments. With the development of GAN [53], more and more data augmentation technologies based on developed GAN have been proposed. In [54], a framework that converts daytime images into synthetic nighttime images based on a generative adversarial network was proposed, the generative adversarial network was trained on a public dataset and the experiment in a real nighttime dataset demonstrated that the performance of the model develops a lot due to augmented dataset. In [55], a generative adversarial network was used to generate real traffic sign images and the experiment demonstrated that the augmented dataset could develop the performance of the model, however, compared to GAN data augmentation, some traditional augmentation techniques could have a better performance which means that GAN is can be naively used for data augmentation but is not always the best choice. There are also some novel techniques for data augmentation. In [56], infrared and visible images were fused based on multi-scale transformation and norm optimization, and the experiment on different public datasets showed that the newly proposed method has better performance in terms of highlighting targets and retaining effective detail information. In [51], a new data generation pipeline was proposed to generate low-light paired images from the daytime images and a light enhancement net (LE-net) based on a convolutional neural network was trained by the paired images. The results demonstrated that the generated low-light images based on LE-net are satisfactory both in quality and quantity.

With the outbreak of COVID-19, there are many new challenges for all walks of life. Masks are essential to protect us, especially for public transport drivers, from the risk of virus infection. However, none of the existing methods consider a large covered area. It is difficult to precisely locate facial landmarks because of masks sheltering the face and masks' ability to reflect light at night. Increasingly more complex network structures result in high computational costs and memory capacity as well. Generally, there are two main challenges for eye location and eye state evaluation based on facial landmarks: (1) Robust landmark location performance under complicated scenarios, such as large poses, mask coverings, illumination, etc. (2) Finding a simple model network structure with low computational costs and high detection speed.

In this paper, we proposed a robust method to locate eyes and monitor the changes in eye states based on facial landmarks for fatigue detection. First, we introduce a face landmark localization algorithm that has a lightweight structure and is robust under larger poses. Then, we design an efficient method to enlarge the face landmark training dataset without any extra costs and obtain a more robust model based on the extended data set. Finally, the eye aspect ratio (EAR) is used to monitor the state of the eyes based on precise facial landmarks. The main contributions of this paper are summarized as follows:

1. A developed face landmark location algorithm that is robust in large poses is introduced to obtain eye key points and locate eye positions.
2. We proposed an effective method to expand existing training datasets without any additional labor costs and obtain a more robust and accurate eye key points location, which decreases the influence of mask coverings and reflected light at night.
3. The EAR is introduced to monitor eye states based on eye key points.

The structure of the remainder of this paper is organized as follows. We present our methods and model in Section II, and the experimental results and evaluation are included in Section III. In Section IV, we present the conclusion.

II. APPROACH

In this section, we introduce our main methods and models for eye location and state evaluation. They mainly include PFLD-eyes, an effective method to augment the training dataset, and eye state evaluation with the EAR.

A. PFLD-EYES

Although many face landmark location algorithms are used for eye detection and state evaluation in driver fatigue detection and perform well in ideal environments, few of them have a satisfactory landmark location with the influence of large poses, changing illumination, and mask coverings. Moreover, many proposed face landmark location models cannot be used in real life due to their detection speed. The main defects of existing models can be summarized as follows: (1) Lack of robustness in a real environment. (2) Prolonged detection speed.

Deep learning has been widely used in image processing due to its powerful feature extraction ability and outstanding performance. With the development of lightweight frame structures such as SqueezeNet, MobileNets, ProjectionNet, etc., increasingly more models achieve real-time detection speed without any loss of accuracy. The Practical Facial Landmark Detector (PFLD) [57] is a robust facial landmark location model with a lightweight structure. It performs well under large poses with a real-time detection speed. Its exceptional performance is due to the novel loss function and light structure. The conventional quadratic loss function is:

$$\mathrm{L}_{\mathrm{loss}} = \frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N}\gamma_n||d_n^m||_2^2 \tag{1}$$

where $\|\cdot\|$ designates a certain metric to measure the distance/error of the nth landmark of the mth input. N is the number of landmarks to predict per face, and M presents the sample number of each batch size. γ_{n} is a parameter that is always set to 1.

However, the novel function loss is:

$$\mathrm{L} = \frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N}(\sum_{k=1}^{K}(1-\cos\theta_n^k))||d_n^m||_2^2 \tag{2}$$

where γ_n was replaced with $\sum_{k=1}^{K}(1-cos\,\theta_n^k)$ in the novel loss function, which represents the head pose angle deviation value in the pitch, yaw, and roll between the estimated value and the ground truth value. In the training phase, the branch structure of the novel loss function could measure facial gestures θ^k and supervise the backbone network to learn extra facial geometrical features. Therefore, it would experience substantial improvements in larger poses.

Although PFLD performs well under large poses, the influence of mask coverings and the illumination intensity cannot be neglected in real environments. The detection results in different simulated driving scenarios with 106 landmarks are shown in Figure 1. It is obvious that PFLD with 106 landmarks is robust in large poses but affected by mask coverings and changing illumination.

To develop the robustness of existing PFLD to locate eyes and monitor the state in real scenarios, we design more

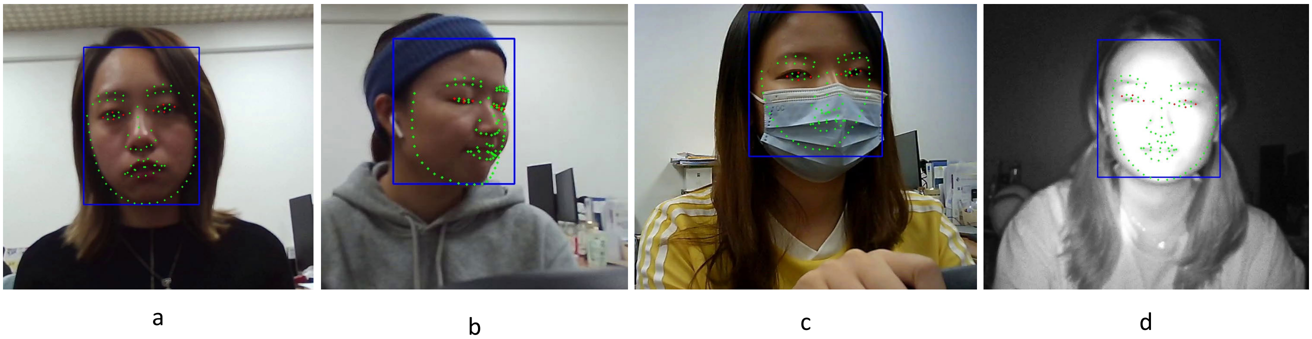


FIGURE 1. The detection results with 106 points in green and 8 eye key points in red. In (a) and (b), precise landmark location-based PFLD is applied to both frontal and large poses. In (c), both the 106 landmarks and 8 eye key points are deflected due to the mask covering. In (d), the location of the 106 landmarks is precise, but the 8 eyes key points are deflected because of the influence of illumination.

