

Controller Fatigue State Detection Based on ES-DFNN

Haijun Liang *, Changyan Liu *, Kuanming Chen, Jianguo Kong, Qicong Han and Tiantian Zhao

College of Air Traffic Management, Civil Aviation Flight University of China, Guanghan 618307, China; chenkuanming@cafuc.edu.cn (K.C.); kongjianguo@cafuc.edu.cn (J.K.); hanqc@cafuc.edu.cn (Q.H.); zhaotiantian@cafuc.edu.cn (T.Z.)

* Correspondence: navyliang@cafuc.edu.cn (H.L.); liuchangyan@cafuc.edu.cn (C.L.)

Abstract: The fatiguing work of air traffic controllers inevitably threatens air traffic safety. Determining whether eyes are in an open or closed state is currently the main method for detecting fatigue in air traffic controllers. Here, an eye state recognition model based on deep-fusion neural networks is proposed for determination of the fatigue state of controllers. This method uses transfer learning strategies to pre-train deep neural networks and deep convolutional neural networks and performs network fusion at the decision-making layer. The fused network demonstrated an improved ability to classify the target domain dataset. First, a deep-cascaded neural network algorithm was used to realize face detection and eye positioning. Second, according to the eye selection mechanism, the pictures of the eyes to be tested were cropped and passed into the deep-fusion neural network to determine the eye state. Finally, the PERCLOS indicator was combined to detect the fatigue state of the controller. On the ZJU, CEW and ATCE datasets, the accuracy, F1 score and AUC values of different networks were compared, and, on the ZJU and CEW datasets, the recognition accuracy and AUC values among different methods were evaluated based on a comparative experiment. The experimental results show that the deep-fusion neural network model demonstrated better performance than the other assessed network models. When applied to the controller eye dataset, the recognition accuracy was 98.44%, and the recognition accuracy for the test video was 97.30%.

Keywords: air traffic control; fatigue detection; MTCNN; transfer learning; DFNN; eye selection; air traffic safety; artificial intelligence



Citation: Liang, H.; Liu, C.; Chen, K.; Kong, J.; Han, Q.; Zhao, T. Controller Fatigue State Detection Based on ES-DFNN. *Aerospace* **2021**, *8*, 383. <https://doi.org/10.3390/aerospace8120383>

Academic Editor: Alexei Sharpanskykh

Received: 1 October 2021
Accepted: 29 November 2021
Published: 7 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction and Background

With the rapid development of the civil aviation industry, there has been an increase in the number of routes and aircraft sorties, the complexity of the sector, and the air traffic controller workload, and thus on-job fatigue is becoming a major issue affecting the safety of civil aviation. In 2011, the FAA recommended double duty at night because of incidents of controllers sleeping on duty. In 2014, China Eastern Airlines Flight MU2528 was forced to turn around during its approach to Wuhan because the controller was asleep on duty.

In 2016, due to fatigue, the tower controller of Shanghai Hongqiao Airport gave conflicting control instructions, which led to the aircraft taking off and crossing the runway using the runway at the same time, resulting in an A-class runway invasion incident. Fatigue seriously affects the safety of the civil aviation industry. Increasingly, more researchers are committed to solving the problem of fatigue, currently from both subjective and objective perspectives.

The subjective aspect is the use of fatigue scales; the objective aspect includes the detection of physical and psychological parameters and their use, of which the most suitable parameter for controller fatigue detection is the detection method based on eye condition.

In 2019, Jin et al. [1] proposed using the support vector machine model to fuse multiple physiological parameters and eye movement indicators to construct a controller fatigue detection model. The accuracy of identifying the normal group and the sleep-deprived group was 94.2%. Zhao et al. [2] proposed an EM-convolution neural network to detect the

state of the eyes and mouth from ROI images. The algorithm performance was better than that of algorithms based on VGG16, InceptionV3, AlexNet and others, with an accuracy and sensitivity of 93.623% and 93.643%, respectively.

Feng et al. [3] proposed adding a central loss function to softmax loss to optimize the problem of large intraclass spacing in deep convolutional networks and improve the accuracy of facial fatigue state recognition. Zheng [4] proposed a method that combines the MTCNN algorithm with an improved discriminative scale space tracking algorithm for face detection and key point positioning, and it uses the MobileNet V2 algorithm to determine the state of the eyes and mouth.

For comprehensively judging whether a driver is fatigued through fatigue indicators such as PERCLOS value, blink frequency, closed eye time, and yawn frequency. Mahmoud et al. [5] used the YOLO algorithm to count the number of people in a specific area and built a face detection system based on deep learning. The YOLO algorithm directly implements an end-to-end training process and has a higher advantage in detection rate, and the recall and accuracy rates also showed great improvement.

Xiao et al. [6] proposed a method to detect the fatigue state of a driver by using the spatiotemporal characteristics of the driver's eyes. In the end, the built model was used to detect the driving state, and an accuracy of 96.12% was achieved. Hu et al. [7] optimized the single shot multibox detector method to improve the robustness under light changes and similar background interference. However, in dealing with the task of data scarcity, the current network often produced data dependence problems. In order to solve the problem of fewer datasets, a transfer learning strategy was proposed.

The concept of deep transfer learning was also proposed, and network-based transfer learning can be widely used in different fields. Xie et al. [8] used the network model trained on ImageNet to fine-tune and migrate on the DeepFashion dataset. Transfer learning effectively improves the classification accuracy and timeliness of the model. In the field of medical imaging, where data are scarce, transfer learning is an effective method. Atabansi et al. [9] used the high-resolution image features of the large dataset to train a relatively small dataset model, which enhanced the generalization ability and verified the transfer learning strategy, and a higher accuracy rate was obtained.

Khan et al. [10] used the public PCG dataset to pre-train a simple and lightweight CNN model for the detection of cardiovascular diseases and obtained a high detection accuracy rate. At present, the methods for eye feature extraction and state determination are mainly machine-learning and deep-learning methods.

Traditional machine learning methods all use shallow structures. These structures contain, at most, one or two layers of nonlinear transformations, including logistic regression, random forest, SVM, maximum entropy, Gaussian mixture models etc. The shallow structure can solve simple problems, or the effect is obvious for more restricted ideal problems; however, due to the manual extraction of feature information, selecting effective texture features often requires a lot of time and rich experience.

With the rapid development of deep learning, a large number of models based on supervised training [11] and unsupervised training [12] have been proposed, such as Deep Convolutional Neural Networks (DCNN) [12], Deep Neural Networks (DNN) [13,14], Deep Belief Networks (DBN) [15], Long Short-Term Memory (LSTM) [16], non-linear unit activation function (Rectified linear unit, ReLU) [14], and the Dropout strategy [14], among others.

In the field of image detection and recognition, compared with traditional methods, deep learning methods have the advantage of omitting the steps for artificial feature extraction. At present, DNN and DCNN models are widely used in this field.

The following points summarize the main contributions of this work:

(1) A single network model needs to be optimized for detection accuracy. Therefore, combining the DNN model extraction vector features and DCNN model extraction texture features, a deep-fusion neural network (DFNN) model was built, which can extract image features more accurately.

(2) In order to solve the problem of insufficient controller fatigue data and the data dependence of the deep learning network model, the transfer learning strategy is used to pre-train the DNN and DCNN network, and the trained parameters are transferred to the DFNN model. The DFNN model has higher accuracy and reliability in detecting small-sized images of eyes compared with the trained VGG [17], ResNet [18] and Inception [18,19] models.

(3) Aiming at the special low-light working environment of the controller, the controller needs to constantly scan the radar screen, issue control instructions and deploy flight conflicts. Combined with the real-time requirements for the fatigue detection task of the controller, an eye selection mechanism (ES) is proposed, which can select a single eye for fatigue detection to increase the detection rate.

