Information Fusion 89 (2023) 121–142

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Distracted driving detection based on the fusion of deep learning and causal reasoning

Peng Ping [a], Cong Huang [a], Weiping Ding [b,*], Yongkang Liu [c], Miyajima Chiyomi [d], Takeda Kazuya [e]

[a] School of Transportation and Civil Engineering, Nantong University, Nantong, Jiangsu, 226001, China
[b] School of Information Science and Technology, Nantong University, Nantong, Jiangsu, 226001, China
[c] Department of Electrical and Computer Engineering, University of Texas at Dallas, Dallas, TX 75080, TX, USA
[d] School of Informatics, Daido University, Nagoya, Aichi, 457-8530, Japan
[e] School of Informatics, Nagoya University, Nagoya, Aichi, 464-0841, Japan

ARTICLE INFO

Keywords:
Behavior recognition
Driving distraction detection
Deep learning network
Causal reasoning
Counterfactual reasoning

ABSTRACT

Distracted driving is one of the key factors that cause drivers to ignore potential road hazards and then lead to accidents. Existing efforts in distracted behavior recognition are mainly based on deep learning (DL) methods, which identifies distracted behaviors by analyzing static characteristics of images. However, the convolutional neural network (CNN) — based DL methods lack the causal reasoning ability for behavior patterns. The uncertainty of driving behaviors, noise of the collected data, and occlusion between body agents, bring additional challenges to existing DL methods to recognize distracted behaviors continuously and accurately. Therefore, in this paper, we propose a distracted behavior recognition method based on the Temporal–Spatial double-line DL network (TSD-DLN) and causal And-or graph (C-AOG). TSD-DLN fuses the attention feature extracted from the dynamic optical flow information and the spatial feature of the single video frame to recognize the distracted driving posture. Furthermore, a causal knowledge fence based on C-AOG is fused with TSD-DLN to improve the recognition robustness. The C-AOG represents the causality of behavior state fluent change and adopts counterfactual reasoning to suppress behavior recognition failures caused by frame features distortion or occlusion between body agents. We compared the performance of the proposed method with other state-of-the-art (SOTA) DL methods on two public datasets and self-collected dataset. Experimental results demonstrate that proposed method significantly outperforms other SOTA methods when acquiring distracted driving behavior by processing consecutive frames. In addition, the proposed method exhibits accurate continuous recognition and robustness under incomplete observation scenarios.

1. Introduction

Over the past decades, distracted driving has become a common phenomenon all over the world, endangering not only distracted drivers, but also passengers, pedestrians or cyclists on the road. If a driver reads a message while driving, his or her eyes are leaving the road for an average of three seconds, at the speed of 60 km per hour, that is like driving the length of about 50 m with blindfolded. According to the report of the US National Highway Traffic Safety Administration (NHTSA), 3142 people were killed in motor vehicle crashes involving distracted drivers in 2019 and more than 420,000 injuries are caused by the distracted driving [1]. Despite the fact that distracted driving can cause serious traffic accidents, the incidence of distracted behavior is still increasing year by year [2,3].

With the improvement of vehicle intelligence, more vehicles now have L2, or even L3 level autonomous driving capabilities, vehicles can replace humans to complete more driving tasks [4,5]. However, L3 level autonomous vehicles do not have 100% correct perception of driving scenarios, and their effective disposal scenarios are limited [6]. Prompt take over action under emergency or extreme scenarios can reduce the incidence of accidents effectively. Therefore, for driving assistance systems and autonomous vehicles, accurate access of the driver's distraction state is an important prerequisite for developing safe and effective driver assistance and human–machine switching strategies [7]. Distracted driving includes activities that divert the driver's attention from driving, such as talking or texting on the phone,

* Corresponding author.
E-mail addresses: pingpeng@ntu.edu.cn (P. Ping), c.huang@ntu.edu.cn (C. Huang), ding.wp@ntu.edu.cn (W. Ding), yongkang.liu@utdallas.edu (Y. Liu), miyajima@daido-it.ac.jp (M. Chiyomi), kazuya.takeda@nagoya-u.jp (T. Kazuya).

https://doi.org/10.1016/j.inffus.2022.08.009
Received 12 November 2021; Received in revised form 19 July 2022; Accepted 9 August 2022
Available online 13 August 2022

eating or drinking, talking to passengers in the vehicle, adjusting the stereo, entertainment or navigation system — any activity that takes the driver's attention away from safe driving. Since most of the distracted driving behavior can be captured by the vehicle-mounted cameras, it is natural to analyze and detect distracted driving behaviors through video frames captured from naturalistic driving.

As a powerful pattern recognition method, DL methods have achieved great success in the field of machine vision [8], and are widely used in object detection [9], semantic segmentation [10], and behavior recognition tasks [11]. Inspired by these successful applications, most scholars use DL methods based on the CNN structure to build behavior classification models by learning the image features of various distracted behaviors. Existing efforts mainly focus on improving the recognition accuracy through optimizing DL architecture or pursuing more effective input through feature fusion.

Although some DL-based methods have achieved good recognition performance on some public datasets, two challenges remain unsolved when detecting distracted behaviors during naturalistic driving. One challenge is to perform continuous behavior pattern discrimination while maintain accuracy. Most existing works built their classification model by learning features from manually labeled standard behavior images. Although such standard and convergent features can effectively improve the model's recognition accuracy for specific behavior pattern, the potential data imbalance caused by feature convergence will limit the effective recognition scope of the classifier [12]. Classifiers will have difficulty in yielding the classification boundary of specific behaviors because of the limited recognition range. Therefore, classification models driven by static data will have blurred recognition boundaries and performance drop when detecting the continuous process of distracted driving. Some efforts attempt to improve the depth of static features by extracting different types of them such as skeleton features and HOG features [13,14]. However, in essence, the related optimization methods only removed the redundant features of static images, instead of solving the problem of insufficient features.

As a relevance learning method, DL-based models achieve behavior classification by constructing the correlation between image features and behavior attributes [15]. Only relying on the DL method is difficult to effectively deal with the distortion or failure of behavior recognition caused by image noise and the unpredictability of driver's actions [16]. Therefore, the second challenge is that the rapid sparseness of image features caused by image noises or occlusion between body agents will cut off the correlation and make the prediction results to be distorted or invalid.