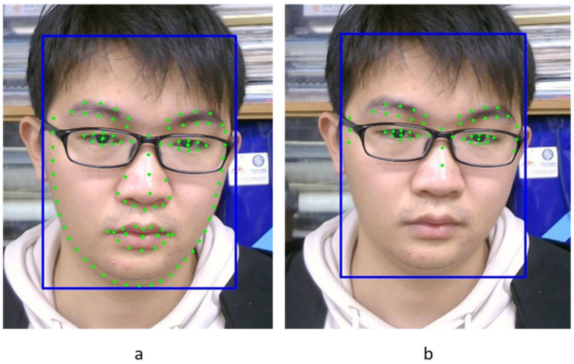


FIGURE 2. The 106 landmarks in (a), and the 46 landmarks based on 106 landmarks in (b).

TABLE 1. The main model structure of the PFLD-eyes.

Input	Operator	t	c	n	s
$112^2 \times 3$	Conv3 × 3	-	64	1	2
$56^2 \times 64$	Depthwise Conv3 × 3	-	64	1	1
$56^2 \times 64$	Bottleneck	2	64	5	2
$28^2 \times 64$	Bottleneck	2	128	1	2
$14^2 \times 128$	Bottleneck	4	128	6	1
$14^2 \times 128$	Bottleneck	2	16	1	1
(s1)$14^2 \times 16$	Conv3 × 3	-	32	1	2
(s2) $7^2 \times 32$	Conv7 × 7	-	128	1	1
(s3) $1^2 \times 128$	-	-	128	1	-
s1, s2, s3	Full Connection	-	92	1	-

The t represents the expansion factor, c represents the dimensionality of outputs, n represents the number of repetitions of the operator, and s represents the stride.

reasonable face landmarks based on an existing dataset with 106 landmarks. The main structure of the developed PFLD is shown in table 1, and we call it PFLD-eyes. The landmark distributions of our PFLD-eyes and PFLD eyes are shown in Figure 2. Compared to PFLD, note that our PFLD-eyes decreases the number of landmarks to 46 landmarks based on the original 106 landmarks, that is, we could use existing datasets with only a little preprocessing. And we would train PFLD-eyes by the augmented dataset which is enlarged by the method proposed in the next section. We use the novel loss function in (2) to train PFLD-eyes. The robustness of our model would improve considerably due to the more reasonable face landmarks and augmented dataset, which will be verified in the next section.

B. AN EFFECTIVE METHOD FOR TRAINING DATASET AUGMENTATION

Although the existing algorithm is robust in large poses, it is not practical in real environments with mask coverings or changing illumination, as shown in Figure 1. Many research studies have demonstrated that the dataset and structure are crucial to the performance of deep learning models [49], [58], [59]. To the best of our knowledge, there are many public datasets for face alignment, but few of them specialize in our scenario. Moreover, it is difficult to annotate landmarks when faces are covered with masks or there are vague outlines due to poor illumination. To obtain a more robust model, we develop an effective method to expand datasets based on original datasets.

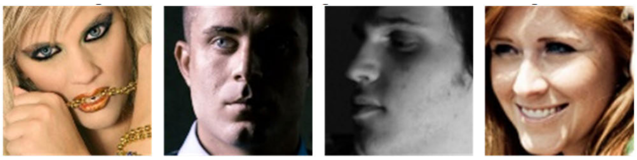


FIGURE 3. Examples of existing data sets for face landmark alignment.

Before introducing our proposed algorithm, it is necessary for us to review the challenge of obtaining precise landmarks. Examples of existing datasets for face landmark alignment are shown in Figure 3, and the video of driving simulations and real driving scenarios captured by common infrared cameras are shown in Figure 4. Compared to the images of the open dataset, the face outline is blurrier due to poor illumination, and facial features such as the nose and mouth are covered due to the mask. Thus, to develop the performance of our model in real scenarios, it is significant for us to generated similar images based on existing datasets.

1) CHANGING BRIGHTNESS AND ILLUMINATION

Illumination has a non-negligible influence on landmark location, and the outline of the face is blurrier due to poor and

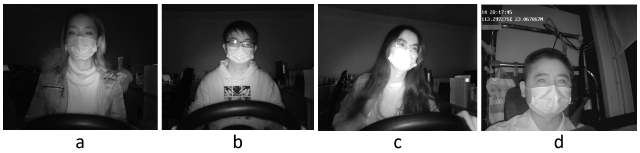


FIGURE 4. Examples of real videos captured in a driving simulation (a), (b), and (c) and a real driving scenario (d).

uneven illumination. Infrared cameras are widely used in cars because they do not disturb drivers at night. An infrared camera has almost the same imaging effect as a normal full-color camera in the daytime; however, because of poor illumination, the image is monochromatic at night. Moreover, the compensatory infrared source diverges due to distance constraints. Thus, an infrared camera has a focus light, and the light intensity gradually decreases around it, as shown in Figure 4. To add the same effect, we propose an effective method to add the illumination interference factor to existing images. The core algorithm is shown in Algorithm 1.

LD_FL represents the focus light landmark on the face, which we select randomly. **AL_add** represents the bright-ness deviation value between target image ***I'*** and original image ***I***. **get_distance** is a function to calculate the distance between each pixel landmark of the image ***I*** and the landmarks of focus light **LD_FL**. Let us define $p(x_0, y_0)$ as the pixel landmark of the image ***I*** and $p(x_1, y_1)$ as the landmark of the focus light **LD_FL**. We can calculate the distance as:

$$dis_{ij} = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \tag{3}$$

where dis_{ij} represents the distance between each pixel (i, j) and the landmarks of focus light **LD_FL**.

Algorithm 1 Changing_Image_Brightness_and_Illumination

```
Input: Original image I, the landmark of focus light LD_FL, the brightness deviation value AL_add
Output: Generated image I'
1: rows ← I. Height, cols ← I. Weight, channel ← I. Depth
2: I' = I. Copy ()
3: for i ← 1 to rows do
4:   for j ← 1 to cols do
5:     for k ← 1 to channel do
6:       p ← (i, j)
7:       dis ← get_distance ( p, LD_FL)
8:       if dis == 0 then
9:         rate ← 1
10:      else
11:        rate ← 1/dis
12:      color = (I [i, j] [k] + AL_add) * rate * a + b
13:      if color > 255 then
14:        color ← 255
15:      else if color < 0 then
16:        color ← 0
17:      I' [i, j] [k]= color
18: Return I'
```

As shown in Algorithm 1, the **LD_FL** represent the coordinates of light focus and the dis (Algorithm 1, line 7) represent the Euclidean distance between the **LD_FL** with each point p (Algorithm 1, line 6), the rate (Algorithm 1, line 11) is inversely proportional to dis, the adjusted color value (Algorithm 1, line 12) is proportionate to rate. Thus, when the point p is overlapped to **LD_FL**, it has the maximal color value which is limited to 255, as the dis increase, the color value of point p would diminish. After traversing all of the points in original image ***I*** (Algorithm 1, line 1), we would obtain a new image ***I'*** (Algorithm 1, line 18) with diminishing color value around the **LD_FL**.