In this paper, by building an ES-DFNN controller fatigue detection model based on transfer learning, the memory of the model is reduced, and the detection accuracy and real-time performance are further improved. The structure of this paper is as follows: Section 2 outlines the fatigue detection process and the key technologies of fatigue detection. Section 3 focuses on the eye fatigue state detection model. The dataset and experimental results are described in detail in Section 4. Finally, the main research results are analyzed and summarized in Section 5.

2. Preliminary Background

The fatigue testing process is shown in Figure 1.

First, the video image is used to detect the face of the controller through MTCNN and, at the same time, the coordinates of the left and right eyes are obtained. Secondly, the left-eye or right-eye image to be detected is obtained through the eye selection mechanism. Thirdly, DCNN and DNN models are pre-trained by transfer learning on the FER2013 [20] and LFW [21] datasets, respectively. The two trained models are fused to build a DFNN model. Fourthly, the eye state dataset is used to fine-tune the DFNN model. Finally, determination of whether the controller is fatigued occurs through PERCLOS.

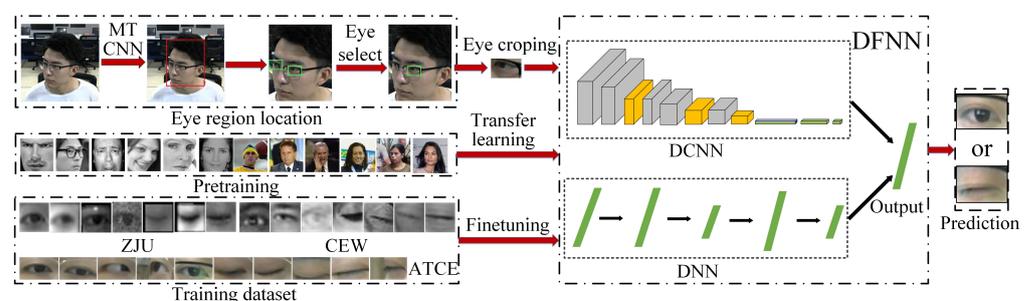


Figure 1. Fatigue detection flow chart.

2.1. Face Detection and Feature Point Positioning

Face detection and feature point positioning are the key parts of fatigue recognition. In the actual complex control environment, because the approach and area controllers need to pay attention to the aircraft dynamics on the radar screen in real time, the light is dimmed to ensure that the controller can see the radar screen in the control room clearly. At present, the traditional face detection method based on Adaboost classifier [22] is susceptible to interference from a complex background and dim lighting conditions, resulting in unstable detection results, and it is easy to falsely detect similar face areas as human faces; thus, the false detection rate is high.

The method based on template matching cannot be adaptively changed due to the size and shape of the template, and it is easily affected by changes in the controller's posture and the occlusion of objects in practical applications. Thus, the requirements for face detection and face key point positioning can no longer be met. MTCNN can combine

face detection and face key point positioning at the same time, and the positioned face key points can be used to realize face correction [23].

The MTCNN algorithm consists of three stages, as shown in Figure 2.

The first stage is the P-Net convolutional neural network, where the candidate windows and boundary regression vector are obtained. The candidate forms are calibrated according to the bounding box, and the nonmaximum value suppression algorithm is used to remove overlapping windows.

The second stage is the R-Net convolutional neural network, which trains the pictures containing candidate forms determined by P-Net in the R-Net network and uses the fully connected neural network for classification. Bounding box vectors are used to fine-tune candidate windows and nonmaximum suppression algorithms to remove overlapping windows.

The third stage is the O-Net convolutional neural network, whose network and function are similar to R-Net, and, while removing the overlapping candidate windows, the positions of five key points of the face are calibrated.

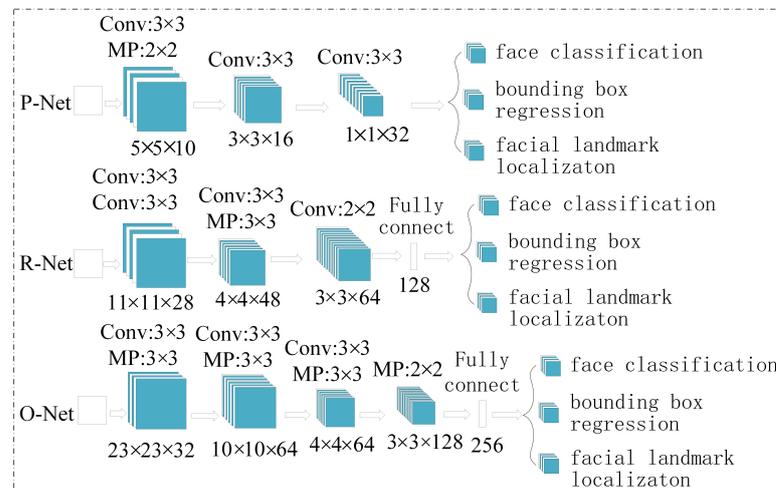


Figure 2. MTCNN network structure diagram.

Face detection and key point positioning are shown in Formula (1).

$$(face, L - eye, R - eye) = MTCNN(image) \quad (1)$$

Among them, *face* is the coordinates of the bounding box of the detected face; *L - eye* and *R - eye* represent the point coordinates of the left eye and right eye respectively; *image* is the video image to be detected.

2.2. Transfer Learning

Transfer learning defines the concepts of domain and task [24]. The domain $D = \{\chi, P(X)\}$ includes two parts: the feature space χ and the edge probability distribution $P(X)$ ($X = \{x_1, x_2, \dots, x_n\} \in \chi$); the task $T = \{y, f(x)\}$ includes two parts: the label space y and the target prediction function $f(x)$. The source domain is defined as D_s , the source task is T_s , the target domain is D_t , and the target domain task is T_t . Transfer learning is to transfer the relevant information based on D_s and T_s to T_t based on D_t in the case of $D_s \neq D_t$ or $T_s \neq T_t$, aiming to extract and transfer the potentially transferable knowledge in D_s and T_s to improve the efficiency of the prediction function. The schematic diagram of transfer learning is shown in Figure 3.

At present, there are two problems in constructing a controller fatigue detection model with high accuracy and reliability. On the one hand, there are less data on eye fatigue of controllers, and data collection is more complex, expensive, and affects normal control tasks. It is difficult to construct a large-scale, high-quality labeled controller fatigue dataset;

on the other hand, the existing deep learning methods are severely data dependent, and large-scale data are needed to understand the potential information under the data.

The feature extraction layer in the deep network model can extract the advanced characteristics of the training data, and the decision-making layer can identify the information needed to help make the final decision.

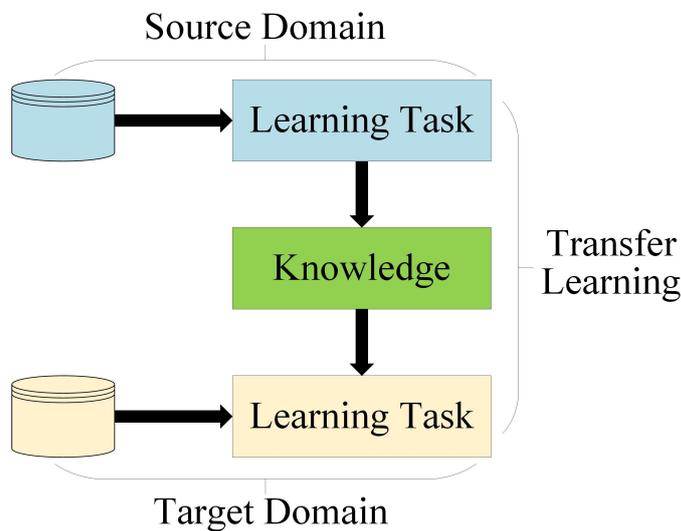


Figure 3. Schematic diagram of transfer learning.