To tackle above mentioned two challenges, in this work, we first propose a distracted driving recognition model based on the TSD-DLN. The model uses dynamic optical information to simulate the gaze transition process of the human vision system, and then classifies the distracted driving behaviors through the potential key feature regions represented by the attention distribution. Compared with the traditional CNN-based model, the TSD-DLN can extract inter-frame dynamic information and thus better track the changing process of distracted behaviors. In addition, to suppress the failure of DL models caused by noisy environments and occluded scenes, we constructed a C-AOG that describes the flow of distracted behaviors to enhance the inference ability of the DL model. Furthermore, by fusing the TSD-DLN and C-AOG through counterfactual reasoning, we construct a suppression mechanism for distorted behavior recognition, which improves the robustness of the behavior recognition method.

This paper makes the following contributions.

(1) From the perspective of the behavior recognition feature, we proposed to fuse the dynamic optical flow of stacked frames and the static spatial feature of each single frames in order to improve the feature dimension and depth for the behavior classification.

(2) From the perspective of the causality inference, we developed a causal inference model based on C-AOG to describe the change process of driver distracted behavior. The model establishes causal relationships between the atomic actions extracted from skeleton feature and behavioral fluent through unsupervised iterative learning.

(3) From the perspective of the pattern recognition by information fusion, we propose a counterfactual inference mechanism, which makes consistent derivation of behavioral fluent in logical and temporal dimensions. By fusing deep learning models with C-AOG through a counterfactual inference mechanism, our behavior recognition method can achieve competitive performance over other state-of-the-art recognition methods.

The organization of this paper is as follows. Section 2 conducts the literature review of the distracted driving recognition based on the DL and research progress of causal reasoning in behavior pattern recognition. In Section 3, the detail framework of our TSD-DLN and C-AOG are described. The experiment setup and validation dataset are presented in Section 4. The experiment result and discussion are described in Section 5. Finally, the conclusion of this study is summarized in Section 6.

2. Related work

In this section, we will give a brief overview of the recent researches on the distracted driving recognition and the causal-effect reasoning in the field of behavior pattern recognition.

2.1. Recent process in the distracted driving behavior recognition

Real-time distracted driving behavior can be recognized by analyzing vehicle status data, driver's physiological state, and driver's gesture or posture. Vehicle status data indirectly reflects the driver's attention state when driving. Distracted driving often leads the vehicle into abnormal driving state such as deviate from the centerline of the road or the abnormal speed fluctuations. Tango et al. analyzed driving data consisting of seven vehicle state parameters to identify distracted driving behavior by various machine learning methods. Experimental results showed that SVM can achieve the best classification rate of 95% compared to other machine learning methods [17]. Aksjonov et al. used a fuzzy logic-based algorithm to analyze nine vehicle status parameters [18]. The method identifies the driver's behavioral state when performing the secondary task by fuzzy rules and Euclidean distance, and the experimental results show that the Euclidian-based recognition model outperforms the adaptive neuro-fuzzy inference method in terms of recognition accuracy. The method proposed by Sun et al. established vehicle motion patterns under normal driving by fusing multiple state space models, and then detects distracted driving online based on exponentially weighted moving average and cumulative sum charts. And the performance of the method was validated by identifying the texting behavior of drivers while driving at high speeds [19].

Behavioral analysis methods based on vehicle data mining generally require accurate real-time data. However, some unexpected situations during driving can bring a lot of noise to the collected naturalistic driving data. These noises make it difficult for machine learning-based binary classifiers to identify whether abnormal driving status of the vehicles are generated by distracted driving or caused by the driver's emergency avoidance maneuvers. Besides, it is also difficult for the binary classifier to give the specific behaviors that lead to distracted driving.

To address the shortcomings of acquiring distracted driving behaviors based on naturalistic driving data, some scholars have attempted to identify the distracted state of drivers by directly analyzing the physiological status and the detail driving manipulation. Initially, constrained by limited data samples of drivers' external features, scholars generally used facial features or physiological status, such as eye and head movements [20], Electrooculogram (EOG) [21], Electroencephalogram (EEG) [22,23], galvanic skin responses (GSR) [24] to analyze the

driver's distraction status. Distraction detection methods based on physiological state recognition can effectively identify a driver's distracted state, but these methods require invasive detection devices that may affect the driver's normal driving, and secondly, similar to detection methods based on natural driving data, physiological states only reflect whether the driver is distracted, but not the specific distracting behavior of the driver.

Later, as the cost of image acquisition data decreased and several large naturalistic driving data collection projects were undertaken, image datasets reflecting drivers' driving processes became abundant [25,26]. Meanwhile, inspired by recent successes of DL in the field of machine vision, a large number of scholars adopted DL methods to obtain specific distracted behaviors [8,27]. A part of scholars focus on the structure optimization of DL networks to improve the accuracy of behavior recognition. Omerustaoglu et al. proposed a distracted driving recognition method based on the fusion of CNN and long short term memory (LSTM) network, the CNN network was used to obtain the features of a single image and LSTM was applied to extract the information feature of the in-vehicle sensors [28]. The fusion model identified 9 distracted driving behaviors with an accuracy of 0.87. Xing et al. compared the recognition accuracy of three DL methods, AlexNet, GoogLeNet and ResNet50, on a self-collected dataset which covers seven common driving activities [29]. The experimental results show that AlexNet performs better than the other two methods, achieving 81.6% recognition accuracy. Further, Mase et al. benchmarked the performance of DL methods on distracted behavior recognition by comparing the recognition performance of ten SOTA DL methods on AUC datasets [30]. The best performing method is a fused DL network based on Inception-V3 and Bi-LSTM, which has a recognition accuracy of 91.7%.

On the other hand, in addition to the structure optimization of DL networks, some scholars attempted to improve the learning performance of DL network models by fusing multiple behavioral image features as input features, which in turn further improved recognition accuracy. Jegham et al. proposed a hybrid model based on attention mechanism named depth-based spatial attention network (DSA) [31]. DSA predicts driver distraction behavior by fusing and analyzing RGB features as well as attention features generated from image depth information. The average classification accuracy of DSA in both side view and front view was 75%, which exceeded SOTA DL methods such as VGG-19 and MCN. The authors also point out that the images acquired in the front view are vulnerable to environmental noise, which could lead to the failure of the DSA method. Dey et al. proposed a context-aware distracted behavior recognition method [32]. Unlike the general behavior recognition method that took the overall behavioral image as the recognition object, the author first identified the distribution of objects in the image that may cause distracted driving behavior, and then analyzed the semantic distribution features of the targets through machine learning methods to achieve distracted behavior prediction. By learning the context distribution, the random forest-based method can achieve the best classification accuracy of 94%. Eraqi et al. acquired raw image features, as well as driver's face, hand features and skin semantics through different CNN modules, and then fuses these features by genetic algorithms to recognize multiple distractions [33]. On the AUC dataset, the accuracy of the thinned GA-based behavior recognition model can reach 84.64%. Zhang et al. proposed a deep unsupervised multi-modal fusion model-UMMFN for distracted task detection [34]. The model combines the sensor signal, acoustic signal and the visual signal by the multiple deep learning model. And then a unsupervised ConvLSTM Encoder–Decoder was adopted to predicted the distraction behavior. The model achieves 97.79% in terms of the average accuracy for ten distracting tasks recognition.

