mathematics

MDPI

Article

Truck Driver Fatigue Detection Based on Video Sequences in Open-Pit Mines

Yi Wang, Zhengxiang He and Liguan Wang *

School of Resources and Safety Engineering, Central South University, Changsha 410083, China; 195512117@csu.edu.cn (Y.W.); hezhengxiang@csu.edu.cn (Z.H.)
* Correspondence: liguan_wang@csu.edu.cn

Abstract: Due to complex background interference and weak space–time connection, traditional driver fatigue detection methods perform poorly for open-pit truck drivers. For these issues, this paper presents a driver fatigue detection method based on Libfacedetection and an LRCN. The method consists of three stages: (1) using a face detection module with a tracking method to quickly extract the ROI of the face; (2) extracting and coding the features; (3) combining the coding model to build a spatiotemporal classification network. The innovation of the method is to utilize the spatiotemporal features of the image sequence to build a spatiotemporal classification model suitable for this task. Meanwhile, a tracking method is added to the face detection stage to reduce time expenditure. As a result, the average speed with the tracking method for face detection on video is increased by 74% in comparison with the one without the tracking method. Our best model adopts a DHLSTM and feature-level frame aggregation, which achieves high accuracy of 99.30% on the self-built dataset.

Keywords: open-pit truck; driver fatigue; feature coding; LRCN

Citation: Wang, Y.; He, Z.; Wang, L. Truck Driver Fatigue Detection Based on Video Sequences in Open-Pit Mines. *Mathematics* **2021**, *9*, 2908. https://doi.org/10.3390/math9222908

Academic Editor: Liangxiao Jiang

Received: 2 October 2021
Accepted: 13 November 2021
Published: 15 November 2021

1. Introduction

Recently, several sizeable open-pit truck accidents have aroused people's attention to driver fatigue detection. Open-pit trucks are among the most critical transportation equipment in surface mines [1]. Because of their high cost and huge size, once an accident occurs, it makes mining enterprises bear huge economic costs. Furthermore, compared to ordinary drivers, truck drivers are more prone to fatigue due to their working mode, driving environment and lifestyle, which results in a significant decrease in driving performance and an increased risk of accidents [2].

Driver fatigue is a behavior relying on timing changes, such as slow blinking, continuous eye closing, yawning, etc. However, traditional methods classify behaviors based on the single frame information. They only analyze features from the image level, such as by using convolutional neural networks (CNN) [3], template matching or binarization [4] to obtain status information of the target and then identifying the state of fatigue by calculating the percentage of eyelid closure over the pupil over time (PERCLOS) [5,6] or the frequency of the mouth (FOM). Burcu and Yaşar [7] applied a multitask CNN model to get face characteristics and calculate the PERCLOS and the FOM to determine driver fatigue. The method has a serious drawback that a single spatial feature is unable to effectively identify video behavior with a lack of temporal information across frames in a video [8].

A deep learning method based on video behavior is regarded as a powerful behavior recognition method. According to the network architecture, it can be roughly categorized into three types: two-stream convolutional networks (two-stream ConvNets), 3-dimensional convolutional networks (3D ConvNets) and fusion method.

The famous two-stream architectures for video behavior have been proposed by Simonyan [9]. The method applies two branch networks separately to a static frame and the dense optical flow to extract features. Then, the method fuses the motion features in

time and space to obtain the expression of video behavior. Furthermore, this method is improved by considering feature fusion [10], multiple convolution isomerism [11] and convolution feature interaction [12], which make better use of the spatiotemporal feature information for video behavior learning. However, it can only process the input of short-term actions and has an insufficient understanding of the time structure of long-term actions [13].

Compared with two-stream networks, 3D ConvNets can better extract spatiotemporal features without calculating the dense optical flow. In 2013, Shuiwang et al. [14] proposed a method using 3D CNN for video behavior recognition, which captures motion features in the time dimension through multi-frame stacking. Tran et al. [15] proposed a 3D ConvNet to extract motion features through multiple convolution kernels in two dimensions of space and time. However, the number of parameters in a 3D convolutional network (C3D) is too huge. To reduce the computational cost and improve the network, Carreira et al. [16] proposed an inflated 3D ConvNet (I3D) by combining two-stream convolution with a 3D ConvNet. Diba et al. [17] constructed a new time layer on a 3D CNN and proposed a temporal 3D ConvNet (T3D). Qie et al. [18] decoupled the 3D convolution kernels into 2D and 1D convolution and proposed pseudo-3D residual networks (P3D). Although the 3D ConvNet method has been improved in many aspects, for long-time motion tasks, it is still not sufficient to extract the motion function of the target through the correlation between the sequence information pieces learned by the 3D ConvNet [19].

Unlike CNNs, recursive neural networks (RNNs) have a good time sequence processing ability. Ed-Doughmi et al. [20] applied an RNN over a sequence of driver's face frames to anticipate driver drowsiness. Long short-term memory neural networks (LSTMs) [21] constitute an improvement of RNNs. It is not only good at dealing with long-term dependence relationships, but also solves the problem of vanishing and exploding gradients [22]. Donahue [23] made a great breakthrough in the behavior classification problem of long-time video sequences. They combined a CNN network and an LSTM network to form a long-term recurrent convolutional network (LRCN). In this way, the network can simultaneously extract features of a single frame from a CNN and process long-duration videos with an LSTM. Although the general video behavior recognition methods can complete most tasks well, the face fatigue behavior, unlike the centralized expression form of large movements, is relatively static, which means that dynamic changes of the face are not easy to capture and are often strongly dependent on temporal changes. An LSTM is a typical time series processing network, and its accuracy and efficiency are far higher than those of CNNs for the processing of time series problems. CNNs have strong image classification and feature extraction capabilities [24]. Linking both networks can help fully extract the spatiotemporal feature expression of facial actions and realize driver fatigue detection using long-term sequence actions in a video. Chen et al. [25] proposed a method based on the key facial points and an LSTM, which is similar to the method proposed in this paper, but it is not suitable for multiangle face detection. The method proposed in this paper has made corresponding improvements in face detection and temporal feature learning networks and performs well for the purposes of this paper.