Transfer learning allows flexibility with regard to the two basic assumptions in traditional classification tasks: (1) the training samples and the new test samples meet the condition of independent and identical distribution; (2) there must be large-scale and high-quality training samples [25]. The theory of transfer learning provides a method to solve this problem.

First, this paper pre-trains the DNN and DCNN network models by using the FER2013 and LFW datasets that are related to the target domain data or pixels similar to each other to obtain the initial parameters of the deep model. Second, the pre-trained DNN and DCNN model parameters are transferred to the fused DFNN model, and the feature extraction layer of the DFNN model is frozen, and part of the fully connected layer and output layer are opened. Finally, the DFNN model is fine-tuned using the controller's eye image to obtain an eye state classification network model.

2.3. Eye Selection Mechanism

In the actual control environment, the controller needs to scan the radar screen back and forth uninterruptedly. Therefore, the head posture of the controller is diversified. When one eye is blocked due to head deflection, it is difficult to correctly detect the state of both eyes at the same time. When this happens, one eye can remain undetected, which can greatly interfere with the detection result. When the head is greatly tilted or deflected, the left and right eye areas are selected to detect the unobstructed eyes. When the left and right eyes are not covered, monocular with high confidence is also detected.

The eye selection mechanism is shown in Figure 4, where f_w and f_h represent the width and height, respectively, of the face regression box detected by MTCNN, and d represents the vertical distance from the midpoint of the abscissa of the left and right eyes to the right boundary of the face regression box. The formula is as follows:

$$E = \begin{cases} E_L & \text{if } d < f_w/2, \\ E_R & \text{if } d \geq f_w/2. \end{cases} \quad (2)$$

In the Formula (2), when d is less than $f_w/2$, the left eye is selected as the eye to be tested; otherwise, the right eye is tested.

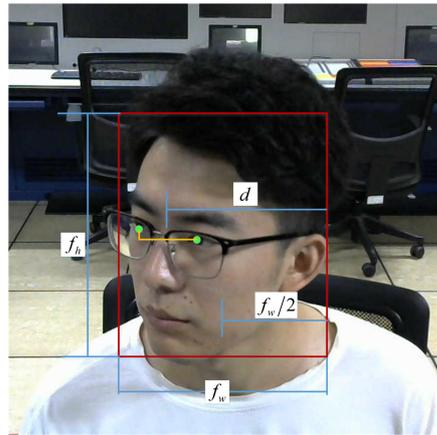


Figure 4. Schematic diagram of the eye selection mechanism.

3. Methodology

At present, the algorithms for recognizing the open and closed state of eyes are divided into two types: manual feature extraction and automatic feature extraction. Among them, manual feature extraction mainly includes the template matching detection method, texture feature detection method, and shape feature detection method [26,27]. These methods rely on the extraction of texture features, and the selection of texture features requires a great deal of experimentation and sufficient experience.

The automatic extraction of features is used in deep learning methods, such as deep neural networks [28,29], deep convolutional neural networks [30] and recurrent neural networks [31], omitting manual extraction of features and automatically extracting advanced features of the dataset. Accuracy and reliability are also better than in manual feature extraction methods.

In deep learning methods, DNN is mainly used for natural language processing and visual target detection and recognition, such as speech recognition [32], wind speed prediction [33] and image classification. However, as the depth of the network increases, the number of parameters exponentially increases. When processing target detection and segmentation tasks, the gradient becomes increasingly sparse and converges to a local minimum.

The deeper the network, the higher the calculation performance requirements. DCNN is mainly used in speech recognition, document analysis [34], language detection, image recognition [35] and other fields, through convolution operations and pooling the dimensionality reduction and fully connected layer process images, which can effectively extract features. A single network model is easily affected by gradient dissipation and local optimization, resulting in poor accuracy and reliability.

The DFNN model can meet real-time requirements with its shallow depth and small memory. The DCNN model used for fusion mainly extracts pictures the texture feature, and the DNN model extracts vector features by converting the picture into a one-dimensional vector. The fused DFNN model can extract eye features more finely, which can meet the accuracy requirements. The advantages and disadvantages of the existing methods are shown in Table 1.

3.1. DCNN Model

A deep convolutional neural network is a network model composed of several layers of “neurons” [12]. Each neuron in the current layer applies a linear filter to the output of the previous layer of neurons and superimposes a bias on the output of the filter. A nonlinear activation function is applied to the result, which allows us to obtain a feature map.

Table 1. The advantage and disadvantage of methods.

Method	Advantage	Disadvantage
Template matching detection	The method is simple.	The method requires a large number of different human eye templates for matching, which requires a large amount of calculations, has poor real-time performance and is susceptible to facial expressions.
Texture feature detection	The method includes statistical calculations in a region with multiple pixels, often with rotation invariance, and it has strong resistance to noise.	The method is seriously affected by resolution and may be affected by illumination and reflection, and the texture reflected from the 2-D image is not necessarily the real texture of the surface of the 3-D object.
Shape feature detection	The algorithm is simple to implement, does not require offline training, and has a fast calculation speed and high detection rate.	The method is not sensitive to face and expression changes at multiple angles, and it is easy to misjudge nonface skin color areas (hands, neck, etc.) and skin-like areas in the background.
DNN	The method has a simple network structure.	The method is prone to sparse gradients and requires high computational performance.
DCNN	The method has higher detection accuracy.	The method is not effective in discriminating samples with extreme head posture and is susceptible to background interference.
DFNN	This method has a faster detection rate, high detection accuracy and good robustness.	This method will produce false detections for extreme head posture samples.

(1) The convolutional layer is the core of the entire neural network, which uses two methods of “local perception” and “weight sharing” to perform dimensionality reduction and feature extraction. Compared with the neural network with different filters applied to all neurons, the number of parameters for the convolution shared filter structure is drastically reduced, reducing its ability to overfit. The formula is as follows:

$$Z^{l+1}(i, j) = [Z^l \otimes W^{l+1}](i, j) + b \quad (i, j) \in \{0, 1, \dots, L_{l+1}\} \quad (3)$$

$$L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1 \quad (4)$$

In Formula (3), Z^l and Z^{l+1} are the input and output of the $l + 1$ layer, $Z^{l+1}(i, j)$ is the pixel of the $l + 1$ layer feature map, W is the convolution kernel, and b is the bias term. In Formula (4), s_0 , p and f are the convolution step size, the number of filling layers and the size of the convolution kernel, respectively. L is the number of network layers, and the convolution step size refers to the step size of the convolution kernel at each time.

(2) The pooling layer is also called the downsampling layer, which performs feature selection and filtering on the feature map. The pooling layer uses max-pooling with a size of 2×2 .

(3) The fully connected layer performs a nonlinear combination of the features extracted by the convolutional layer and the pooling layer to achieve classification.

$$A^l = f(W^T A^{l-1} + b) \quad (5)$$

In Formula (5), A^{l-1} and A^l are the input and output of the l layer, f is the activation function, and W and b are the weight and bias, respectively.

The DCNN model consists of six convolutional layers, three pooling layers and one fully connected layer, as shown in Figure 5. The size of the convolution kernel of the first convolutional layer is $32 \times 3 \times 3$, and the size of the convolution kernel of the other convolutional layers is $128 \times 3 \times 3$. In all convolutional layers, the boundary mode of the convolution operation is the same, that is, the dimensions of the input and output feature maps in the convolution operation are the same. The pooling layer uses the max-pooling

strategy to reduce the dimensionality of the feature map, and the dimensionality reduction ratio of all pooling layers is 2×2 .