Distracted driving recognition based on deep learning has made great progress, but a large number of existing studies are based on static images for behavioral pattern analysis, and few studies consider the recognition of distracted behavioral process features. In actual driving, distracted behavior is a procedural action, and recognition methods based on static features are often difficult to give clear boundaries of behavior, which is crucial for ADAS or automated driving systems to make accurate driver state judgments. Therefore, in our study, we propose a continuous recognition model based on driving dynamic information and attention mechanism to improve the detection accuracy for distracted behavior.

2.2. Causal reasoning in explainable behavior model

Explainable AI and the causability of the DL model are of increasing interest to researchers. Explainable AI focus on the decision relevant parts of DL model i.e the component which could be technically revised to improve the performance of the DL model [35]. Causability of the DL model is the measurable extent to which the human expert can achieve the level of causal understanding [36]. Current researches for DL model mainly focus on the explainability improvement to find better structure or learning components. However, in some pattern recognition area, such as intelligent medical or human behavior analysis, the causability is more important as the potential causality knowledge can lead to a more effective medical decision or behavior assistant system [37]. At present, although DL has greatly improved the accuracy of distracted driving behavior recognition, its essence is establishing a shallow correlation between observed features (image or skeleton feature) and types of distracting behaviors. Once partial observation features are affected by occlusion or noise and become sparse rapidly, DL-based classifiers will suffer a sharp decline in behavior recognition performance due to the lack of reasoning ability for feature changes. Therefore, how to improve the causability and causal inference ability of pattern recognition methods based on deep learning is a challenge in the field of strong artificial intelligence.

Generally, there are two popular causality models to extract the high-level human knowledge: Bayesian network [16,38] and grammar models [39,40]. In recent years, grammar model, especially the C-AOG gradually has get the attention of researchers in the field of behavior pattern recognition [41–43]. Fire and Zhu proposed a causal grammar-based model to infer the behavior fluent in a office scene [44]. The C-AOG based method treats the interactions between each agent as atomic actions, and then uses the And-or points to project the actions into the corresponding behavior. With C-AOG, the flow of behavior within the given scenario can be accurately derived while its contextual causality can be uniquely determined. Xu et al. proposed a probabilistic graphical model to represent and reason about the visibility change of objects in terms of C-AOG [45]. By jointly modeling short-term occlusions and long-term occlusions, the C-AOG based model can accurately infer the visibility of observation objectives as well as their locations in the videos. Li et al. proposed a Hierarchical And-Or Graph (H-AOG) which could predict the visibility pattern of the vehicle [46]. The H-AOG achieves a promising prediction accuracy of the vehicle pattern, compared to some state-of-the-art models like LSTM [47] or Gated Recurrent Unit (GRU) [48]. Zhang et al. represents a semantic hierarchy on the top of conv-layers by C-AOG [49]. By the mapping relationship between the convolutional features and the object-part patterns through C-AOG, the invisible changes within the CNN can be established as interpretable associations.

Compared with the DL-based recognition model, the C-AOG based model has higher robustness and better reasoning ability, because the C-AOG based model can deduce the behavior pattern changing trends through the observable object state. However, C-AOG relies on abstract events consisting of high-dimensional features. Therefore, C-AOG generally does not enable end-to-end behavior classification. Compared with DL, C-AOG is less general and can only perform behavioral pattern recognition in simpler scenarios because the state of the underlying nodes of C-AOG is often a binary variable corresponding to some specific actions.

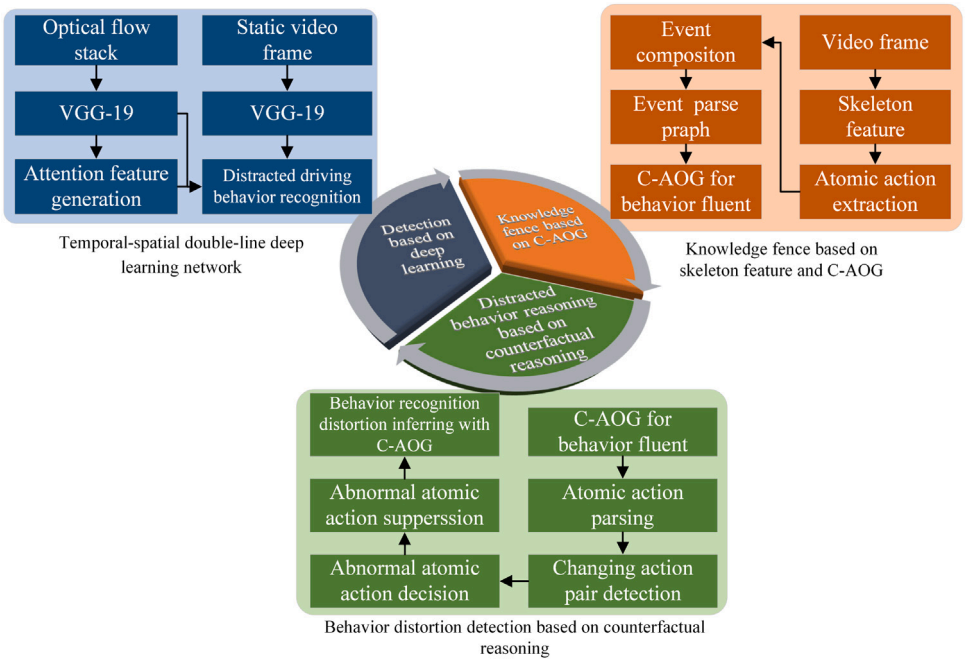


Fig. 1. Procedural causal inference framework for distracted driving recognition.

To address the shortcomings in the above studies, in this work, we use the DL network as the main body of distracted behavior recognition and construct the C-AOG as the monitor for the behavior recognition. By fusion the DL model and C-AOG, we aimed to build a universal and robust model to implement more accurate the distracted behavior recognition.