This paper proposes an open-pit truck driver fatigue detection method based on an open-source library for face detection (Libfacedetection) and an LRCN. The method takes the spatiotemporal fatigue features of the face as the detection target. First, the face detection algorithm is used to detect the face in the image sequence and quickly extracts two regions of interest (ROI) which are the eye and the mouth from the complex scene. Second, the image sequence processed by the face detection algorithm is input into an LRCN. Finally, the image sequence is encoded as feature vectors by a CNN and an LSTM network is used to learn the temporal features of fatigue from the feature vectors to realize the classification task.

2. Methods

The method proposed in this paper mainly consists of three modules, including a face detection and tracking module, a feature coding module and a temporal classification module. Figure 1 shows the overall flow of the method.

- Extracting the frames from the video. The method needs to locate the key facial points and extract the ROI of the eye and the mouth one by one through the face detection and tracking module.
- Extracting features of the ROI sequence through a CNN and encoding the feature information to construct feature vectors through a frame aggregation method.
- Inputting the feature vector sequence into a double-hidden long short-term memory neural network (DHLSTM) to learn time sequence features and then make a global decision on the video sequence to predict whether the driver is fatigued.

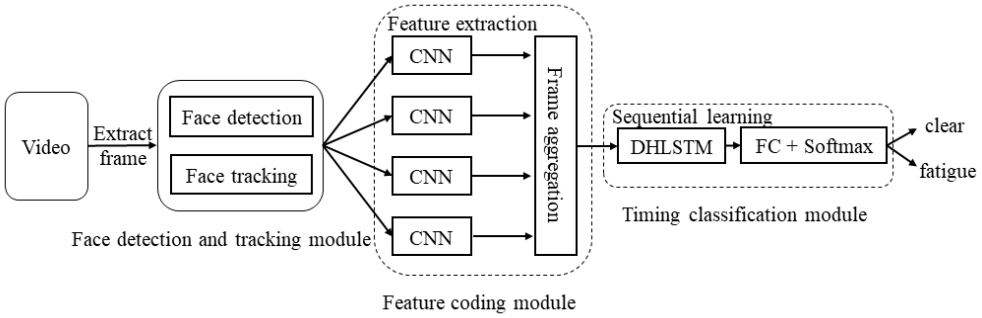


Figure 1. Overall flowchart of the method.

2.1. Face Detection and Tracking

There are some challenges for face detection in open-pit trucks. Firstly, the proportion of the face in the image is small, and the complex background interferes greatly with face detection. Secondly, due to irregular installation of cameras on different trucks and rotation of the driver's head, the orientation of the face in the video sequence varies widely. When dealing with such a challenging task, the general face detection algorithms are not able to take into account the speed and accuracy at the same time.

Libfacedetection is an open-source library for image face detection which adopts a lightweight face detection algorithm based on the SSD architecture proposed by Yu Shiqi. The algorithm is robust and suitable for face detection in complex backgrounds. Moreover, it can detect multiangle faces fast and accurately.

Although face detection speed in a single frame is very fast, it is still very time-consuming to detect the face in an image sequence frame by frame without considering the connection between the previous and subsequent frames. In an image sequence, the driver's actions are continuous in the time domain and change relatively slowly in the space domain. According to these features, this paper integrates a tracking method to optimize the face detection module. The method tracks the face regions of adjacent frames by utilizing the spatiotemporal relationship between the adjacent frames. Figure 2 shows the overall workflow of the algorithm.

The method needs to obtain the previous frame detection result before the current frame is processed. The centerpoint of the bounding box belonging to the previous frame is taken as the prediction centerpoint of the face region in the current frame. According to this centerpoint, this method doubles the bounding box of the previous frame to obtain a new bounding box and uses the new bounding box to crop the current frame. The cropped image is input into the face detection model. In the dataset, the size of the face in the picture has a lower limit, and the ratio of the side length of the next frame to the side length of the previous frame should not be lower than a certain threshold. Thus, by judging the side length of the bounding box and the size of the bounding box in two adjacent frames, the prediction bounding box of the current frame is adjusted accordingly. The method

limits the detection region of the next frame through the connection between the adjacent frames, which can effectively eliminate background interference and greatly reduce the computational cost.