In order to prevent the model from overfitting due to the small dataset, set BatchNormalization after the convolutional layer, add Dropout regularization after the pooling layer, and set the Dropout regularization parameter to 0.25. The number of units in the fully connected layer is 512. Finally, a softmax classifier is added to the top layer as the output of the model. The activation functions of all layers in the model are ReLU functions.

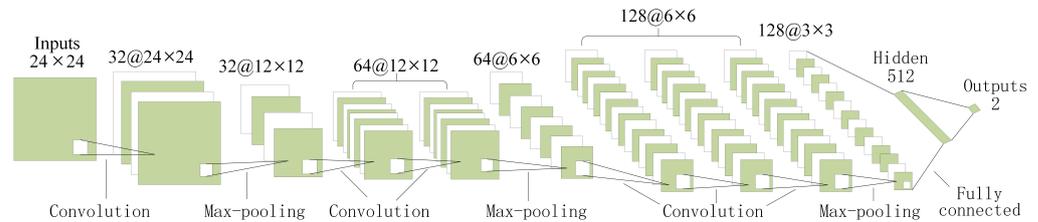


Figure 5. DCNN structure diagram.

The DCNN model mainly performs convolution calculation, pooling dimensionality reduction and fully connected flattening into a one-dimensional vector on the controller's eye image obtained by the eye screening mechanism. The texture feature extraction is performed on the image of the controller's eyes to determine the state of the controller's eyes as open or closed.

3.2. DNN Model

The full name of DNN is deep neural network [28]. Its model structure is shown in Figure 6. It consists of one input layer, three hidden layers and one output layer. The number of input layer units is $24 \times 24 = 576$; the numbers of neurons in the hidden layer are 256, 512 and 256; the output layer is a softmax classifier, and the number is 2. First, the DNN model preprocesses the eye image and converts the extracted eye image size into pixels. Second, it converts the two-dimensional image into a one-dimensional vector by fully connecting the input image of the controller's eye.

The input vector is normalized, and the vector features of the eye image are extracted through the hidden layer through the weight parameter and the nonlinear unit activation function. Finally, the softmax judges the state of the eyes as open or closed. All activation functions in the model are ReLU functions, and the Dropout value of each layer is set to 0.5.

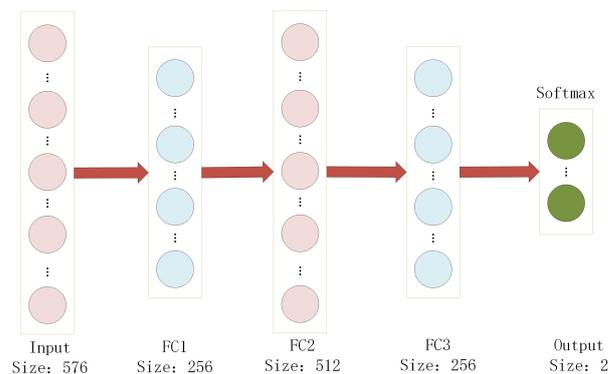


Figure 6. DNN structure diagram.

3.3. DFNN Model

The built DNN model and DCNN model are fused in the proposed DFNN model. The DCNN model mainly extracts the texture features of the eye image through convolution operation, pooling dimensionality reduction and the fully connected layer, while DNN mainly extracts the eye through the fully connected layer and the vector features

of the picture. The DFNN model can more finely extract the features useful for eye state classification. The DFNN structure diagram is shown in Figure 7.

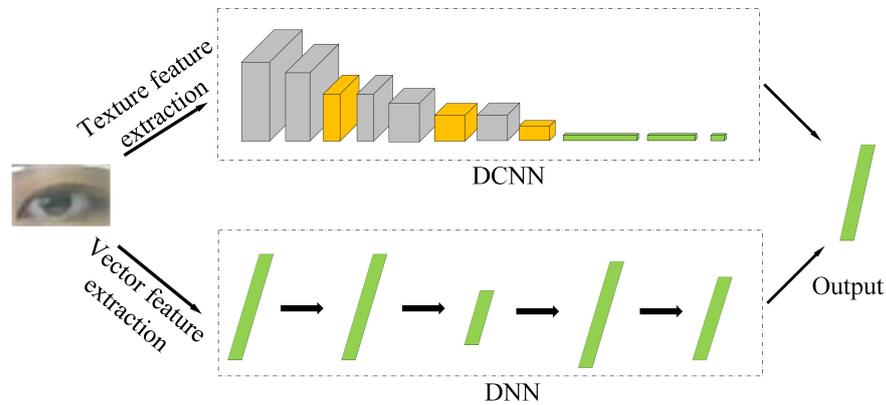


Figure 7. DFNN structure diagram.

First, the eye image is input to the DCNN model, and the eye image is converted into a one-dimensional vector and input to the DNN model. Then the result weighted average method is used to fuse the output results of the fully connected layers of the two models, where the weight of the DCNN model is 0.6 and the weight of the DNN model is 0.4, the fusion flow chart is shown in Figure 8. Finally, the softmax classifier is used to classify the fused features.

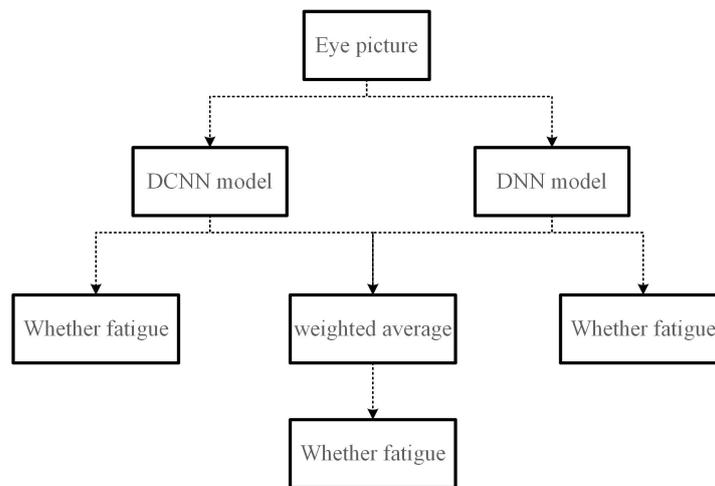


Figure 8. Fusion flow chart.

3.4. Control Fatigue Judgment Index

When the controller has scanned the radar screen for a long time, adjusting the flight interval and issuing control instructions, fatigue characteristics will begin to appear, such as slow blinking, long-term continuous closed eyes etc. Therefore, the controller’s fatigue level can be judged by obtaining the controller’s eye status information. *PERCLOS* represents the ratio of the number of closed eye frames to the total number of frames in that period of time [36],

$$PERCLOS = \frac{m}{M} \times 100\% \tag{6}$$

In Formula (6), *m* represents the number of closed-eye frames, and *M* represents the total number of eye-detected frames during this period. When *PERCLOS* is greater than the threshold, the controller is determined to be in a fatigue state. In the specific test, there are three measurement methods: EM, P70 and P80, as shown in Table 2.

Table 2. PERCLOS judgment standard.