3. Method

In this section, we will give the detail design process of the proposed distracted driving recognition method. As shown in Fig. 1, the distracted driving behavior recognition method consists of three parts. First, we propose an attention-based DL recognition model, which forms the backbone of the behavior recognition method. Then, based on the C-AOG, we construct a knowledge fence to establish a derivable mechanism for changes in distracted driving behavior. Finally, we design a counterfactual inference mechanism based on the C-AOG knowledge fence. The inference mechanism aims to improve the accuracy and robustness of the deep learning network under incomplete observation conditions formed by noise and occlusion.

3.1. Detail framework of the TSD-DLN

3.1.1. Double-line structure for static feature extraction

TSD-DLN framework is adopted as the backbone network to detect the potential distracted behavior, as shown in Fig. 2. Temporal line utilizes inter-frame optical flow information to capture the historical dynamic feature, and the spatial line is designed to obtain the deep features of the current frame. Inspired by the previous study of the human gaze transition prediction [50], a GRU-based module is added into the framework to generate attention. At last the current RGB frame with attention information will be input into the pruned ResNet-101 [51] to predict the category of the distracted driving behavior.

3.1.2. Temporal feature extraction

The temporal line stacks the optical flow of the past few frames and then the stacked feature will be input into VGG-19 with five convolutional layers [52]. The stacked optical flow O_t is a dense fusion of consecutive frames from the past N frames to current frame. The dense features are represented as a set of pixel-level displacement vector d_{t-i}, as shown in Fig. 3.

The displacement vector of single point (u, v) at time t is denoted as $d_t(u, v)$. $d_t(u, v)$ contains horizontal and vertical position offset $d_{t,x}$ and $d_{t,y}$. Assuming the size of each video frame is W × H (width × height), the size of the stacked flow can be represented as (W, H, 2N), 2N means the horizontal and vertical position displacement are considered separately. Then the mathematical expression of stacked optical flow O_t can be constructed as follows;

$$\begin{aligned} O_t(u, v, 2n-1) &= d^x_{t-n+1}(u, v) \\ O_t(u, v, 2n) &= d^y_{t-n+1}(u, v) \\ u = [1; W], v &= [1; H], n = [1; N] \end{aligned} \tag{1}$$

Optical flow is obtained based on a theory for warping [53] and the stacked volume N is five, examples of the optical flow information for the consecutive video frames is shown in Fig. 4. The optical flow both in the horizontal and vertical direction are linearly rescaled into [0,255] to reduce the data dimension. The stacked optical flow O_t will be put into a VGG-19 based convolutional network that omits the final max-pooling layer for further high-dimensional feature extraction. We denote the output of the 5-layer VGG-19 as F_t^T which is a 14*14*512 stacked feature. As F_t^T contains the saliency transition tendency of the past five frame time, in the later attention generation module, F_t^T will serve as a dynamic index to adjust the effect of the GRU on the predicted attention.

3.1.3. Spatial feature extraction

In addition to temporal information, spatial feature of the current frame is another key factor for the attention area prediction and behavior recognition. Compared with the temporal feature, the spatial feature describes the detail objects distribution which is a crucial clue to decide the drivers' detail action. Like temporal feature, the spatial feature F_t^S is also extracted by a VGG-19 network. After the spatial feature extraction, the static feature of single video frame is encoded as a 14 × 14 × 512 stacked feature. In the latter attention module, F_t^S will serve as an attention decision factor for the behavior recognition process of the next frame.

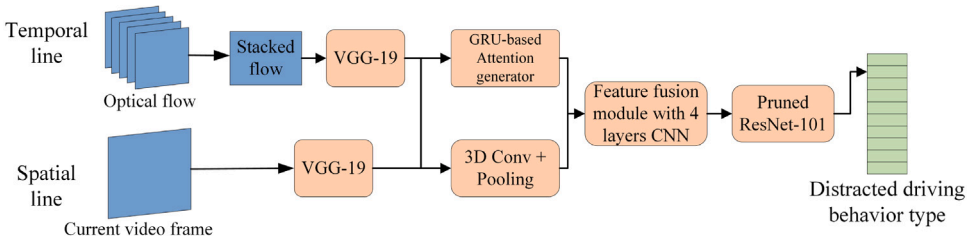


Fig. 2. Overview of the TSD-DLN.

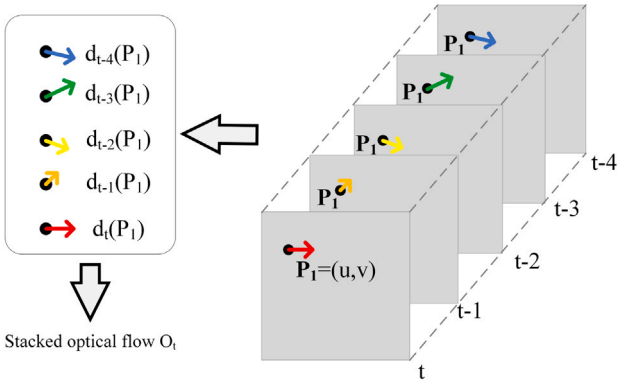
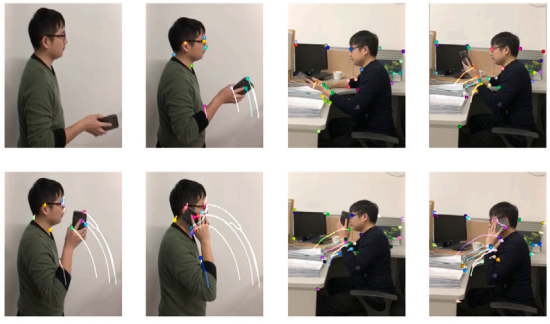
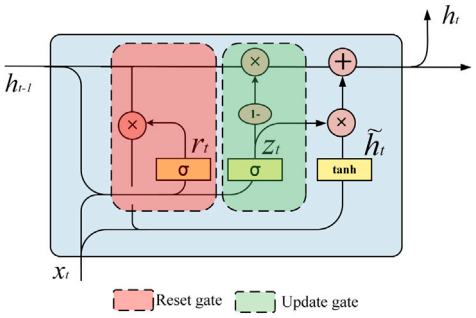


Fig. 3. Dense compression process to obtain stacked optical flow.