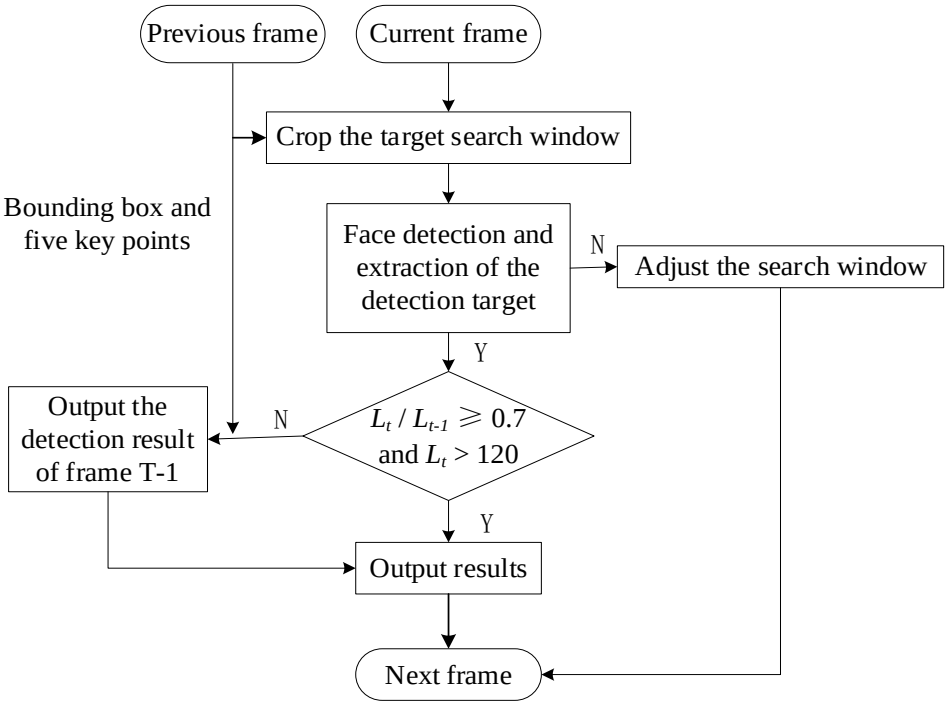
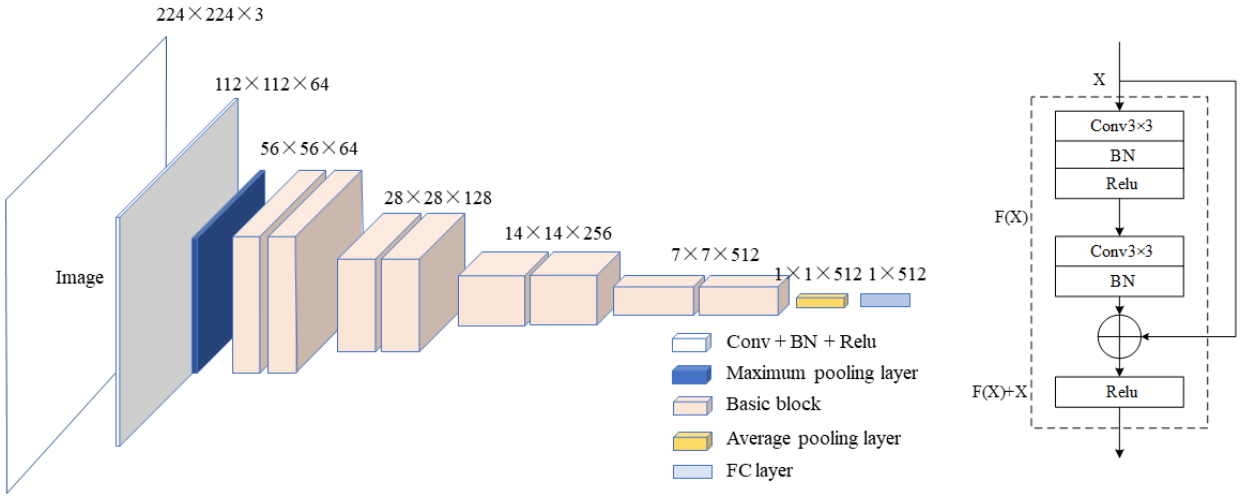


Figure 2. Flowchart of the face tracking and detection algorithm. L_t is the side length of the bounding box in current frame; 120 means that the side length of the bounding box should be greater than 120; 0.7 means that the ratio of the side length of the next frame to the side length of the previous frame should be over 0.7.

2.2. LRCN for Fatigue State Classification

An LRCN is a network constructed by combining a CNN and an LSTM, which has the ability of spatial feature extraction and long-term sequence learning. This paper proposes a network structure by combining Resnet and a DHLSTM to deeply explore the spatiotemporal features of driver fatigue.

Residual network is the most widely used CNN network at present. It adds residual units based on Vgg19 [26] and makes a skip connection between every two convolutional layers to form residual learning, which makes Resnet a great success in solving the problems of gradient disappearance and gradient explosion in deep networks. It maintains the advantage of deep networks in image feature excavation. Taking account of the complexity of features and the size of the model, Resnet18 is used as the feature extraction network to process video frames. The Resnet18 structure consists of 18 network layers, including the convolution layer with a convolution kernel of 7×7, the maximum pooling layer, eight basic blocks, the average pooling layer and the full connection layer. Figure 3a shows the structure. Each basic block is composed of two convolution layers with a convolution kernel of 3×3. Each convolution is configured with a batch normal (BN) layer and a Relu layer. Figure 3b shows a basic block.



(**a**) (**b**)

Figure 3. Resnet network. (**a**) It consists of 18 network layers, including the convolution layer, the maximum pooling layer, four basic blocks, the average pooling layer and the full connection layer. (**b**) A basic block consists of two convolution layers, two BN layers and two Relu layers [27].

The learning principle of residual units can be defined as follows.

$$y = F(x, \{W_i\}) + W_s x \tag{1}$$

A long short-term memory (LSTM) network is an improved RNN. Unlike convolutional networks which are better for processing single images, LSTMs are good at dealing with long-time dependent problems. An LSTM is composed of multiple units. Each unit contains three essential parts, which are the input gate, the forget gate and the output gate. Through the gates, an LSTM can integrate and filter the input information of multi-moments to achieve long-term memory. Figure 4 shows the unit structure.

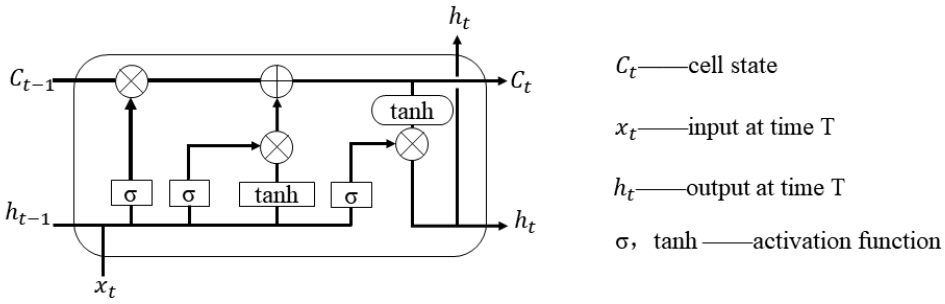


Figure 4. LSTM unit structure [28].