Judgement Standard	Percentage of Average Eyes Closed	Threshold
EM	More than 50%	$PERCLOS > 0.5$
P70	More than 70%	$PERCLOS > 0.7$
P80	More than 80%	$PERCLOS > 0.8$

4. Experimental

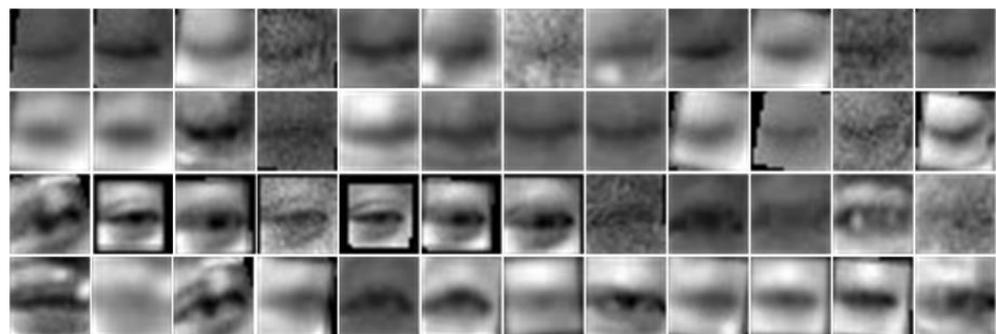
4.1. Experimental Environment

The verification experiment was conducted on a Windows operating system, equipped with an Intel Xeon Silver 4110 CPU and two NVIDIA GTX1080Ti 11 G independent graphics display cards. The storage hardware specifications were 128 GB 2666 MHz ECC memory, 480 G SSB and a 4 TB SATA hard disk. Keras and Tensorflow were used to build the neural network model.

4.2. Experimental Datasets

Considering the real-world scenario of the controller's work, it may be affected by individual differences and various environmental changes, including lighting, masking, and blurring. To study the performance, accuracy and loss rate of the DFNN model under the above conditions, ZJU, CEW and ATCE datasets were collected, where 70% of the datasets were selected as the training dataset, and 30% of the datasets were used as the test dataset.

(1) The ZJU dataset [37] is an open source dataset published by Zhejiang University. In the 20-person flashing video database, there are a total of 80 video clips, and each person has four clips: (a) frontal viewing fragments without glasses, (b) viewing fragments wearing thin-rim glasses, (c) frontal viewing fragments wearing black-rimmed glasses and (d) upwards viewing fragments without glasses. Images are manually selected during each blinking process, including open, half-open, closed and half-closed eye images. In addition, images of the left and right eyes are collected separately. These images may be blurred, low resolution or obscured by glasses. Some samples of this dataset are shown in Figure 9. The first two lines are closed-eye images, and the last two lines are open-eye images.

**Figure 9.** The ZJU dataset.

(2) The CEW dataset [38] was released by Nanjing University of Aeronautics and Astronautics, including 2423 images, of which 1192 closed-eye images were collected from the internet, and 1231 open-eye images were from the Labeled Faces in the Wild database. The eye images in this dataset are shown in Figure 10.



Figure 10. The CEW dataset.

(3) ATCE dataset. First, the real-time facial images of controllers in the Civil Aviation Flight University of China were collected when they were carrying out radar simulator control tasks. Then, the collected facial images are recognized and extracted by the MTCNN model to obtain the ATCE dataset. The dataset has a total of 4326 images, of which 2516 images are of open eyes and 1810 of closed eyes, and some of the images are shown in Figure 11.



Figure 11. The ATCE dataset.

4.3. Experimental Analysis

The eye state recognition model in this paper is experimentally analyzed on three different datasets of ZJU, CEW and ATCE. First, the accuracy, loss rate, F1 score and area under the receiver operating characteristic curve (AUC) values are compared for the VGG16, InceptionV3, ResNet50 and DFNN network models on the three datasets mentioned above. Second, on the ZJU and CEW datasets, the recognition accuracy and AUC value of this method are compared with those of the methods proposed by other researchers.

4.3.1. Test Results of Different Networks on the ZJU Dataset

Comparing the VGG16 model, InceptionV3 model, ResNet50 model and the DFNN model presented in this paper on the ImageNet competition classification task, the comparison results of accuracy and loss rate are shown in Figure 12, the recall rate, recognition accuracy, F1 score, loss rate, AUC, model size, running time and training time are shown in Table 3.

In Figure 12 left, the DFNN model training dataset and test dataset have the highest accuracy, the training dataset accuracy rate is 96.97%, and the test dataset accuracy rate is 96.30%. ResNet50 has the lowest accuracy rate, 89.58% for the training dataset and 84.79% for the test dataset. The accuracy rate of the training dataset and test dataset of the VGG16 model is 92.36%. The accuracy rate of the training dataset of the InceptionV3 model is 93.45%, and the accuracy rate of the test dataset is 92.79%.

The recognition accuracy of the DFNN model is 4.61% higher than that of the VGG16 model, 4.18% higher than the InceptionV3 model, and 7.39% higher than the ResNet50 model. In Figure 12 right, the loss rate of the training dataset of the DFNN model is 8%,

and the loss rate of the test dataset is 9%. The effect of the ResNet50 model is the worst, the loss rate of the training dataset is 26.78%, the loss rate of the test dataset is 34.70%, the loss rate of the training dataset and the test dataset of the VGG16 model is 18%, and the loss rate of the InceptionV3 model is 18%.

The loss rate of the training dataset is 17.19%, and the loss rate of the test dataset is 15.72%. The loss rate of the DFNN model is 8.97% lower than that of the VGG16 model, 8.16% lower than that of InceptionV3, and 17.75% lower than that of ResNet50. From the above experiments, it can be seen that the accuracy rate of the 30th generation of the DFNN model on the training set and the test set is stable at approximately 96%, and it starts to converge in the 20th generation, and the loss rate approaches 9%. The DFNN model is superior to the other three models in the task of eye small-size image classification.

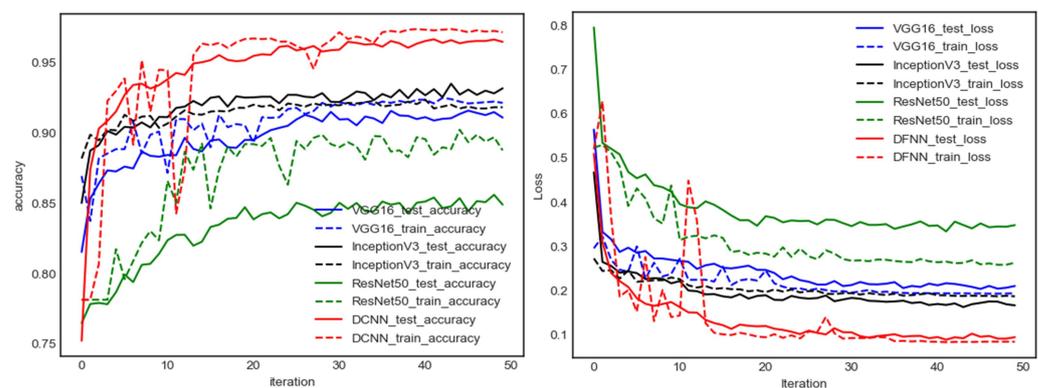


Figure 12. Comparison results of DFNN and the other three models on the ZJU dataset.

F1 score is the harmonic average of recall and precision. In Table 3, the F1 score of the DFNN model is 96.97%, while the F1 score of the ResNet50 and InceptionV3 models is about 92%. The DFNN model is better than the other three models. The DFNN model has a model size of 53 MB. The runtime is 326.96 s. The training time was 57 ms/step. Regarding all three aspects, the DFNN model is superior to the three network models. It can better meet the needs of control tasks and meet the requirements of safety, accuracy and real-time operations.

Table 3. VGG16, ResNet50, InceptionV3, and DFNN evaluation indicators on the ZJU dataset.