2) ADD "MASK" TO FACE

With the outbreak of COVID-19, wearing masks is necessary to prevent the spread of the virus. For example, many face characteristics, such as the nose and mouth, would be covered due to masks. There is no special face alignment dataset for faces with masks to the best of our knowledge. To develop the performance of our proposed method in different scenarios, we artificially add a "mask" to the faces based on the existing dataset.

There are 106 landmarks for each face in the existing dataset, and the locations of the 106 landmarks are shown in appendix A. It is obvious that the 106 landmarks contain almost all facial structures and features. We define **FMC** $= \{(x_i, y_i)\}, i \in$ (11, 12, 13, 14, 15, 16, 2, 3, 4, 5, 6, 7, 8, 0, 24, 23, 22, 21, 20, 19, 18, 32, 31, 30, 29, 28, 27, 74) to represent the bottom part of the face, and the area encircled by **FMC** is a closed region that covers the mouth and nose. Based on this, we propose an effective method to add a "mask" to faces at night. The core procedure is shown in Algorithm 2.

There are 13 steps to obtain the face "mask" based on 106 landmarks. **Judge_in_polygon** is a function that judges whether the given point is in a polygon. The detailed implementation procedure of **Judge_in_polygon** is shown in appendix B.

Algorithm 2 Add_Mask_to_Face

```
Input: Original image I, the bottom half of the face contour landmarks set FMC
Output: Generated image I'
1: rows ← I. Height, cols ← I. Weight, channel ← I. Depth
2: I' ← I. Copy ()
3: for i ← 1 to rows do
4:   for j ← 1 to cols do
5:     for k ← 1 to channel do
6:       p ← (i, j)
7:       Is_in ← Judge_in_polygon (p, FMC)
8:       if Is_in then
9:         color ← 255
10:      else
11:        color ← I [i, j] [k]
12:      I' [i, j] [k]= color
13: Return I'
```

As shown in Algorithm 2, the p (Algorithm 2, line 6) presents the coordinates of each point. If p is in the region which is encircled by **FMC**, the color value of p would be adjusted to 255, otherwise, the color value of p would remain unchanged. Note that the region that is encircled by **FMC** has a similar shape to a real mask. After traversing all of the points in original image ***I*** (Algorithm 2, line 1), we would add a "mask" to the original image ***I***.

3) MAIN PROCESS OF THE PROPOSED METHOD

In this section, we introduce the entire procedure to generate new images based on our proposed method. The 106 landmark locations of the existing dataset per image are shown in appendix A. Let (x, y) denote a landmark point on image ***I***. **FL** represents the 106 landmarks of the face in image ***I***. **FBC** $= \{(x_i, y_i)\}, i \in (1, 16, 0, 32, 17, 104, 49)$ represents the basic face contour point of image ***I***. **FMC** $= \{(x_i, y_i)\}, i(11, 12, 13, 14, 15, 16, 2, 3, 4, 5, 6, 7, 8, 0, 24, 23, 22, 21, 20, 19, 18, 32, 31, 30, 29, 28, 27, 74)$ represents the bottom part of the face. The overall algorithm is shown in Algorithm 3. As showed in Algorithm 3, There are 8 steps to generate a new image:

(1) Transform the original BGR image ***I*** to gray image ***I'***, where BGR_TO_GRAY is a function of OpenCV.[1]

(2) Obtain the average brightness (**AL**) of the face in image ***I'***, and the face is the region encircled by **FBC**. **Get_average_brightness** is a function that is used to calculate the average brightness of the restricted area in the image. The detailed implementation procedure of **Get_average_brightness** is shown in appendix C.

(3) Obtain the face brightness deviation value (**AL_add**) between the brightness of the target image and the brightness of the original image. The brightness of the target image is a random number that ranges from 120 to 170.

(4) Randomly obtain the landmarks of focus light (**LD_FL**). Because of the distance constraint, the infrared light is divergent rather than parallel. Selecting an **LD_FL** could improve the quality of the image.

(5) Change the brightness and illumination of ***I'***. **Changing_image_brightness_and_illumination** is a function that is used to change the image brightness and Illumination by the given **LD_FL** and **AL_add**. The randomly selected **LD_FL**is the light focus of the face, and the light intensity gradually decreases around it. The detailed implementation procedure of **Changing_image_brightness_and_illumination** is shown in algorithm 1.

(6) Add "mask" to face. A "mask" is added using the encircled region of the given **FMC**. **Add_mask_to_face** is a function that is used to add a mask to the restricted area in the image. In algorithm 2, we list its detailed implementation procedure.

(7) Smooth image ***I'*** using a Gaussian filter, which is a function of OpenCV.

(8) Obtain the processed image ***I'***.

[1] OpenCV is a public library that is widely used in image processing.

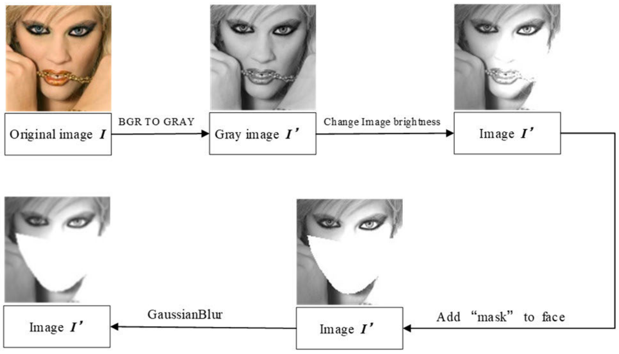


FIGURE 5. The processing procedure of an image.

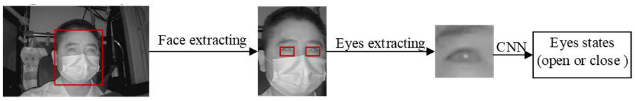


FIGURE 6. Existing methods for eyes state evaluation.

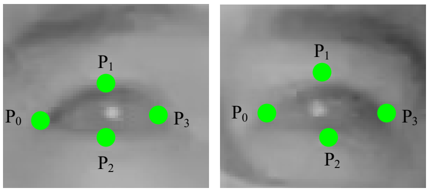


FIGURE 7. The 4 points around each eye.

The image processing procedure is shown in Figure 5. Compared to the original image, the generated image is covered by a "mask", and the outline of the face is blurrier, which is similar to captured images at night.