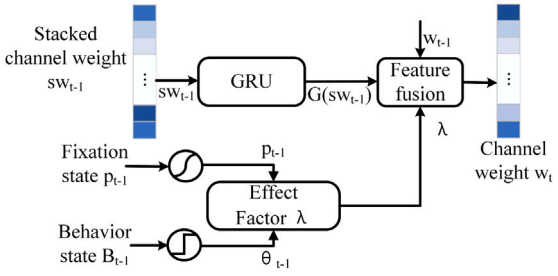


(a) Optical flow on a solid color background (b) Optical flow on a complex background

Fig. 4. Stacked optical flow for the consecutive video frames.



(a) Detail inner structure of GRU.



(b) Prediction process for the potential attention region.

Fig. 5. Attention generation module.

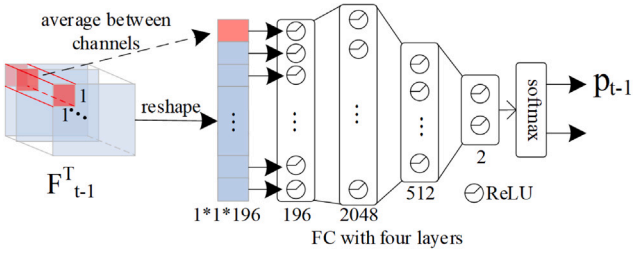


Fig. 6. Attention fixation probability prediction network.

3.1.4. GRU based attention generation module

We designed a top to bottom attention generation module, which predicts the attention distribution on the current behavior image based on the past consequent video frames' feature.

(A) Overview of the attention generation module

The attention generation module assigned different weights to the features of the image according to observation task and historical sequence data. In the proposed method, GRU is adopted to generate the attention feature, the detail structure of GRU is shown in Fig. 5(a). The reset gate is used to reduce the influence of past information on the current judgment, and the update gate decides which information should be discarded or what new information need to be added. Like LSTM, GRU is a kind of recurrent neural network that making predictions based on time series data. Compared with LSTM, GRU has the similar data processing performance and fewer tensor operations. Besides, the training speed of GRU is faster than that of the LSTM.

As shown in Fig. 5(b), given the stacked attention region feature sw_{t-1} of the past five frames, GRU predicts the potential attention region of the current frame by giving $G(sw_{t-1})$. The detail encoding process of the sw_{t-1} will be given in the part B of this section.

Besides the attention area distribution of the historical frame, we also considered the attention fixation effect and behavior transition boundary effect. The attention fixation effect means the attention area will remain stable when relatively few feature changes between frames [50]. The attention fixation effect is expressed as a probability value p_{t-1}. p_{t-1} represents possibility of attention area transition which is measured by the temporal feature. The inferring process of p_{t-1} is based on a fully connected neural network (FCNN), as shown in Fig. 6. p_{t-1} is a probabilistic score of fixation state, range from 0 to 1.

Behavior transition boundary effect means; when the behavior just changed, the potential attention area tends to transfer drastically in order to find new features. Unlike the attention fixation effect are soft and with long-term dependence, the behavioral transition boundary

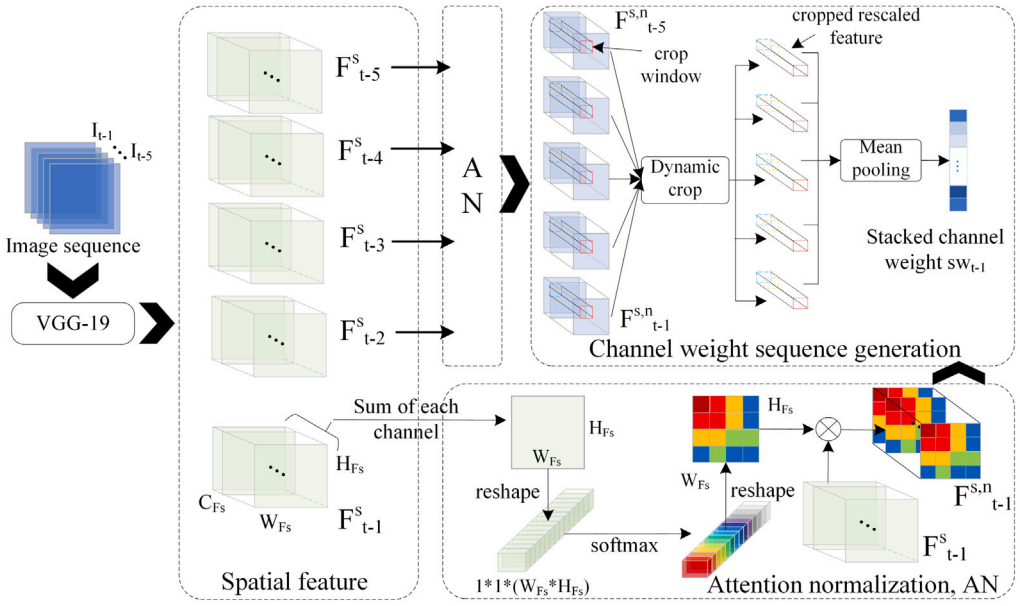


Fig. 7. Encoding process of stacked channel weight.

effect is relatively hard and transient. So, when the behavior transition happens, the effect of GRU needs to be enhanced immediately. The mathematical expression of the behavior transition boundary effect is shown in Eq. (2).

$$\text{Let } i \in [2,10], \text{ then } \theta_{t-1} = \begin{cases} \frac{\sigma}{\min(I)-1}, I = \{i \mid B_{t-1} \neq B_{t-i}\} \\ 0, (\forall i, B_{t-1} = B_{t-i}) \end{cases} \tag{2}$$

In Eq. (2), θ_{t-1} is the quantitative representation of the behavior transition boundary effect, σ is the maximum intensity of the effect, here we set σ to be 0.2. $min(I)-1$ is the transition decay factor. Combining the attention fixation and behavior transition effect, the predict attention region feature w_t for the current frame is expressed as the following equation.

$$\begin{aligned} w_t &= (1-\lambda) \times \mathrm{G}\left(\mathrm{sw}_{t-1}\right) + \max(0,\lambda) \times \mathrm{w}_{t-1} \\ \lambda &= \mathrm{p}_{t-1} - \theta_{t-1} \end{aligned} \tag{3}$$

(B) Stacked channel weight construction

GRU module predicts the potential attention distribution $G(sw_{t-1})$ of current frame based on the stacked channel weight sw_{t-1}. In this part, we will give the encoding process of sw_{t-1}. sw_{t-1} is constructed with the time sequence data, the detail encoding process is shown in Fig. 7. Each past image sequence contains five frames from $t-1$ to $t-5$, which are denoted as I_{t-1} to I_{t-5}. As mentioned in the spatial feature extraction part, the images will be imported into the VGG-19 to get the spatial feature, the image spatial feature for each time t is denoted as F_t^S.