Index	VGG16	ResNet50	InceptionV3	DFNN
Recall(%)	92.56	89.58	92.79	96.97
Precision(%)	92.40	89.22	92.64	96.96
F1 score(%)	92.36	89.23	92.66	96.97
AUC(%)	96.92	93.73	97.18	99.03
Network size	344 MB	1.11 GB	372 MB	53 MB
Run time	691.47 s	954.17 s	935.33 s	326.96 s
Training time	131 ms/step	154 ms/step	163 ms/step	57 ms/step

4.3.2. Test Results of Different Networks on the CEW Dataset

The comparison of the accuracy and loss rate curves of DFNN and the other three models on the CEW dataset for eye image training and testing is shown in Figure 13. It can be seen from the figure that the DFNN model begins to converge in about 10 generations. The accuracy rate of the model training set and test set is close to 97%, while the loss rate of the model training and testing is around 6%. The VGG16 model and the InceptionV3 model converge earlier than the DFNN model. However, the recognition accuracy of the DFNN model is about 3% higher than the two types. The ResNet50 model lags behind DFNN in terms of the convergence speed, model accuracy and loss rate.

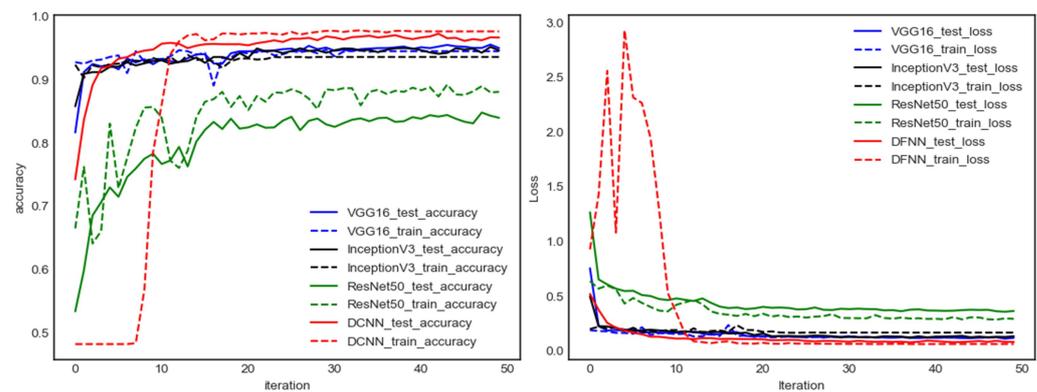


Figure 13. Comparison results of DFNN and the other three models on the CEW dataset.

In Table 4, the F1 score of the DFNN model is 97.36%, the F1 score of the VGG16 model is 95.38%, and the F1 score of the ResNet50 model is 89.09%. Among the four models, the F1 score of DFNN model is about 2% to 7% higher than those of the other three models. The DFNN model has a model size of 53 MB, a running time of 182.69 s and a training time of 65 ms/step and is superior to the other three network models in these three aspects. On the CEW dataset, the DFNN model has a model size of 53 MB, a running time of 182.69 s and a training time of 65ms/step, which is still better than the other three network models.

Table 4. VGG16, ResNet50, InceptionV3, and DFNN evaluation indicators on the CEW dataset.

Index	VGG16	ResNet50	InceptionV3	DFNN
Recall(%)	95.38	89.11	93.98	97.36
Precision(%)	95.40	89.21	94.07	97.37
F1 score(%)	95.38	89.09	93.97	97.36
AUC(%)	99.20	95.58	98.55	99.71
Network size	344 MB	1.11 GB	372 MB	53 MB
Run time	375.75 s	510.07 s	946.91 s	182.69 s
Training time	127 ms/step	162 ms/step	158 ms/step	65 ms/step

4.3.3. Test Results of Different Networks on the ATCE Dataset

Figure 14 shows the comparison of the accuracy and loss rate curves of DFNN and the other three models on the ATCE dataset for eye image training and testing. It can be seen from the figure that, in the task of distinguishing eye states, the DFNN model starts to converge after the number of iterations reaches 30. The accuracy of training and testing reaches 98.4%, and the loss rate is 4.57%. In Figure 14 left, the accuracy rate of the training dataset and test dataset of the VGG16 model is about 97%, the accuracy rate of the training dataset and test dataset of the InceptionV3 model is about 97%, and the accuracy rate of the training dataset of the ResNet50 model is about 91.40%.

The accuracy of the test dataset is about 87.21%. In Figure 14 right, the effect of the ResNet50 model is the worst. The loss rate of the training dataset is about 22.55%, the loss rate of the test dataset is about 28.71%, and the loss rate of the training dataset and test dataset of the VGG16 model is near 7%. The loss rate of the training dataset and test dataset of the InceptionV3 model is about 6%. The loss rate of the DFNN model is 2.43% lower than that of the VGG16 model, 1.43% lower than InceptionV3 and 17.98% lower than ResNet50.

In Table 5, the F1 score of the DFNN model is 98.43%, the F1 score of the VGG16 model is 97.51%, the F1 score of the ResNet50 model is 91.45%, and the F1 score of the InceptionV3 model is 97.69%. The F1 score of the DFNN model is 0.92% higher than that of the VGG16 model, 6.98% higher than the ResNet50 model and 0.74% higher than the InceptionV3 model. On the CEW dataset, the DFNN model has a model size of 53 MB, a running time of 188.62 s, and a training time of 59 ms/step, which is better than the other three network models.

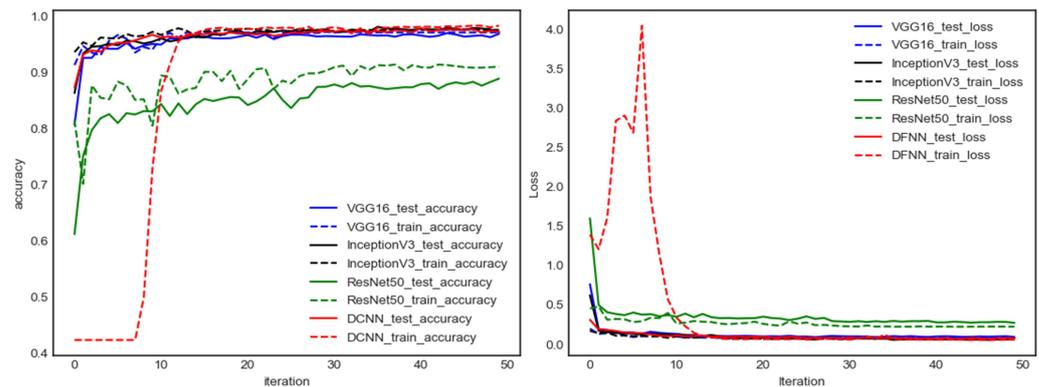


Figure 14. Comparison results of DFNN and the other three models on the ATCE dataset.

Table 5. VGG16, ResNet50, InceptionV3, and DFNN evaluation indicators on the ATCE dataset.

Index	VGG16	ResNet50	InceptionV3	DFNN
Recall(%)	97.50	91.40	97.69	98.43
Precision(%)	97.54	91.87	97.70	98.44
F1 score(%)	97.51	91.45	97.69	98.43
AUC(%)	99.65	96.73	99.71	99.85
Network size	344 MB	1.11 GB	372 MB	53 MB
Run time	323.32 s	510.26 s	446.05 s	188.62 s
Training time	116 ms/step	154 ms/step	159 ms/step	59 ms/step

According to the comparative experimental results of the DFNN model and the other three models, it can be seen that the recognition accuracy of the DFNN model is better than that of the other three large-scale network models. Since the input of the DFNN network model is 24×24 , the number of convolutional layers and model parameters are less than the other three models. In terms of training performance, the DFNN model is more suitable for the classification task of the controller's eye image, which has smaller pixels and fewer features.

By longitudinally comparing the recognition accuracy and recall of the DFNN model on the three datasets, the DFNN model has a higher accuracy rate on the ATCE dataset and can detect the fatigue state of the controller more accurately and quickly.