The gates are defined as follows.
Input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

Forget gate:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{5}$$

Output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tan h(C_t) \tag{7}$$

where i_t, f_t and o_t represent the attenuation coefficient of learning different memories, respectively, C_{t-1}, $\widetilde{C}_t$ represent the memory learned before time T and the memory learned at the current moment, respectively, C_t represents the memory state at time T and h_t represents the network output at time T.

A DHLSTM is a variant of an LSTM. It adds a layer of hidden cells to the original LSTM. Two hidden layers are stacked for calculation, which is beneficial for the network to excavate more information between the sequences [29]. In addition, an LSTM has variants such as gated recurrent (GRU) and bidirectional long short-term memory (Bi-LSTM) networks.

2.3. Feature Coding Strategy

As shown in Figure 5, the hidden states of all moments in a DHLSTM are collected as features of the input sequence and then input into the classifier for time sequence classification [30]. The hidden states at each moment represent the learning situation of the input data at the current moment. Before learning these features, the data cannot be directly input into a DHLSTM in the form of image sequences, and they need to be converted to a vector sequence when a DHLSTM processes the time sequence data of the image type. This process is called coding.

The coding of image sequences involves frame aggregation and uses a feature vector to represent the image sequence over a period of time [31]. Different coding strategies of the image sequence obtain different feature representations, which affect the classification results of the DHLSTM model and the performance of the whole network. Thus, this paper proposes three coding strategies based on feature-level and decision-level frame aggregation. Figure 6 shows the structure of each strategy.

Figure 6a shows the structure of feature-level frame aggregation. The coding strategy performs convolution processing on each frame to obtain a feature map containing information on edges and shapes. The feature map of each frame is converted into a 512-dimensional feature vector and stacked across time. This coding strategy can retain the representation of most original image information so that the LSTM can pay attention to more details when learning time sequence features. The initial parameters of the model are obtained through transfer learning. The parameters of the convolutional layers in the pretraining model [24] are frozen as the initial parameters used to extract the spatial features of the eye–mouth stitching images. Then, the convolution layer and the top layer are jointly trained to update the network weight.

Decision-level frame aggregation is a process that integrates the decision information of multiple feature classifiers. As shown in Figure 6b,c, this paper constructs a dual-convolution network to classify eye features and mouth features, respectively. The strategy obtains two classification probabilities through the two classifiers. Then, all local information is encoded through different feature fusions. In the process of time sequence learning, the classification results of multiple features can change from local classification decisions to a global classification decision. There are two strategies based on different feature fusions. Figure 6b shows decision-level frame aggregation based on vector stitching. After obtaining two two-dimensional vectors from the eye classifier and the mouth classifier, the strategy fuses the features of the two vectors by stitching the vectors and stacks them across time. Figure 6c shows another strategy which adopts the method of the feature vector dot product for feature fusion and coding. Both training methods are the same. Firstly, the eye and mouth classifiers are trained through the two self-built datasets of the eye and the mouth. To obtain classification results of the driver's eye status (open or closed) and mouth status (open or closed), the Resnet18 pretrained model is used to train the eye and

mouth classification models. Then, the output results of the classifiers are aggregated and input into the DHLSTM for global training.

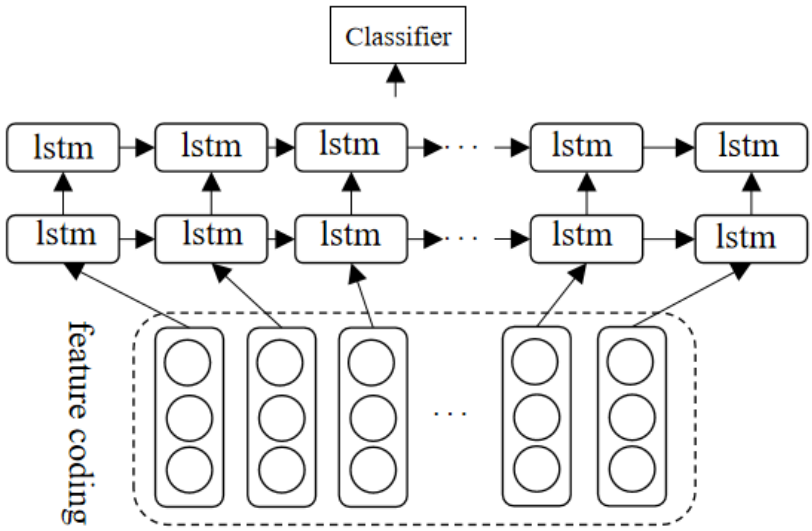


Figure 5. DHLSTM time sequence classifier [30].