Each spatial feature F_{t-i}^S will be normalized by the attention normalization module (AN) as shown in Fig. 7. The AN adopt a parameter free structure to improve the balance of the spatial feature [54]. Firstly, in AN, each channel of the spatial feature F_{t-i}^S will be summarized and then reshaped into a $1 \times 1 \times (W_{Fs} \times H_{Fs})$ flatten feature. Then the softmax is applied on the flatten feature to improve the feature's diversity. The softmax module compares the features from different spatial location and assigns each part with unique weight based on the global saliency. Original feature F_{t-i}^S is then rescaled by multiplying with the reshaped output of softmax, the rescaled feature is denoted as $F_{t-i}^{S,n}$. The AN module assigns the spatial features with weight by global softmax that could improve the robust of the later GRU module, because the input features are more equally distributed. Besides, the significant difference between features is further reduced, so that some relatively insignificant features can be captured by the later GRU network. In other words, the introduction of AN is aimed to further improve the receptive field of the encoding process.

At last, the attention dynamic crop and the mean-pooling method are designed to combine each $F_{t-i}^{S,n}$ into sw_{t-1}. The dynamic crop module crops part of the rescaled feature from AN's output and then imports all the cropped feature stacks into the mean-pooling module. As mentioned in the previous study [55], the top convolution layers are the projection of different high-level semantics, so by giving the top spatial feature with different weight, we could represent which part is more important for the behavior recognition. As shown in Fig. 7, crop window decides the coverage scope of the cropped rescaled feature. The crop window in fact simulate the human's gaze location when judging certain behavior, so the crop window's center is the attention area center C_{t-1}. And the size of crop window will be modified upon the average feature transition variance factor δ, which represent the fluctuation characteristics of the past five stacked channel weights. The mathematical expression of δ is shown in Eq. (4).

$$\delta = \frac{1}{L} \sum_{i=t-1}^{t-L} \left(\frac{\mathrm{sum}\left(sw_i\right)}{n} - E(sw) \right)^2 \tag{4}$$

L is the length of the sequence images which equals to five. sw_i is the predicted channel weight of each sequence image, n is the dimension of the stacked channel weight sw_i. $E(sw)$ is the mean value of past give five channel weights. δ represents how the crop window is resized. If the last L channel weights keep stable, the attention area would be similar between each time interval. By thresholding the δ, we adopt a fuzzy choice on the size of crop window. The crop window size is represented by its width W_{crop} and height H_{crop}, the detail fuzzy choosing of crop window size is as follows:

$$\mathrm{W}_{\mathrm{crop}} = H_{\mathrm{crop}} = \begin{cases} 3, 0 \leq \delta < 0.33 \\ 6, 0.33 \leq \delta < 0.67 \\ 9, 0.67 \leq \delta \leq 1 \end{cases} \tag{5}$$

According to the δ, the crop windows will be assigned with different size dynamically, so as to make the receptive field corresponding with attention fluctuation. By using the dynamic crop window, the cropped feature may have different size, so a mean pooling module is constructed to reshape the rescaled feature into a stacked channel weight. Mean pooling module averages the five rescaled features of the same channel and then reshape the average results into a flatten 512-dimensional channel weight sw_{t-1}.

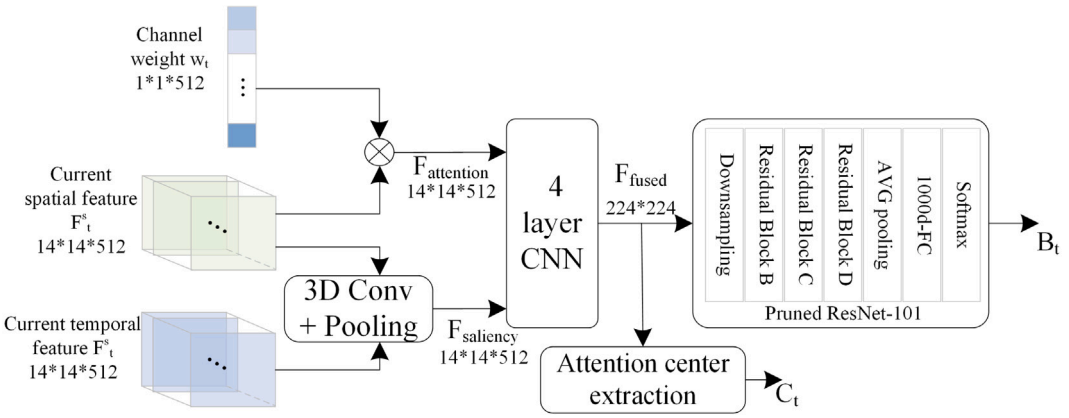


Fig. 8. Distracted driving behavior recognition network based on fused features.

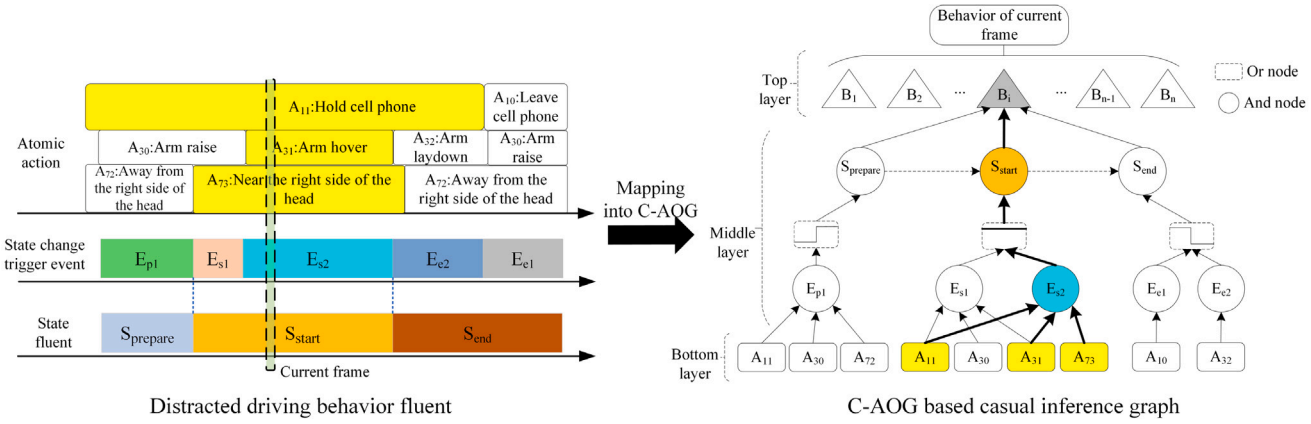


Fig. 9. The C-AOG structure for describing the changing process of distracted driving behavior fluent.