C. EYES STATE EVALUATION WITH EAR

Eye state assessment is of great importance to driver fatigue detection. Most visual-based fatigue measurements, e.g., blink frequency, PERCLOS, etc., are based on the analysis of the eye state. Many existing methods to measure the eye states have been designed, as shown in Figure 6. Three steps are included: (1) Locating the face in a given image, (2) Extracting the eye region, and (3) Obtaining the eye state via a convolutional neural network (CNN) model. Although it is a feasible method for eye state evaluation, using an extra CNN model to measure the state of eyes would requires considerable time and computing resources. Moreover, the degree of eye closure cannot be obtained.

To develop the detection time and obtain the degree of eye closure. We introduce the Eye Aspect Ratio (EAR) as a measurement of eye state. Based on the 46 landmarks we obtained in section A, we select 4 landmarks around each eye to measure the state of the eyes, as shown in Figure 7.

The EAR can be calculated as followed:

$$EAR = \frac{||p_1 - p_2||}{||p_0 - p_3||} \tag{4}$$

Algorithm 3 Image Generate Frame

```
Input: Original image $I$, 106 face landmark set FL, face basic contour landmark set FBC, the bottom half of the face contour landmark set FMC.
Output: Generated image $I'$
$I' \leftarrow$ BGR_TO_GRAY($I$)     ◃ Transform original BGR image $I$ to gray image $I'$
AL ← $G$ et_average_brightness ($I'$, FBC)     ◃ Obtain average brightness of face in image $I'$ depending on the basic facial contour that is formed by FBC.
AL_add ← random. radiant (120,170)- AL     ◃ Obtain face brightness deviation value AL_add between the brightness of the target image and brightness of the original image AL
LD_FL ← FL ( random. radiant (0,60))     ◃ Randomly select a face landmark LD_FL.
$I' \leftarrow$ $C$ hange_image_brightness_ and_illumination ($I'$, LD_FL, AL_add)     ◃ Change the brightness of image $I'$
$I' \leftarrow$ Add_mask_to_face ($I'$, FMC)     ◃ Add a 'mask' to the face in image $I'$
$I' \leftarrow$ GaussianBlur($I'$)     ◃ Smoothing $I'$ with a Gaussian filter
Return $I'$     ◃ Obtain generated image $I'$
```

where p_0, p_1, p_2, and p_3 represent the landmarks of eye key points in Figure 7. It is obvious that the EAR is sensitive to the eye state and that the EAR decreases gradually as we close our eyes.

III. EXPERIMENTS

In this section, we first introduce our data set and experimental environment. Then, we evaluate the performance of the proposed algorithms from different aspects.

A. DATA SET AND ENVIRONMENT

We use the open-source data set as our training set.[2] It contains 21080 faces, and there are 106 landmarks in each face, as shown in appendix A. To evaluate the performance of our model on the landmarks of eye key points, we develop a specialized driving simulation dataset for eye key point detection (EKPDD), which contains 815 images including 15 different subjects, and each image was artificially labeled with 8 eye key points. There were 4 different simulated driving scenarios, including daytime driving with a mask, daytime driving without a mask, night driving with a mask, and night driving without a mask. The examples of the EKPDD data set are shown in Figure 8. BioID [15] is a public database that is widely used in eye state evaluation. It contains 1521 gray images, and it is challenging due to its various illuminations and large poses. To further verify the robustness of eye state evaluation, we also test our model on EKPDD.

Our experimental platform is an Intel Core i5-8500 (main frequency: 3.0 GHz) with the x86 architecture, GTX 1050ti (CUDA: 10.0 and CUDNN: 7.4) with the Pascal architecture, 16 GB of DDR4 memory, the opencv4.3.0 image library, and the TensorFlow 1.13.0 deep learning computing framework.

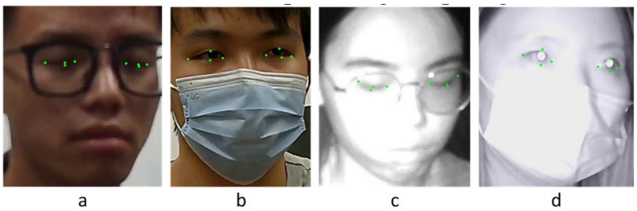


FIGURE 8. Examples of eye key point detection data set (EKPDD). Four different simulated driving scenarios are included: daytime driving without a mask in (a), daytime driving with a mask in (b), night driving without a mask in (c), and night driving without a mask in (d).

TABLE 2. The MNE of different models.

	Dlib Toolkit	PFLD	PFLD-eyes
Daytime without mask	4.75	**3.71**	4.20
Daytime with mask	6.91	6.42	**5.90**
Night without mask	6.45	5.45	**5.20**
Night with mask	10.45	9.79	**8.85**
all	6.86	6.07	**5.83**

There are 4 different scenarios in the EKPDD data set. There are 241, 228,179, and 167 images in each scenario, respectively. Some examples are shown in Figure 8.

B. EYES KEY POINTS LOCATION

1) EVALUATION OF PFLD-EYES

a: EVALUATION INDICATORS

In this section, we evaluate the location performance of the improved model on the eye key point detection data set (EKPDD). The interocular distance normalized error (ION) and the mean normalized error (MNE) are important measurements of landmark locations. ION is the ratio of the

[2] https://github.com/JACKYLUO1991/106-landmarks-dataset

distance between ground truth landmarks and predicted landmarks to the distance between the outer corners. It can be computed as follow:

$$ION_i = \frac{||x_{pre_i} - x_{gt_i}||_2}{d_{iod}} \tag{5}$$

where x_{pre_i} and x_{gt_i} denote the predicted landmarks and ground truth landmarks, respectively; and d_{iod} represents the distance between the outer corners. The MNE is the average ION of facial landmarks and is calculated as follows:

$$MNE = \frac{\sum_{i=0}^{N} ION_i}{N} \tag{6}$$

where N denotes the total number of face landmarks. MNE is an important parameter that is widely used to evaluate the average location error.

b: EVALUATION USING THE MNE

We pay more attention to the accuracy of locating the eyes; thus, different from many face alignment algorithms that average all face landmarks' IONs as MNEs, we select 8 eye key points, which are shown in Figure 7, to calculate the MNE. To verify the excellent performance of our proposed PFLD-eyes, we compare it using the original PFLD and OpenCV Dlib toolkit that is widely used in many research studies. Note that the eye key points predicted by the Dlib toolkit do not all overlap with the 8 manually labeled eye key points in EKPDD. Therefore, we selected 4 overlapping points to calculate the MNE. We train PFLD and PFLD-eyes by the original data set,[3] and we evaluate our model on EKPDD. The results are shown in table 2.