4.3.4. Comparison of the Results of Different Methods on the ZJU Dataset

The DNN, DCNN and DFNN models are compared with the eye state recognition models proposed by Wu, Dong, Eddine, Liu and Song on the ZJU dataset. The comparison results are shown in Table 6. According to the experimental results, it can be seen that the average precision and AUC values of the multi-feature fusion recognition method based on MultiHPOG, LTP and Gabor are higher than other geometric feature methods. The precision and AUC values of the DNN and DCNN models are lower than the method proposed by Song. The precision and AUC values of the DFNN model based on the fusion of DNN and DCNN are higher than other methods.

4.3.5. Comparison of the Results of Different Methods on the CEW Dataset

The DNN, DCNN and DFNN models are compared with the eye state recognition methods proposed by Song and Dong on the CEW dataset. The comparison results are shown in Table 7. According to the experimental results, it can be seen that the precision of the projection-based recognition method is clearly poor. The average precision and AUC of the recognition method based on MultiHPOG, LTP and Gabor multi-feature fusion are significantly improved, while the precision and AUC of the DFNN based on the fusion of DNN and DCNN models are better than other methods.

Table 6. Comparison results with other methods on the ZJU dataset.

Research	Method	Precision(%)	AUC(%)
Wu [39]	LBP + SVM	90.37	-
Dong [27]	HOG + Random Forrest	94.70	98.37
Eddine [40]	Multi-TPLBP + MLP	95.18	97.83
Liu [41]	Gabor + LBP + HOG + SVM	95.42	98.02
Song [42]	MultiHPOG + LTP + Gabor + SVM	96.83	99.27
Ours	DNN	94.12	96.75
	DCNN	95.38	98.46
	DFNN	96.96	99.03

Table 7. Comparison results with the other methods on the CEW dataset.

Research	Method	Precision(%)	AUC(%)
Dong [27]	Projection	70.10	-
	HOG + Random Forrest	94.57	98.17
Song [42]	MultiHPOG-LTP + Gabor + SVM	94.72	95.19
	DNN	95.21	97.93
Ours	DCNN	96.13	98.76
	DFNN	97.37	99.71

4.3.6. Comparison of Real-time Fatigue Test Results

The method in this paper is compared with the methods proposed by others. The experimental results are shown in Table 8. Among them, the method proposed by Liu uses an ASL eye tracker to extract eye feature parameters, and the method of an SVM classifier to determine fatigue with poor recognition accuracy. This paper proposes MTCNN to achieve eye localization, ES-DFNN to extract eye features and, finally, the *PERCLOS80* index to detect fatigue. The recognition accuracy and speed are superior to the other two methods and can meet the real-time requirements.

Table 8. Fatigue testing performance comparison with the other methods.

Research	Method	Precision(%)	Rate/ms·Frame ⁻¹
Liu [43]	ASL + SVM	83.92	-
Xiao [6]	CNNs + LSTM	96.12	132.34
	MTCNN + ES-DFNN + EM	91.02	91.31
Ours	MTCNN + ES-DFNN + P70	92.41	91.42
	MTCNN + ES-DFNN + P80	97.30	91.24

5. Conclusions

Eye condition detection is the primary method for fatigue detection in air traffic controllers. In order to improve the accuracy and detection rate of fatigue detection, a ES-DFNN model based on the classification task of small pixel images of the eyes was proposed to realize the method for fatigue detection in a controller. The following conclusions are drawn:

- (1) In order to improve the robustness of the fatigue detection model, the MTCNN detection algorithm can be used to detect nonfrontal face images in real time.
- (2) An eye-screening mechanism was proposed. By detecting the deflection or tilt angle of the head and comparing the left and right eye detection confidence, the eye pictures to be tested were selected to replace traditional binocular detection. The detection rate was improved and meets the requirements for the real-time detection of fatigue status.
- (3) In order to improve the detection efficiency and accuracy, the DFNN model fused with DCNN and DNN was used to learn and extract eye fatigue features. Applying the DFNN model on the ZJU dataset resulted in the accuracy being increased by 7%. The increase for the CEW dataset ranged from 3% to 7%. On the ATCE dataset, the test accuracy of the DFNN model was improved by 2% compared with the ZJU dataset and the CEW dataset.

When this model recognizes extreme head postures, nondetection may occur. In future work, we will enrich the eye dataset under extreme head postures, optimize face detection methods and increase the diversity of detection to make it more consistent with the actual control situation.

Author Contributions: Conceptualization, H.L. and C.L.; methodology, C.L.; software, C.L.; validation, K.C., H.L. and J.K.; formal analysis, H.L.; investigation, Q.H.; resources, J.K.; data curation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, H.L.; visualization, K.C.; supervision, T.Z.; project administration, H.L.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This study was co-supported by National Natural Science Fund (No.U1733203) and Provincial College Students Innovation and Entrepreneurship Training Program (No.S202010624094).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to restrictions, e.g., privacy or ethical.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

ZJU	Zhejiang University
CEW	Closed Eyes in the Wild
ATCE	Air Traffic Controller Dataset
VGG	Visual Geometry Group
ResNet	Residual Block
YOLO	You Only Look Once
AUC	Area Under Curve
MTCNN	Multi-Task Convolutional Neural Network
DFNN	Deep-Fusion Neural Network
EM	Eye and Mouth
ROI	Region Of Interest
CNN	Convolution Neural Network
SVM	Support Vector Machines
DCNN	Deep Convolutional Neural Network
DBN	Deep Belief Network
LSTM	Long Short-Term Memory
DNN	Deep Neural Network
ReLU	Nonlinear Unit Activation Function
ES	Eye Selection Mechanism
LFW	Labeled Faces in the Wild
PERCLOS	Percentage of Eyelid Closure Over the Pupil
LBP	Linear Back Projection
HOG	Histogram of Oriented Gradients
TPLBP	Three-Patch Local Binary Pattern
MLP	Multilayer Perceptron
LTP	Local Ternary Patterns
MultiHPOG	Multi-Scale Histograms of Principal Oriented Gradients
ASL	Applied Science Laboratories

References

1. Jin, H.; Zhu, G.; Lv, C. Research on pipe fatigue detection model based on support vector machine. *J. Saf. Environ.* **2019**, *19*, 99–105.
2. Zhao, Z.; Zhou, N.; Zhang, L.; Yan, H.; Xu, Y.; Zhang, Z. Driver Fatigue Detection Based on Convolutional Neural Networks Using Em-cnn. *Comput. Intell. Neurosci.* **2020**, *2020*, 7251280. [[CrossRef](#)] [[PubMed](#)]
3. Feng, W.; Cao, Y.; Li, X.; Hu, W. Face fatigue detection based on improved deep convolutional neural network. *Comput. Intell. Neurosci.* **2020**, *20*, 5680–5687.