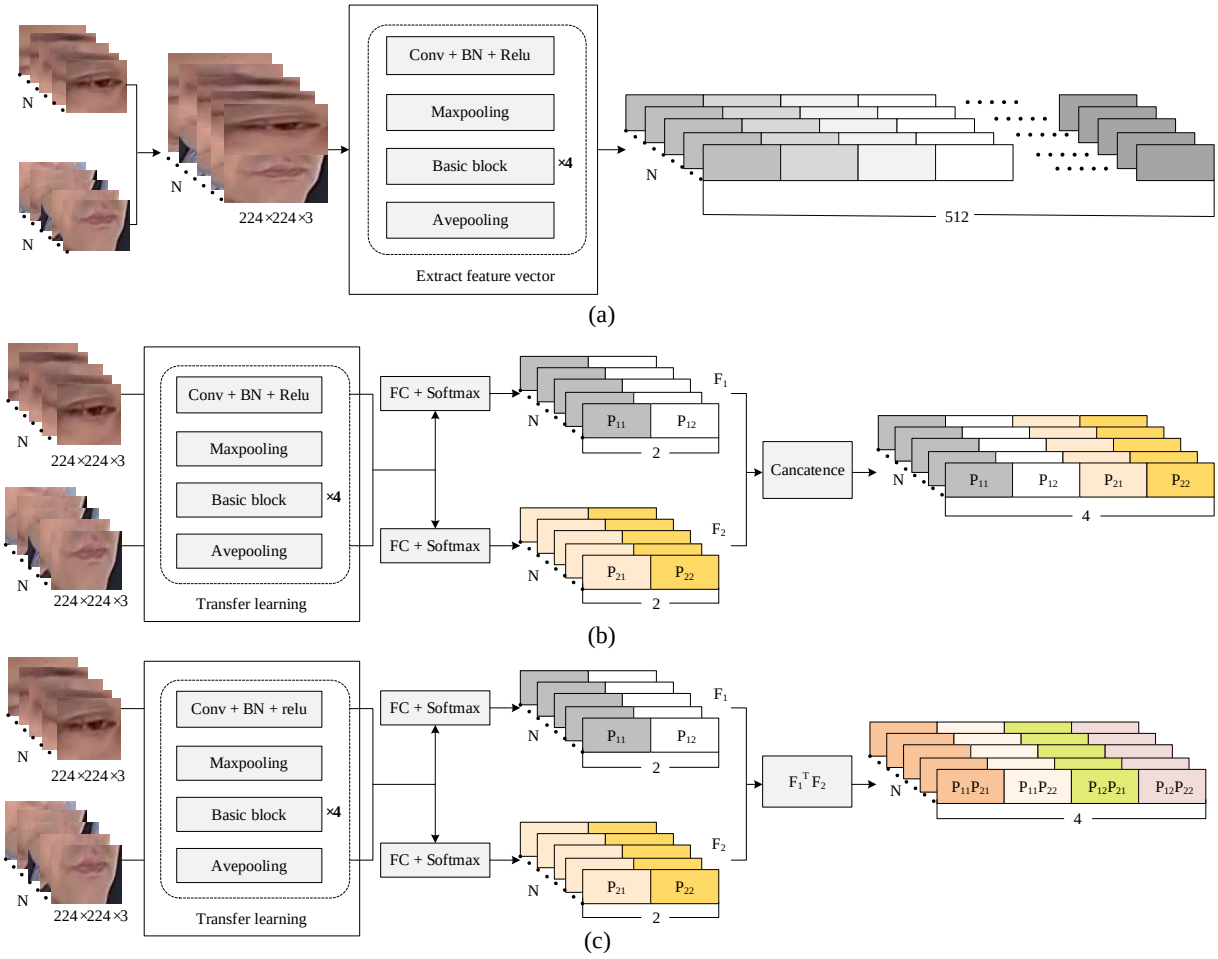


Figure 6. Feature coding strategies. (**a**) Feature-level frame aggregation. (**b**) Decision-level frame aggregation based on vector stitching. (**c**) Decision-level frame aggregation based on the vector dot product.

3. Results

3.1. Data Preparation

The data of truck drivers came from open-pit mines. We obtained more than 50 recorded videos of truck drivers from different trucks. The video duration varied from 3 min to 5 min, including natural driving states such as normal driving, yawning, slow blinking, frequent blinking, talking, laughing and so on. The frame rate of the video was 25 fps. At the preprocessing stage, the video stream was first parsed into image frames, and the frame sequences were filtered to remove the frames where no face appears or the face is occluded. Then, a sample was constructed from the valid video sequences with every 30 frames; 354 awake samples and 338 fatigue samples were obtained. The driver usually faces the camera with his side face when driving. Therefore, we only preserved the regions of the right eye and the mouth after face detection and angle correction. Figure 7 shows the process.

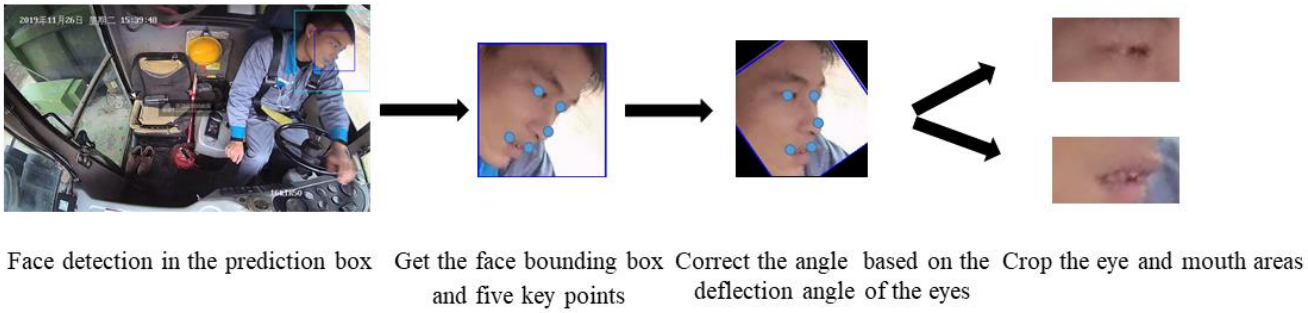


Figure 7. Sample preparation and processing.

We adopted the operation of data augmentation. The operation randomly selected one of five methods: Gaussian blur, median blur, mean blur, box filter and bilateral filter to blur the video sequence. The operation performed a random center rotation of 0–15° on the image. The operation added salt and pepper noise to the images. For the classification label of every sample, this paper obtained qualitative judgment by observing the facial state of the driver while driving. The data were spilt into the training and test samples according to the ratio of 0.75 to 0.25, and then we augmented the divided dataset to obtain 2597 samples in the training set and 863 samples in the test set. The training set contained 1328 awake samples and 1269 fatigue samples. The test set contained 442 awake samples and 421 fatigue samples.

The public dataset NTHU Drowsy Driver Detection Dataset (NTHU-DDD) [32] was also used in this experiment. It contains the driving state videos of different volunteers in the simulated environment with a fixed lens which are used for the comparative evaluation of different time sequence learning networks in this task. The expression of the public dataset in terms of time domain features was consistent with the self-built dataset. The main difference was the distribution of spatial features, such as video scenes, facial expressions, light and other environmental interference factors. Therefore, it was feasible to use the public dataset to compare the performance of different time sequence learning networks.