As shown in table 2, compared to the OpenCV Dlib toolkit, the PFLD has a better performance in all scenarios. This is because the PFLD is more robust in larger drive poses due to the novel loss function. Compared to PFLD, our developed PFLD-eyes has poor performance in the first scenario but better performance in other scenarios, especially for the scenarios where the driver wears a mask. It is obvious that PFLD-eyes has almost the same performance as PFLD when the driver does not wear a mask; however, PFLD-eyes achieves a prominent improvement in eye key point location compared to PFLD when there is a mask covering the driver's face. This is because 46 landmarks could reduce the characteristic dependence of the lower part of the face and would reduce the interference of the covering compared to 106 landmarks.

c: EVALUATION ON SPEED

The speeds of different models are shown in table 3. It is obvious that the OpenCV Dlib Toolkit has the best FPS of approximately 502; however, the frequently used camera has an image capture speed of approximately 30 fps. Therefore, both PFLD and PFLD-eyes are fast enough to conduct real-time detection.

TABLE 3. The detection speed of different models.

	Dlib Toolkit	PFLD	PFLD-eyes
Speed (fps)	**502**	252	262

TABLE 4. The MNE of different models.

	PFLD-eyes (Original data set)	PFLD-eyes (Extended data set)
Daytime without mask	4.20	**4.18**
Daytime with mask	5.90	**5.05**
Night without mask	5.20	**5.10**
Night with mask	8.85	**5.79**
all	5.83	**4.95**

TABLE 5. The comparison of different models on BioID database.

Method	Recall	Precision	F1	Accuracy
Ö.F.Söylemez	94.50%	97.81%	96.10%	93.30%
Cheng	88.58%	98.00%	93.05%	94.00%
Ours	**98.58%**	**99.45%**	**99.01%**	**97.70%**

TABLE 6. The comparison of model on EKPDD database.

	Recall	Precision	F1	Accuracy
Daytime without mask	95.42%	96.15%	95.78%	95.50%
Daytime with mask	94.27%	96.73%	95.48%	94.00%
Night without mask	95.65%	88.89%	93.10%	91.90%
Night with mask	98.25%	92.56%	95.32%	93.50%
All	95.74%	94.03%	94.88%	93.90%

2) EVALUATION ON DATA SET AUGMENT

a: EVALUATION INDICATOR

To verify the effectiveness of the data set augment method, each image in the original data set is processed by the method proposed in section II-B, and the augmented data set is doubled to the original data set. Some examples of original images to processed images are shown in Figure 9. We train our PFLD-eyes model using the original data set and extended data set, respectively. In order to evaluate the performance of the models in the simulated driving environment, we calculate the MNE on EKPDD.

b: EVALUATION ON MNE

The decrease of the MNE using 46 facial landmarks with the number of epochs is shown in Figure 10. We set the initial learning rate is 0.0001 and decrease it progressively at 0, 200, and 400 epochs. The final MNE of PFLD-eyes (extended datasets) is 4.9 whereas the MNE of PFLD-eyes (original datasets) is 5.3. It is obvious that the extended datasets could

[3] https://github.com/JACKYLUO1991/106-landmarks-dataset

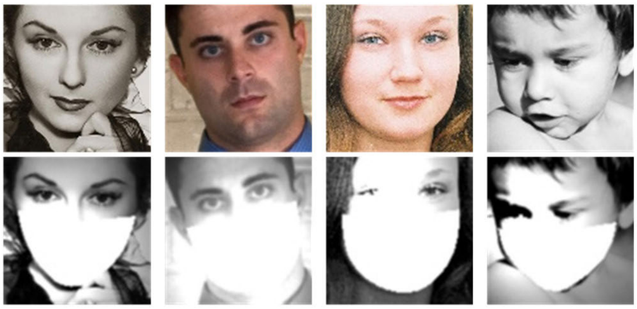


FIGURE 9. Some examples of images processed from the original dataset. The top image is the original image and the bottom image is the processed image.

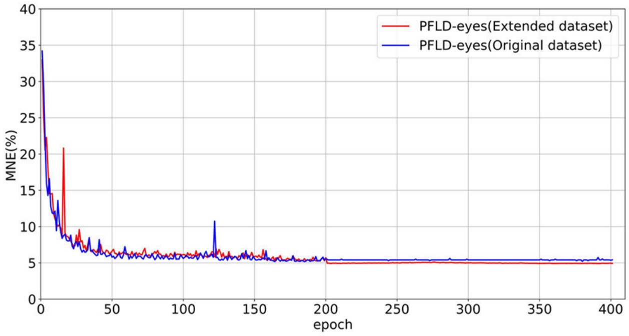


FIGURE 10. The decrease in the MNE of 46 facial landmarks with the number of epochs.

improve the performance of PFLD-eyes. In order to assess the measurements of models in a driving environment, we test our models on EKPDD. The results are shown in table 4. The PFLD-eyes trained by the extended data set has better performance than PFLD-eyes (original data set). In particular, there was a significant improvement in the scenario when the drivers wore a mask at night, and the results were 5.79 and 8.85, respectively. It is obvious that the method used to enlarge the original data improves the robustness of our model for mask coverings and changing illuminance, and it would increase the costs when adding new images or human annotation.

C. EYE STATE EVALUATION

1) EVALUATION INDICATOR

To evaluate the performance of our model in eye state, we tested it on BioID and EKPDD, respectively. We use the EAR as the eye state evaluation parameter, and we set 0.2 as the threshold value for binary eye state evaluation which is used in many researches. This means that if the EAR is greater than 0.2, the eyes are open; otherwise, the eyes are closed. Let *TP* represents the number of times where the predicted eye state is True and the ground truth eye state is also True, FN represents the number of times where the predicted eye state is False whereas the ground truth eye state is True, FP represents the number of times where the predicted eye state is True whereas the ground truth eye state is False, TN represents the number of times where the predicted eye state is False and the ground truth eye state is also False.

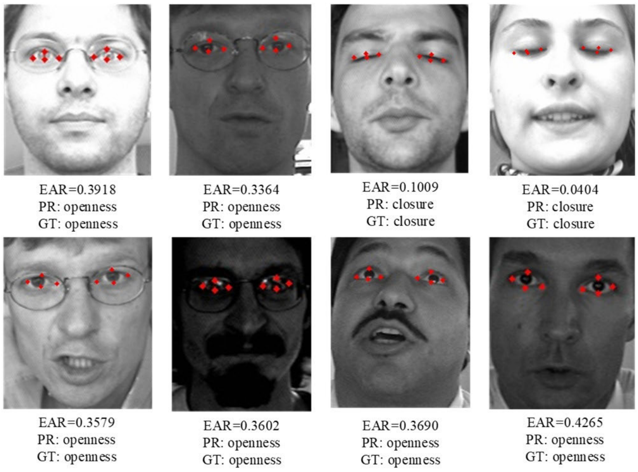


FIGURE 11. Some examples of the detection results on the BioID dataset. PR represents the predicted results based on the EAR, and GT represents ground truth.