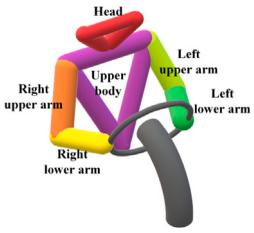


Fig. 10. The considered skeleton characteristics of the driver's upper body.

3.1.5. Distracted driving behavior detection

A pruned ResNet-101 framework is used to judge the distracted driving behavior pattern from the RGB image fused with attention information. The detail framework of the behavior recognition process is shown in Fig. 8.

$F_{attention}$ is the spatial feature of current frame fused with the channel attention weight w_t. $F_{saliency}$ is the saliency feature of the current frame which is extracted from the fusion of spatial and temporal features through 3D Conv + Pooling method [56]. $F_{attention}$ and $F_{saliency}$ are then fused by a 4 layers CNN to generate the behavior feature F_{fused}. The behavior feature F_{fused} is then input into the ResNet-101 which omit the conv1 and conv2_x layer. The output of pruned ResNet-101 is the probability of each distracted driving behavior. B_t is the behavior type for the current video frame that has the maximum probability. C_t is the attention area center for the current video which will be used for predicting the next attention area. C_t can be obtained by finding the weight distribution center of the F_{fused}.

3.2. Skeleton feature-based knowledge fence

Once the image pixel features are rapidly sparse or the key features are affected by noise, the accuracy of the DL-based model will drop sharply. For this reason, we propose a knowledge fence based on skeleton features to suppresses the behavior misrecognition by continuously judging the driver's behavior causality. The knowledge fence is embodied in a causality and-or graph which decomposes behavioral patterns into events composed of skeleton features. In this section, the detail framework of the knowledge fence will be given.

3.2.1. Overview of the C-AOG framework

As shown in Fig. 9, C-AOG is represented as a multi-layer structure [41]. The bottom layer is composed of the events formed by atomic action and their relation. The middle layer is composed of And-nodes and Or-nodes, which associates the bottom-level events with the behavior patterns and state flow defined in top layer. The top layer is a collection of various behavior patterns, which are often in a fluent change mode. The activation source of pattern fluent comes from the bottom layer, and the way of activation is decided by the middle layer.

3.2.2. Representation of atomic action

(A) Atomic action definition

In C-AOG, an atomic action is the smallest unit that constitutes an event and ultimately causes a fluent change in the video stream. As shown in Fig. 10, the atomic actions in the C-AOG describe the characteristics of the six body agents and their interrelationships.

The atomic action is composed of the agent state and the relationship (relative position) between each other. Agent state is used to describe the agent's own attributes, for example, right lower arm drops

Table 1
The atomic actions in the C-AOG.

Agent state			
Action node	Semantic content	Action node	Semantic content
A11	Right hand holds cell phone	A10	Right hand leaves the cell phone
A13	Right hand holds bottle or cup	A12	Right hand leaves bottle or cup
A21	Left hand holds cell phone	A20	Left hand leaves the cell phone
A23	Left hand holds bottle or cup	A22	Left hand leave bottle or cup
A14	Right hand free	A24	Left hand free
A30	Right lower arm raises	A40	Left lower arm raises
A31	Right lower arm hovers	A41	Left lower arm hovers
A32	Right lower arm drops	A42	Left lower arm drops
A50	Turn the head to the right	A51	Turn the head to the left
A60	Turn the body to the right	A61	Turn the body to the left
Relationship between agents			
Action node	Semantic content	Action node	Semantic content
A71	Near the top of the head	A70	Away from the top of the head
A73	Near the right side of the head	A72	Away from the right side of the head
A75	Near the left side of the head	A74	Away from the left side of the head
A77	Near the bottom of the head	A76	Away from the bottom of the head
A79	Near the steering wheel	A78	Away from the steering wheel
A81	Near the top of the head	A80	Away from the top of the head
A83	Near the right side of the head	A82	Away from the right side of the head
A85	Near the left side of the head	A84	Away from the left side of the head
A87	Near the bottom of the head	A86	Away from the bottom of the head
A89	Near the steering wheel	A88	Away from the steering wheel

Fig. 11. Upper body skeleton feature of the driver from the side view.

or right lower arm hovers. Besides, mutual relations between agents can also construct certain atomic actions, for example, near the right side of the head or away from the right side of the head. All atomic actions contained in the C-AOG are listed in Table 1. The first digital of action node represents the action group which also means each node in the same group is mutual exclusive. For example, when the driver is executing A11, he or she cannot simultaneously implement A10.

In order to effectively obtain the atomic action, we firstly transform the image features into the driver's skeleton features. From the skeleton feature, we can obtain the head and upper body's orientation state. Then by using the neural network, the skeleton feature is mapped into a 2D plane which is constructed by the gaussian mixture distribution. At last, by comparing the difference of gaussian mixture distribution between frames, the positional relationship among multi-agents can be solved.

(B) Orientation state recognition of the head and upper body

The skeleton feature is extracted from the RGB frame by Openpose [57], which detects the joint points of the body parts and then connect these points to get the human skeleton feature. As shown in Fig. 11, Openpose could give the clear upper body skeleton feature of the driver from the side view. The orientation state of the head and upper body is inferred by the skeleton feature. The driver's facial orientation can be distinguished by the angle between the connected line of the nose and the eyes. As shown in Fig. 12, when the driver looks ahead, we denote the angle as θ_{fn} which could be learned from the training data. The angle between two connected line of current

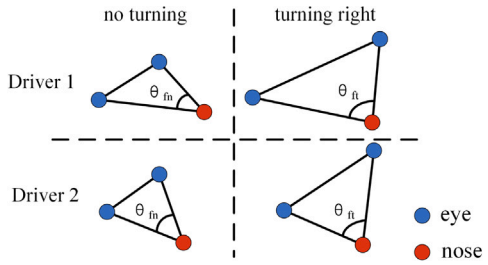


Fig. 12. Head turning recognition based on facial key points.

frame is denote as θ_{ft}. When the driver's head turns to right, θ_{ft} is relatively larger than the θ_{fn}. When the head turning, the relative distance between the eyes becomes larger and the nose point moves downward, consequently θ_{ft} becomes larger.