3.2. Result

To accelerate the convergence speed of the model, the paper used the optimization strategy of the learning rate gradient descent, and the learning rate was decreased by one order of magnitude for each 30 epochs. All the models adopted a cross-entropy loss function as follows:

$$Loss = \frac{1}{N}\sum_{i} -[y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)] \quad (8)$$

where y_i is the true label of the sample, 1 represents fatigue, 0 represents clear, p_i is the probability that the sample is predicted to be fatigued.

Table 1 shows the parameters set in the network.

Table 1. Parameter settings of the network.

Parameter	Feature-Level Frame Aggregation	Decision-Level Frame Aggregation
Input size	30*224*224*3	30*224*224*3
Epoch	100	100
Batch size	12	12
Learning rate	0.00005	0.00005
Latent dim	512	4
Hidden size	256	128
Optimizer	adam	adam

There were three different schemes: one was a time sequence classification model based on feature-level frame aggregation and the other two were time sequence classification models based on decision-level frame aggregation. Two types of time sequence classification models based on decision-level frame aggregation were, respectively, the model based on vector stitching and the model based on the vector dot product. Models 1, 2 and 3 were used to represent the three schemes.

Figure 8a shows the changes of loss and accuracy during the training process of model 1. In the initial stage of the iteration, the loss and accuracy of the training set and the test set oscillated significantly. After 30 epochs, the loss and accuracy gradually stabilized. The learning rate gradient descent strategy was used, decreasing the learning rate by one order of magnitude every 30 epochs. After 100 epochs of training, the loss finally converged to about 0.3, and the accuracy converged to about 99%.

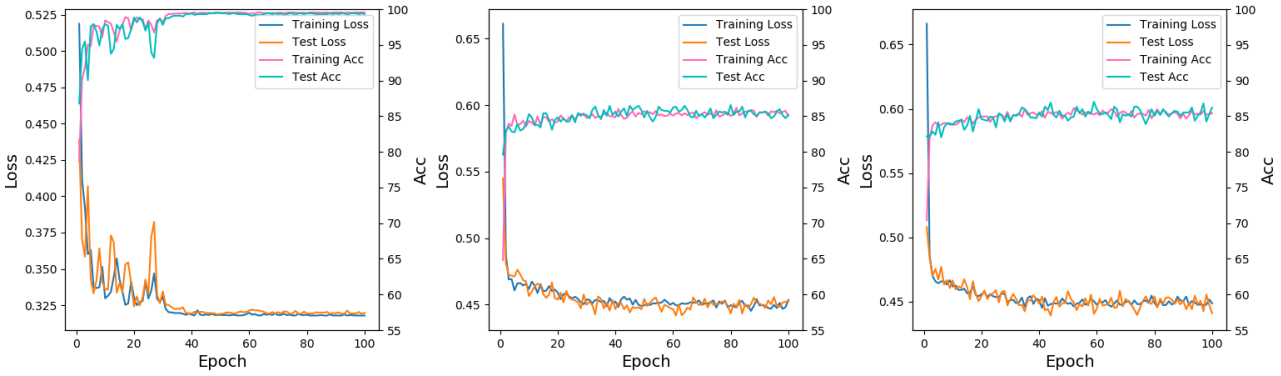


Figure 8. Loss and accuracy change curve. (**a**) Training process of model 1; (**b**) Training process of model 2; (**c**) Training process of model 3.

The training process of model 2 is shown in Figure 8b. At the initial training stage, the loss of the training set and the test set decreased rapidly and tended to converge, while the accuracy rose. After 100 epochs of training, the loss of the training set finally converged to about 0.4, and the accuracy was stable at about 85%. As shown in Figure 8c, the change trend of the loss and accuracy curves of model 3 are roughly the same as those of model 2.

This paper uses two parameters of precision and recall to evaluate the experimental results. Precision is defined as the fraction of predictions that are accurate, and recall is defined as the fraction of instances that are accurately predicted.

Table 2 and Figure 9, respectively, show the accuracy results by using different coding strategies and the classification results by using different coding strategies. The accuracy

of the test set for models 1, 2 and 3 was 99.30%, 85.98% and 86.21%, respectively. In model 1, the detection precision of each class was 99.32% and 99.29%, respectively, and their recall rates were 99.32% and 99.29%. In model 2, the detection precision of each class was 89.05% and 83.19%, respectively, and their recall rates were 82.81% and 89.31%. In model 3, the detection precision of each class was 89.29% and 83.41%, respectively, and their recall rates were 83.03% and 89.55%.

Table 2. Accuracy comparison of the three algorithms.

Model	Training Set Accuracy (%)	Test Set Accuracy (%)
1	99.5379	99.3048
2	85.0597	85.9791
3	85.4447	86.2109

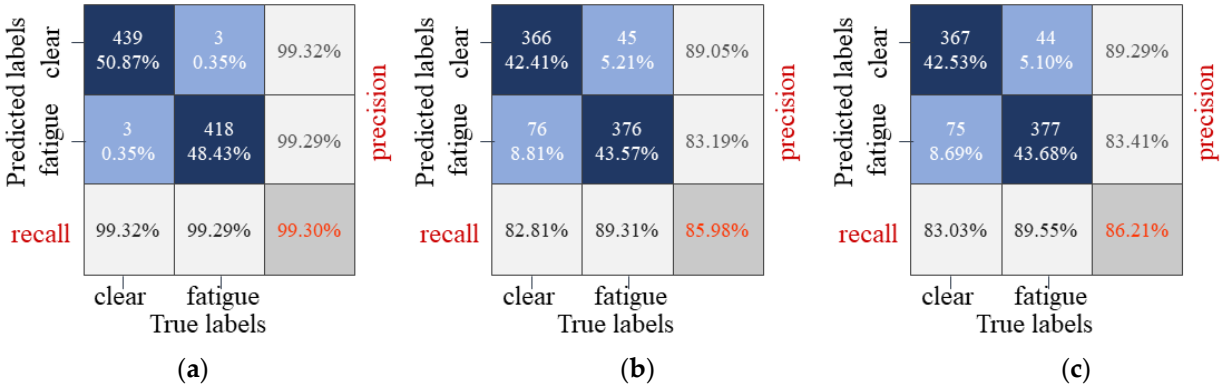


Figure 9. Confusion matrix diagram. (**a**) Classification result of model 1; (**b**) classification result of model 2; (**c**) classification result of model 3.