We can calculate the evaluation indicators as follow:

$$Accuracy = \frac{TP + TN}{(TP + FN + FP + TN)}\% \tag{7}$$

$$Pr\,ecision = \frac{TP}{(TP + FP)}\% \tag{8}$$

$$\mathrm{Recall} = \frac{TP}{(TP + FN)}\% \tag{9}$$

$$F1 = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Recall}}{(\mathrm{Pr}\,ecision + \mathrm{Recall})} \tag{10}$$

where Accuracy, Precision, and Recall evaluate the performance of a model in different aspects and F1 is a comprehensive indicator that is calculated by Precision and Recall.

2) TEST ON BIOID

We compare the performances of different models in eye state evaluation on BioID. The results are shown in table 5, and some examples of the detection results are shown in Figure 11.

As shown in table 5, compared to the proposed method, Cheng [39], and Sö ylemez and Ergen [60], our model obtains satisfactory results with significant improvements in all indicators. Though the BioID is challenging due to its various illuminations and large poses, we obtain satisfactory results on Precision and Recall metrics, which means our model has low missing rates and misjudgment rates on eye state detection. The good performance on F1 also means our model has a good balance between Recall and Precision. our model also has prominent performance on accuracy with a value of about 97.70%. As shown in Figure 11, though poor illuminations and large head pose, the detection results of our model are reliable and accurate.

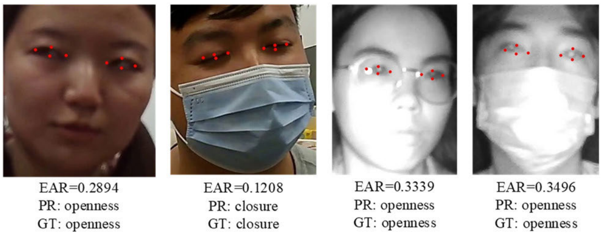


FIGURE 12. Examples of detection results on the EKPDD dataset. PR represents the predicted results based on the EAR, and GT represents the Ground Truth.

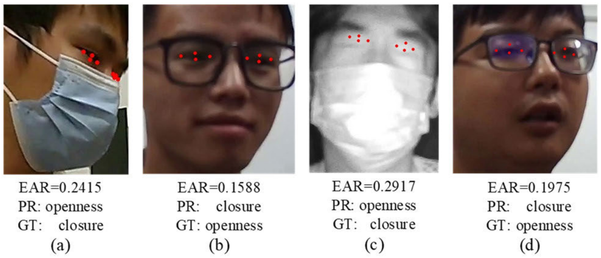


FIGURE 13. Eye state estimation examples of failures on the testing database. PR represents the predicted results based on the EAR, and GT represents the Ground Truth.

3) TEST ON EKPDD

To further obtain the performance of our proposed model in a real driving scenario, we test it on EKPDD. Compared to the BioID data set, the images contained in EKPDD are more challenging with mask coverings, changing illumination from daytime to nighttime, and large head poses. There are four different scenarios including daytime driving with a mask, daytime driving without a mask, night driving with a mask, and night driving without a mask. We label the eye states manually and test our model in four different scenarios. The results are shown in table 6. Compared to BioID, although there were more challenges with mask coverings, illumination, and poses in EKPDD. The performance of our model was satisfactory with an accuracy of approximately 93.9% on the whole data set, and the model was robust in all scenarios with little difference in metrics. Some examples of detection results are shown in Figure 12.

We further analyze the failed samples in the test data set, as shown in Figure 13. We find that some detections failed due to the large poses of the head, and the right eyes were out of sight, as shown in (a). Some detections failed due to the influence of a thick eyelid together with poor illumination, which results in it being difficult to recognize the state of eyes, as shown in (b). Some samples failed because the images were too seriously blurred to recognize the eye region, as shown in (c). In (d), the detection failed due to the strong reflection of the glasses. Most of them failed due to severe interference.

Generally, our model achieves satisfactory results for eye state evaluation on both BioID and EKPDD. EKPDD contains images in a simulated driving environment with different scenarios, which are close to real conditions. The preeminent performance for different evaluation indicators such as Accuracy, Precision, Recall, etc. in different datasets means that our model is precise and robust for eye state measure. Based on the detection results, PERCLOS, blink frequency and other fatigue parameters that are appropriate for fatigue detection could be calculated.

IV. CONCLUSION

Eye location and eye state evaluation are crucial to fatigue detection. With the outbreak of COVID-19, masks are essential to protect bus drivers from virus infection. To develop the accuracy of eye location and state evaluation based on facial landmarks in complicated environments, we proposed a more robust model in this paper. First, we develop an existing lightweight facial landmark model that is robust to large poses. Then, we propose a method to augment the training data set based on the original landmark data set to improve the performance of our model in real driving scenarios, such as scenarios with mask coverings and changing illumination. The experiment shows that the proposed model obtains better performance in eye key point locations on a driving simulation data set. Finally, we introduce the EAR to classify the eye states. The experiment shows that our model obtains satisfactory performance on the BioID data set with an accuracy of approximately 97.7%, and further testing on EKPDD achieves a satisfactory result with an average accuracy of approximately 93.9%.

In the future, we will obtain fatigue parameters based on reliable eye state detection and develop a driver fatigue detection system.

APPENDIX A

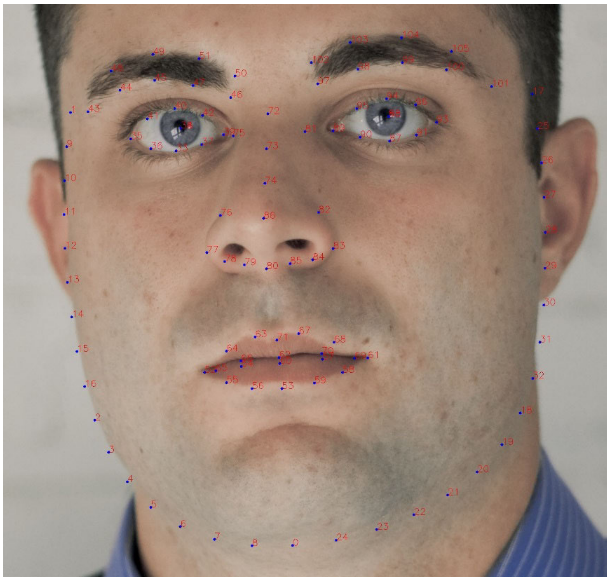


FIGURE 14. The locations of the 106 landmarks.

APPENDIX B

See Algorithm 4.