4. Zheng, W. *Fatigue Driving Detection Based on Deep Learning*; Liaoning University of Science and Technology: Anshan, China, 2020.
5. Elsis, M.; Tran, M.Q.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M.F. Deep Learning-Based Industry 4.0 and Internet of Things towards Effective Energy Management for Smart Buildings. *Sensors* **2021**, *21*, 1038. [[CrossRef](#)] [[PubMed](#)]
6. Xiao, Z.; Hu, Z.; Geng, L.; Zhang, F.; Wu, J.; Li, Y. Fatigue Driving Recognition Network: Fatigue Driving Recognition Via Convolutional Neural Network and Long Short-term Memory Units. *IET Intell. Transp. Syst.* **2019**, *13*, 1410–1416. [[CrossRef](#)]
7. Hu, X.; Huang, B. Fatigue Driving Detection System Based on Face Feature Analysis. *Sci. Technol. Eng.* **2021**, *4*, 1629–1636.
8. Xie, X.; Lu, J.; Li, W.; Liu, C. Huang, H. Classification model of clothing image based on migration learning. *Comput. Appl. Softw.* **2020**, *37*, 88–93.
9. Atabansi, C.C.; Chen, T.; Cao, R.; Xu, X. Transfer Learning Technique with Vgg-16 for Near-infrared Facial Expression Recognition. *J. Phys. Conf. Ser.* **2021**, *1873*, 12033. [[CrossRef](#)]
10. Khan, K.N.; Khan, F.A.; Abid, A.; Olmez, T.; Dokur, Z.; Khandakar, A.; Chowdhury, M.E.H.; Khan, M.S. Deep Learning Based Classification of Unsegmented Phonocardiogram Spectrograms Leveraging Transfer Learning. *Physiol. Meas.* **2021**, *42*, 095003. [[CrossRef](#)]
11. Lalithadevi, B.; Dubey, K.; Trivedi, Y.; Gautam, A.S. Novel Technique for Price Prediction By Using Logistic, Linear and Decision Tree Algorithm on Deep Belief Network. *Int. J. Psychosoc. Rehabil.* **2020**, *24*, 1751–1761.
12. Florkowski, M. Classification of Partial Discharge Images Using Deep Convolutional Neural Networks. *Energies* **2020**, *13*, 5496. [[CrossRef](#)]
13. Sarker, I. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *Preprints* **2021**, *2*, 1–7. [[CrossRef](#)] [[PubMed](#)]
14. Khumprom, P.; Grewell, D.; Yodo, N. Deep Neural Network Feature Selection Approaches for Data-Driven Prognostic Model of Aircraft Engines. *Aerospace* **2020**, *7*, 132. [[CrossRef](#)]
15. Hu, G.; Li, H.; Luo, L.; Xia, Y. An improved dropout method and its application into DBN-based handwriting recognition. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 1222–1226.
16. Sherstinsky, A. Fundamentals of Recurrent Neural Network (rnn) and Long Short-term Memory (lstm) Network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
17. Jun, H.; Shuai, L.; Jinming, S.; Yue, L.; Jingwei, W.; Peng, J. Facial Expression Recognition Based on VGGNet Convolutional Neural Network. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 4146–4151.
18. Swinney, C.J.; Woods, J.C. Unmanned Aerial Vehicle Operating Mode Classification Using Deep Residual Learning Feature Extraction. *Aerospace* **2021**, *8*, 79. [[CrossRef](#)]
19. Lu, S.; Wang, B.; Wang, H.; Chen, L.; Linjian, M.; Zhang, X. A real-time object detection algorithm for video. *Comput. Electr. Eng.* **2019**, *77*, 398–408. [[CrossRef](#)]
20. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.-H.; et al. Challenges in Representation Learning: A Report on Three Machine Learning Contests. *Neural Netw.* **2015**, *64*, 59–63. [[CrossRef](#)] [[PubMed](#)]
21. Rim, D.; Hasan, K.; Puech, F.; Pal, C.J. Learning From Weakly Labeled Faces and Video in the Wild. *Pattern Recognit.* **2015**, *48*, 759–771. [[CrossRef](#)]
22. Xu, Y.; Qiu, T. Human Activity Recognition and Embedded Application Based on Convolutional Neural Network. *J. Artif. Intell. Technol.* **2020**, *1*, 51–60. [[CrossRef](#)]
23. Ghofrani, A.; Toroghi, R.M.; Ghanbari, S. Realtime Face-detection and Emotion Recognition Using Mtcnn and Minishufflenet V2. In Proceedings of the 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, 28 February–1 March 2019; pp. 817–821.
24. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN, Rhodes, Greece, 4–7 October 2018; pp. 270–279.
25. Qiang, Z. *Research on Transfer Learning Algorithm for Image Classification*; Beijing University of Posts and Telecommunications: Beijing, China, 2021.
26. Hu, R. Random Neural Networks for Dimensionality Reduction and Regularized Supervised Learning. Ph.D. Thesis, University of Iowa, Iowa City, IA, USA, 2019.
27. Dong, Y.; Zhang, Y.; Yue, J.; Hu, Z. Comparison of Random Forest, Random Ferns and Support Vector Machine for Eye State Classification. *J. Phys. Conf. Ser.* **2015**, *75*, 11763–11783. [[CrossRef](#)]
28. Moolayil, J. Deep Neural Networks for Supervised Learning: Classification. In *Learn Keras for Deep Neural Networks*; Apress: Berkeley, CA, USA, 2019; pp. 101–135.
29. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)]
30. Aggarwal, C.C. Weighted Voting of Multi-Stream Convolutional Neural Networks for Video-Based Action Recognition using Optical Flow Rhythms. *J. Vis. Commun. Image Represent.* **2021**, *77*, 103112.
31. Yalçın, O.G. Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks. *Nat. Commun.* **2020**, *11*, 1551.
32. Hu, G.; Szu-Han K.; Neal M. Natural Language Processing Based Method for Clustering and Analysis of Aviation Safety Narratives. *J. Artif. Intell. Technol.* **2021**, *1*, 138–143.

33. Chen, J.; Liang, Z. Research on Speech Enhancement Algorithm Based on EEMD Data Preprocessing and DNN. *J. Ordnance Equip. Eng.* **2019**, *40*, 96–103.
34. Rose, R.L.; Puranik, T.G.; Mavris, D.N. Deep Neural Network-based Speaker-Aware Information Logging for Augmentative and Alternative Communication. *Aerospace* **2020**, *7*, 143. [[CrossRef](#)]
35. Xie, X.; Xue, S. Application of Cifar10 Model for Improved in Armor Target Binary Classification. *J. Ordnance Equip. Eng.* **2019**, *40*, 141–144.
36. Ed-Doughmi, Y.; Idrissi, N. Driver Fatigue Detection using Recurrent Neural Networks. In Proceedings of the 2nd International Conference on Networking, Information Systems & Security, Rabat, Morocco, 27–29 March 2019; pp. 1–6.
37. Pan, G.; Sun, L.; Wu, Z.; Lao, S. Eyeblink-based Anti-spoofing in Face Recognition From a Generic Webcam. In Proceedings of the IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; Volume 11, pp. 14–20.
38. Learned-Miller, E.; Huang, G.B.; RoyChowdhury, A.; Li, H.; Hua, G. Labeled Faces in the Wild: A Survey. *Adv. Face Detect. Facial Image Anal.* **2016**, 189–248.
39. Wu, Y.S.; Lee, T.W.; Wu, Q.Z.; Liu, H.S. An Eye State Recognition Method for Drowsiness Detection. In Proceedings of the 2010 IEEE 71st Vehicular Technology Conference, Taipei, Taiwan, 16–19 May 2010; pp. 1–5.
40. Eddine, B.D.; Dos Santos, F.N.; Boulebtateche, B.; Bensaoula, S. Eyelsd a Robust Approach for Eye Localization and State Detection. *J. Signal Process. Syst. Signal Image Video Technol.* **2017**, *90*, 99–125. [[CrossRef](#)]
41. Liu, X.; Tan, X.; Chen, S. Eyes Closeness Detection Using Appearance Based Methods. In *International Conference on Intelligent Information Processing*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 398–408.
42. Song, F.; Tan, X.; Liu, X.; Chen, S. Eyes Closeness Detection From Still Images with Multi-scale Histograms of Principal Oriented Gradients. *Pattern Recognit.* **2014**, *47*, 2825–2838. [[CrossRef](#)]
43. Liu, Z.; Song, X.; Wang P.; Zhou G. An Identification Method of Fatigue Driving Based on Eye Features. *J. Chongqing Univ. Technol.* **2016**, *30*, 11–15.