4. Discussion

4.1. Training Process

Loss and accuracy of three schemes above performed well in training. Among them, model 2 and 3 were faster than model 1 in convergence speed when training. Meanwhile, the loss and accuracy of models 2 and 3 preformed more stably at the initial training stage, but model 1 was more accurate in the training set and the test set.

4.2. Accuracy

It is worth noting that it was impossible to obtain the real label of the driving status in our task. Based on the general facts, this paper makes a qualitative judgment to label samples by observing the driver state, such as slow blinking, continuous eye closing, yawning, etc. The time sequence classification model based on feature-level frame aggregation performed better than the time sequence classification model based on decision-level frame aggregation in terms of accuracy and recall rate. It may be that frame aggregation based on the feature level can obtain more time sequence information in this task, which helps the DHLSTM network to deeply excavate the time sequence features of the target object. However, the information obtained by means of decision-level frame aggregation is relatively singular; the accuracy of the global classification decision is limited by the accuracy of local classifiers. Therefore, it is difficult to improve the accuracy of models 2 and 3 to a higher level. Model 1 was better suited to driver fatigue detection in this task.

4.3. Time Sequence Classification Models

Different time sequence learning networks have different performance in excavating the time domain information, which has a great impact on the classification accuracy. In order to find a model to better excavate time sequence features, this paper compares multiple networks. Due to the small sample size of the self-built dataset, there were certain

limitations. The public dataset was used to test the performance of different time sequence learning models based on feature-level frame aggregation. The results are shown in Table 3.

Table 3. Comparison of different time sequence classification models based on feature-level frame aggregation.

Model	Training Set Accuracy (%)	Test Set Accuracy (%)
Resnet + LSTM	94.7692	84.4749
Resnet + GRU	95.5266	88.8127
Resnet + BiLSTM	95.1953	85.3120
Resnet + DHLSTM	94.9112	92.6080
Resnet + double-hidden layer BiLSTM	95.1953	90.0304

According to Table 3, the accuracy of the time sequence learning model based on the several LSTM variants in the training set above was over 90%. Among them, the model composed of GRU and Resnet had the best fitting effect on the training set, and the accuracy was 95.5266%. The model composed of a DHLSTM and Resnet performed best on the test set with the accuracy of 92.6180%.

The accuracy difference of the training set and the test set can reflect the generalization ability of the model. The model composed of an LSTM and Resnet and the model composed of a Bi-LSTM and Resnet showed a larger accuracy difference than the other models, and the accuracy of the two kinds of models was 10.3% and 9.8%, respectively, which represents a trend of overfitting. The time sequence learning model with a DHLSTM network showed the smallest accuracy difference and the best fitting effect. Therefore, it can be seen that a DHLSTM performs better in terms of exploring time sequence features.

4.4. Model Speed

A tracking method was added to optimize face detection, which was helpful to improve detection accuracy and reduce detection time. The detection speed and the false detection rate are shown in Table 4, and the detection time expenditure of each module after optimization is shown in Table 5.

Table 4. Comparison of face detection with tracing method and without tracing method.

Model	Detection Time (Singe Frame)	False Face Detection Rate (%)
Without the tracing method	0.2886 s	16.7
With the tracing method	0.075 s	3.33

Table 5. Detection time expenditure of each module.

Model	Detection Time (30 Frames)
Face detection and tracking module	2.25 s
Encoding module based on feature-level frame aggregation	0.517 s
Time sequence classification module	0.076 s

Table 4 shows the time expenditure and accuracy results of face detection with the tracking method and without the tracking method. When detecting the same image sequence, the average detection time without the tracking method was 0.2886 s per frame and the average detection time with the tracking method was 0.075 s per frame. The time difference is obvious. In comparison with the face detection module without the tracking method, the face detection module with the tracking method reduced the detection time expenditure by 74%, and the false face detection rate decreased from 16.7% to 3.33%.

Table 5 shows the time expenditure of each module for a sample. The total inference time for detecting a 30-frame image sequence was about 2.85 s. Among them, the detection

time of the face detection and tracking module was 2.25 s per 30 frames. The time of the coding module based on feature-level frame aggregation was 0.517 s per 30 frames. The time of the time sequence classification module was 0.076 s per 30 frames.

It can be seen from the time analysis that the detection time was mainly concentrated in the face detection module. It takes a long time because a scene on the paper is more complex than other scenes, such as the face angle relative to the camera, light and other environmental interference factors. In order to speed up, this paper proposes a tracking method, which shows a significant reduction in time expenditure. Although the optimized face detection module had a great improvement in detection speed, the real-time performance still needs further research.

5. Conclusions

This paper proposed a video-based driver fatigue detection method for open-pit truck drivers. The method can overcome the interference caused by a complex environment. The innovation of this paper is to combine Resnet with a DHLSTM to build a spatiotemporal network model suitable for this task. Resnet extracts the spatial features of each frame, and then the extracted spatial features are aggregated and input into a DHLSTM for temporal feature learning and classification. Meanwhile, the paper adds a tracking method to the face detection module to reduce the detection time expenditure. The face region is tracked by utilizing the spatiotemporal relationship between the adjacent frames. The experimental results show that the method can effectively detect the driver fatigue behavior in image sequences. In comparison with the face detection module without the tracking method, the time expenditure of the proposed method is reduced by 74% at the face detection stage, which is better suited to face detection in complex environments. Moreover, the LRCNs composed of Resent and a DHLSTM perform better in features excavation and generalization than LSTM, GRU and Bi-LSTM and can achieve more accurate classification results. The paper determined driver fatigue on the basis of video sequences of 30 frames. The results show that the method can meet the requirements of practical applications to a certain extent in terms of accuracy and speed. When combined with other necessary hardware equipment, this method can be deployed for application in practice.