Algorithm 4 Judge_in_Polygon

```
Input:  the landmark of p (x, y), the contour landmark set of polygon ((x_1 , y_1 ), (x_2 , y_2 ), ...)
Output: is_in_polygon
px ← p.x, py ← p.y, is_in_polygon ← False, k ← FMC.length, i ← 0, j ← k − 1
for i ← 0 to k do
sx ← polygon[i].x, sy ← polygon[i].y, tx ← polygon[j].x, ty ← polygon[j].y
if (sx == px and sy == py) or (tx == px and ty == py) then
    Return True
if (sy < py and ty >= py) or (sy >= py and ty < py) then
    x = sx + (py - sy) * (tx - sx)/(ty - sy)
    if x == px then
      Return True
      if x > px then
        is_in_polygon = True
j ← i, i ←i + 1
Return is_in_polygon
```

Algorithm 5 Get_Average_Brightness

```
Input: Original image I, the face basic contour landmark set FBC,
Output: image_average_brightness
rows ← I. Height, cols ← I. Weight, channel ← I. Depth, color ←0, m ←0
for i ← 1 to rows do
for j ← 1 to cols do
   for k ← 1 to channel do
     p ← (i, j)
     is_in_ polygon = Judge_in_polygon (p, FBC)
     if is_in_ polygon then
      color = I [i, j] [k] + color
      m = m + 1
image_average_brightness = color/m
Return image_average_brightness
```

APPENDIX C

See Algorithm 5.

ACKNOWLEDGMENT

Thanks to the open-source dataset, 106 face land-marks for providing data. https://github.com/JA-CKYLUO1991/106-landmarks-dataset.

REFERENCES

[1] J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1139–1152, Mar. 2014.

[2] Z. Ye, Y. Li, A. Fathi, Y. Han, A. Rozga, G. D. Abowd, and J. M. Rehg, "Detecting eye contact using wearable eye-tracking glasses," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, Sep. 2012, pp. 699–704.

[3] F. You, X. Li, Y. Gong, H. Wang, and H. Li, "A real-time driving drowsiness detection algorithm with individual differences consideration," *IEEE Access*, vol. 7, pp. 179396–179408, 2019.

[4] X. Li, X. Lian, and F. Liu, "Rear-end road crash characteristics analysis based on Chinese in-depth crash study data," in *Proc. CICTP*, Jul. 2016, pp. 1536–1545.

[5] A. Williamson, D. A. Lombardi, S. Folkard, J. Stutts, T. K. Courtney, and J. L. Connor, "The link between fatigue and safety," *Accident Anal. Prevention*, vol. 43, no. 2, pp. 498–515, Mar. 2011.

[6] A. Amodio, M. Ermidoro, D. Maggi, S. Formentin, and S. M. Savaresi, "Automatic detection of driver impairment based on pupillary light reflex," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3038–3048, Aug. 2019.

[7] *Drowsy Driving 2015, A Brief Statistical Summary*, Nat. Center Health Statist., Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Oct. 2017, p. 2.

[8] S. Kar, M. Bhagat, and A. Routray, "EEG signal analysis for the assessment and quantification of driver's fatigue," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 13, no. 5, pp. 297–306, Sep. 2010.

[9] N. Galley, "Blink Parameter as indicator of drivers sleepiness," *Nursing Standard*, vol. 23, no. 35, pp. 7–26, 2003.

[10] J. Ahmed, J.-P. Li, S. A. Khan, and R. A. Shaikh, "Eye behaviour based drowsiness detection system," in *Proc. 12th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2015, pp. 268–272.

[11] X.-Q. Luo, R. Hu, and T.-E. Fan, "The driver fatigue monitoring system based on face recognition technology," in *Proc. 4th Int. Conf. Intell. Control Inf. Process. (ICICIP)*, Jun. 2013, pp. 384–388.

[12] K. Li, Y. Gong, and Z. Ren, "A fatigue driving detection algorithm based on facial multi-feature fusion," *IEEE Access*, vol. 8, pp. 101244–101259, 2020.

[13] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 99–111, Aug. 1992.

[14] M. Nixon, "Eye spacing measurement for facial recognition," *Proc. SPIE*, vol. 575, pp. 279–285, Dec. 1985.

[15] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the Hausdorff distance," in *Audio- and Video-Based Biometric Person Authentication* (Lecture Notes in Computer Science), vol. 2091. Berlin, Germany: Springer, 2001, pp. 90–95.

[16] W.-B. Horng, C.-Y. Chen, Y. Chang, and C.-H. Fan, "Driver fatigue detection based on eye tracking and dynamic template matching," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Mar. 2004, pp. 7–12.

[17] E. Skodras and N. Fakotakis, "Precise localization of eye centers in low resolution color images," *Image Vis. Comput.*, vol. 36, pp. 51–60, Apr. 2015.

[18] D. Monzo, A. Albiol, J. Sastre, and A. Albiol, "Precise eye localization using HOG descriptors," *Mach. Vis. Appl.*, vol. 22, no. 3, pp. 471–480, May 2010.

[19] J. Song, Z. Chi, and J. Liu, "A robust eye detection method using combined binary edge and intensity information," *Pattern Recognit.*, vol. 39, no. 6, pp. 1110–1125, Jun. 2006.

[20] S. A. Sirohey and A. Rosenfeld, "Eye detection in a face image using linear and nonlinear filters," *Pattern Recognit.*, vol. 34, no. 7, pp. 1367–1391, Jan. 2001.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[22] X. Tang, Z. Ou, T. Su, H. Sun, and P. Zhao, "Robust precise eye location by adaboost and SVM techniques," in *Advances in Neural Networks* (Lecture Notes in Computer Science), vol. 3497. Berlin, Germany: Springer, 2005, pp. 93–98.

[23] L. Jin, X. Yuan, S. Satoh, J. Li, and L. Xia, "A hybrid classifier for precise and robust eye detection," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2006, pp. 731–735.

[24] Pentland, Moghaddam, and Starner, "View-based and modular eigenspaces for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 84–91.

[25] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[26] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[27] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[28] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 259–289, May 2008.

[29] X. Tan, F. Song, Z.-H. Zhou, and S. Chen, "Enhanced pictorial structures for precise eye localization under incontrolled conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1621–1628.

[30] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3476–3483.

[31] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 386–391.

[32] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[33] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2034–2043.

[34] Z.-T. Liu, S.-H. Li, M. Wu, W.-H. Cao, M. Hao, and L.-B. Xian, "Eye localization based on weight binarization cascade convolution neural network," *Neurocomputing*, vol. 378, pp. 45–53, Feb. 2020.

[35] H. Liu, Y. Wu, and H. Zha, "Eye state detection from color facial image sequence," in *Proc. 2nd Int. Conf. Image Graph.*, Jul. 2002, p. 693.

[36] Q. Wang and J. Yang, "Eye detection in facial images with unconstrained background," *J. Pattern Recognit. Res.*, vol. 1, no. 1, pp. 55–62, 2006.