The method proposed in this paper still has certain limitations. The method does not classify the level of fatigue. In the future, this method will be improved to classify the level of driver fatigue. For occlusion situations, such as mouth covering, eye rubbing, etc., this paper does not make an effective response. Future research will pay more attention to unobstructed parts when the face is occluded to detect driver fatigue. The optimized face detection stage has a great reduction in detection time expenditure, but the real-time performance still needs further improvement.

Author Contributions: Y.W., L.W. and Z.H. conceived, designed and performed the experiments. Y.W. analyzed the data, wrote the manuscript and implemented the source code. Z.H. revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Key R&D Program of China (2017YFC0602905).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors would like to thank the public datasets used in this research. We also thank the reviewers for their comments and suggestions to improve the quality of the paper.

Conflicts of Interest: The authors declare no competing interests.

References

1. Sun, E.J. The fuzzy neural network based haul truck driver fatigue detection in surface mining. In Proceedings of the 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, Guilin, China, 29–31 July 2017; pp. 1522–1526.
2. Åkerstedt, T.; Peters, B.; Anund, A. Impaired alertness and performance driving home from the night shift: A driving simulator study. *J. Sleep Res.* **2010**, *14*, 17–20. [CrossRef] [PubMed]
3. Fang, Z.; Su, J.; Lei, G. Driver Fatigue Detection Based on Eye State Recognition. In Proceedings of the 2017 International Conference on Machine Vision and Information Technology, Singapore, 17–19 February 2017; pp. 105–110.
4. Wang, T.; Shi, P. Yawning detection for determining driver drowsiness. In Proceedings of the IEEE International Workshop on Vlsi Design and Video Technology, Suzhou, China, 28–30 May 2005; pp. 373–376.
5. Dinges, D.F.; Grace, R. *PERCLOS: A Valid Psychophysiological Measure of Alertness as Assessed by Psychomotor Vigilance*; Tech Brief: New York, NY, USA, 1998.
6. Grace, R.; Byrne, V.E.; Bierman, D.M. A drowsy driver detection system for heavy vehicles. In Proceedings of the 17th Digital Avionics Systems Conference, Bellevue, WA, USA, 31 October–7 November 1998; pp. 1361–1368.
7. Burcu, K.S.; Yaşar, B. Real Time Driver Fatigue Detection System Based on Multi-Task ConNN. *IEEE Access* **2020**, *8*, 12491–12498.
8. Huang, H.X.; Wang, R.P.; Liu, X.Y. Review of Human Action Recognition Technology Based on 3D Convolution. *Comput. Sci.* **2020**, *47*, 139–144.
9. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the 28th Conference on Neural Information Processing Systems, Montreal, QU, Canada, 8–13 December 2014; pp. 1–8.
10. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 1933–1941.
11. Wang, L.; Xiong, Y.; Wang, Z. Towards Good Practices for Very Deep Two-Stream ConvNets. 2015. Available online: https://arxiv.org/abs/1507.02159 (accessed on 8 July 2015).
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 770–778.
13. Wang, L.; Xiong, Y.; Wang, Z. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 20–36.
14. Ji, S.; Xu, W.; Yang, M. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal.* **2013**, *35*, 221–231. [CrossRef] [PubMed]
15. Tran, D.; Bourdev, L.; Fergus, R. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 4489–4497.
16. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4724–4733.
17. Diba, A.; Fayyaz, M.; Sharma, V. Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification. 2017. Available online: https://arxiv.org/abs/1711.08200v1 (accessed on 22 November 2017).
18. Qiu, Z.; Yao, T.; Mei, T. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In Proceedings of the 16th IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5534–5542.
19. Dai, W.; Chen, Y.; Huang, C. Two-Stream Convolution Neural Network with Video-stream for Action Recognition. In Proceedings of the 2019 International Joint Conference on Neural Networks, Budapest, Hungary, 14–19 July 2019; pp. 1–8.
20. Ed-Doughmi, Y.; Idrissi, N.; Hbali, Y. Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network. *J. Imaging* **2020**, *6*, 8. [CrossRef] [PubMed]
21. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
22. Bengio, Y. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **2002**, *5*, 157–166. [CrossRef] [PubMed]
23. Donahue, J.; Hendricks, L.A.; Guadarrama, S. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
24. Jia, D.; Wei, D.; Socher, R. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 20–25 June 2009; pp. 248–255.
25. Chen, L.; Xin, G.J.; Liu, Y.L. Driver Fatigue Detection Based on Facial Key Points and LSTM. *Secur. Commun. Netw.* **2021**, *2021*, 1–9. [CrossRef]
26. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. Available online: https://arxiv.org/abs/1409.1556 (accessed on 10 April 2015).
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645.
28. Cai, R.; Li, S.; Tian, J.; Ren, L. Short-Term Load Forecasting Based on Electricity Price in LSTM in Power Grid. *IOP Conf. Ser.* **2019**, *569*, 042–046. [CrossRef]

29. Wei, X.; Le, Y.; Han, J.H.; Lu, Y. Vehicle behavior dynamic recognition network based on long short-term memory. *J. Comput. Appl.* **2019**, *39*, 1894–1898.
30. Hu, Y.; Luo, D.Y.; Hua, K. Overview on deep learning. *CAAI Trans. Intell. Syst.* **2019**, *14*, 1–19.
31. Shan, L.; Deng, W. Deep Facial Expression Recognition: A Survey. *IEEE Trans. Affect. Comput.* **2020**, 1. Available online: https://ieeexplore.ieee.org/abstract/document/9039580 (accessed on 1 July 2021).
32. Weng, C.H.; Lai, Y.H.; Lai, S.H. Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 117–133.