[37] Y. Du, P. Ma, X. Su, and Y. Zhang, "Driver fatigue detection based on eye state analysis," in *Proc. 11th Joint Int. Conf. Inf. Sci.*, 2008, pp. 1–6.

[38] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognit.*, vol. 47, no. 9, pp. 2825–2838, Sep. 2014.

[39] E. Cheng, B. Kong, R. Hu, and F. Zheng, "Eye state detection in facial image based on linear prediction error of wavelet coefficients," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Feb. 2009, pp. 1388–1392.

[40] R. Sanyal and K. Chakrabarty, "Two stream deep convolutional neural network for eye state recognition and blink detection," in *Proc. 3rd Int. Conf. Electron., Mater. Eng. Nano-Technol. (IEMENTech)*, Aug. 2019, pp. 1–8.

[41] A. Zadeh, Y. C. Lim, T. Baltrušaitis, and L.-P. Morency, "Convolutional experts constrained local model for 3D facial landmark detection," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2519–2528.

[42] F. Zhang, J. Su, L. Geng, and Z. Xiao, "Driver fatigue detection based on eye state recognition," in *Proc. Int. Conf. Mach. Vis. Inf. Technol. (CMVIT)*, Feb. 2017, pp. 105–110.

[43] R. Huang, Y. Wang, and L. Guo, "P-FDCN based eye state analysis for fatigue detection," in *Proc. IEEE 18th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2018, pp. 1174–1178.

[44] W. Deng and R. Wu, "Real-time driver-drowsiness detection system using facial features," *IEEE Access*, vol. 7, pp. 118727–118738, 2019.

[45] Y. Sun, P. Yan, Z. Li, J. Zou, and D. Hong, "Driver fatigue detection system based on colored and infrared eye features fusion," *Comput., Mater. Continua*, vol. 63, no. 3, pp. 1563–1574, 2020.

[46] Q. Cheng, W. Wang, X. Jiang, S. Hou, and Y. Qin, "Assessment of driver mental fatigue using facial landmarks," *IEEE Access*, vol. 7, pp. 150423–150434, 2019.

[47] M. E. Colak and A. Varol, "Easymatch-an eye localization method for frontal face images using facial landmarks," *Tehnički Vjesnik*, vol. 27, no. 1, pp. 205–212, 2020.

[48] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Jan. 2009.

[49] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[50] J. Munoz-Bulnes, C. Fernandez, I. Parra, D. Fernandez-Llorca, and M. A. Sotelo, "Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 366–371.

[51] G. Li, Y. Yang, X. Qu, D. Cao, and K. Li, "A deep learning based image enhancement approach for autonomous driving at night," *Knowl.-Based Syst.*, vol. 213, Feb. 2021, Art. no. 106617.

[52] G. Li, Y. Yang, and X. Qu, "Deep learning approaches on pedestrian detection in hazy weather," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8889–8899, Oct. 2020.

[53] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 3, Nov. 2014, pp. 2672–2680.

[54] H. Lee, M. Ra, and W.-Y. Kim, "Nighttime data augmentation using GAN for improving blind-spot detection," *IEEE Access*, vol. 8, pp. 48049–48059, 2020.

[55] N. Soufi and M. Valdenegro-Toro, "Data augmentation with symbolic-to-real image translation GANs for traffic sign recognition," 2019, *arXiv:1907.12902*. [Online]. Available: https://arxiv.org/abs/1907.12902

[56] G. Li, Y. Lin, and X. Qu, "An infrared and visible image fusion method based on multi-scale transformation and norm optimization," *Inf. Fusion*, vol. 71, pp. 109–129, Jul. 2021.

[57] X. Guo, S. Li, J. Yu, J. Zhang, J. Ma, L. Ma, W. Liu, and H. Ling, "PFLD: A practical facial landmark detector," 2019, *arXiv:1902.10859*. [Online]. Available: https://arxiv.org/abs/1902.10859

[58] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 843–852.

[59] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 8–12, Mar. 2009.

[60] F. Söylemez and B. Ergen, "Eye location and eye state detection in facial images using circular Hough transform," in *Computer Information Systems and Industrial Management* (Lecture Notes in Computer Science), vol. 8104. Berlin, Germany: Springer, 2013, pp. 141–147.

YANCHENG LING received the B.E. degree in traffic engineering and Internet of Things engineering from East China Jiaotong University, Nanchang, China, in 2017, and the M.S. degree in traffic engineering from the South China University of Technology, Guangdong, China, in 2019, where he is currently pursuing the Ph.D. degree in traffic information engineering and control.

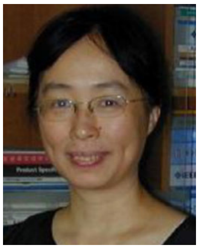
Since 2017, his research interests include mobile phone information collection and processing, ITS (intelligent transportation system), computer vision, video analysis, and objection detection.

RUIFA LUO received the B.S. degree from the South China University of Technology, Guangzhou, China, in 1998, and the M.S. degree from Sun Yat-sen University, Guangzhou, in 2012. He is currently pursuing the Ph.D. degree in transportation information engineering with Tongji University.

He is also a Pioneer, a Leader, and a National Standard Contributor of ETC Industry and a Main Promotor of ITS, China. He is also the Founder and the Chairman of Shenzhen Genvict Technologies Company Ltd. He is also a Senior Engineer leading talents in the National Ten Thousand Talent Program, Shenzhen Local-Level Talent, and Nanshan District High-Level Talent. He is also the Director of the Research and Development Center, Transport Industry of Key Technologies and Equipment for Intelligent Vehicle-Infrastructure Cooperative System, Ministry of Transport of China, and the Shenzhen Intelligent Transportation Industry Association. He is also a Committee Member of Shenzhen Federation of Industry and Commerce. He has over 20 years of experience in the ITS (intelligent transportation system) industry.

XIAOXIAN DONG received the B.S. degree in traffic transportation from Fuzhou University, Fuzhou, Fujian, China, in 2020. She is currently pursuing the master's degree with the South China University of Technology, Guangzhou, China.

Her research interests include behavior understanding, driving fatigue and detection, and bus priority scheduling.



XIAOXIONG WENG received the bachelor's degree in industrial automation from the Dalian University of Technology, China, the master's degree in automatic control theory and application from Shanghai Jiao Tong University, China, and the Ph.D. degree in control theory and control engineering from the South China University of Technology, China.

She is currently a Professor with the School of Civil Engineering and Transportation, SCUT. She is the author of two books and multiple inventions. She has repeatedly undertaken the design and consulting of government management departments for the Guangzhou Asian Games, Shenzhen Universiade, Guangzhou University, Urban Expressway System, Subway, Urban Public Transportation System, and other projects. Her research interests include intelligent transportation systems, computer vision, dynamic modeling of urban traffic flow, data mining on transit systems, traffic signal control systems, and transit commuter behavior analysis.