Therefore, based on the causal relationship between θ_{ft} and head turning, the probability of head turning action is expressed as Eq. (6).

$$p\left(A_{5i} \mid I_t\right) = \frac{1}{1+e^{\left(-\gamma_f \times \left(\frac{\theta_{ft}}{\theta_{fn}}-1\right)\right)}} \tag{6}$$

In the same way, the body's turning probability is estimated by the turning angle. We denote the θ_{bn} representing the baseline angle when the driver's body is facing ahead. As shown in Fig. 13, θ_{bn} and

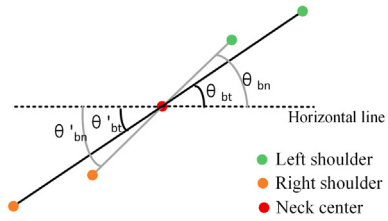
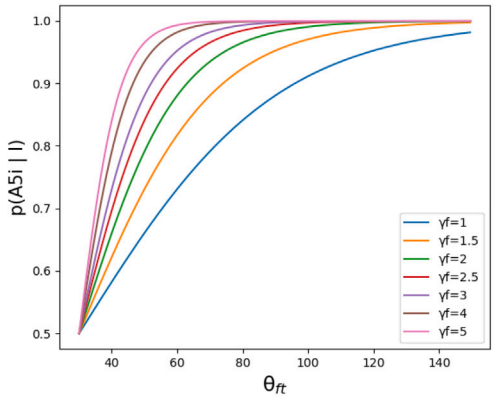
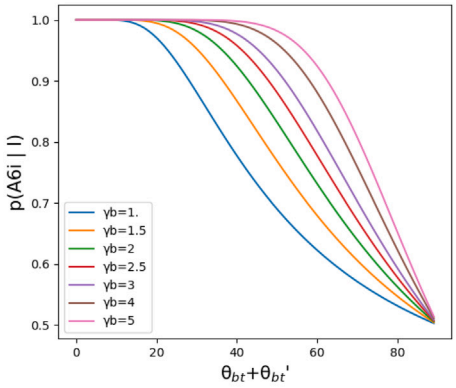


Fig. 13. Body turning recognition based on shoulders-neck points.



(a) Head turning probability distribution with different γ_f



(b) Body turning probability distribution with different γ_b

Fig. 14. Probability distribution of head and body turning with different effect factor.

θ_{bn}' are the angles between the shoulders-neck connected line and the horizontal line when the driver is driving normally. When video frame is obtained from the fixed camera, both θ_{bn} and θ_{bn}' are constants that can be calculated from the training data.

Let θ_{bn} and θ_{bn}' be the angle between shoulder's skeleton line and the horizontal line at current time t, then the probability of upper body rotation can be expressed as:

$$p\left(A_{6i} \mid I_t\right) = \frac{1}{1 + e^{\left(-\gamma_b \times \left(\frac{\theta_{bn}+\theta_{bn}'}{\theta_{bt}+\theta_{bt}'} - 1\right)\right)}} \tag{7}$$

In Eqs. (6) and (7), θ_{fn} and θ_{bn} are the expectations from the training data, which considering the physical differences between different drivers. γ_f and γ_b are the effect factor of the camera which reflect the sensitivity of the action detection. The probability distribution of head and body turning with different effect factor are shown in Fig. 14.

(C) Fusion mapping process of the head and upper body

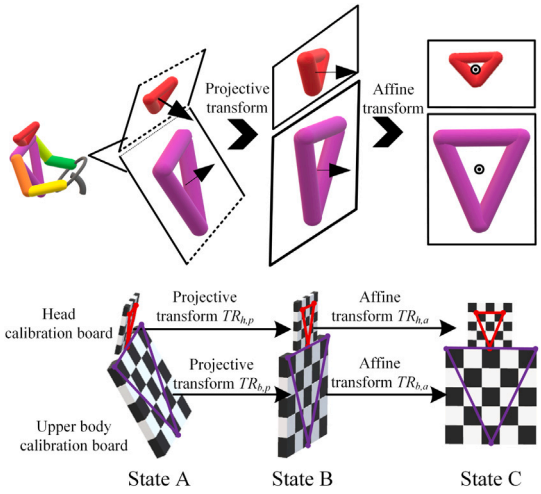


Fig. 15. Projection process of upper body's 3D features.

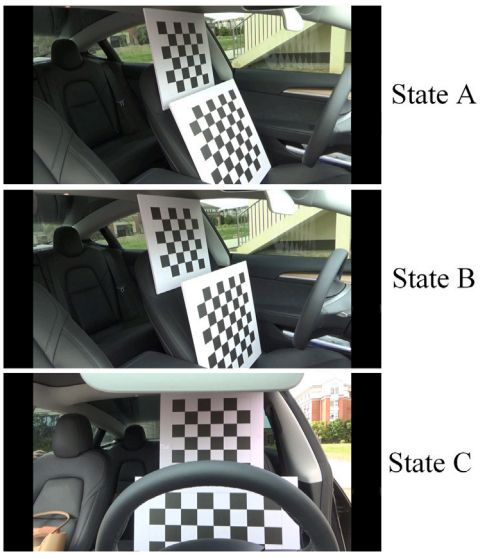


Fig. 16. Three calibration board states.

Since the skeleton features are represented by the key points on the RGB image, which essentially is a 3D feature, it is difficult to directly infer the relative relationship between different body agents. For this reason, we design a feature mapping method which combines the key points and their composed body plane into the same 2D plane, so as to obtain the relative positional relationship between different agents, then the corresponding atomic actions for the C-AOG can be naturally inferred. The features mapping method consists of two steps. The first step is the mapping and locating of the head and the upper body. As shown in Fig. 15, the affine and projective transformation are used to transform the key points of head and upper body into the same 2D plane. We used the calibration board to determine the head and upper body mapping matrix TR_h, TR_b. To clearly introduce the mapping process, we divide the calibration board into state A, B and C as shown in Fig. 16. Take the mapping process of the upper body agent for example, TR_b is expressed as follows:

$$TR_b = TR_{b,p} \times TR_{b,a} \tag{8}$$

$TR_{b,p}$ is the transform matrix from State A to B. Assume the point in state A is [u,v], and the corresponding point in state B is [x,y], then according to the projective transform rule we can get the following

expressions:

$$\begin{aligned}[x', y', w'] &= [u, v, w] \times TR_{b,p} \\ &= [u, v, w] \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \end{aligned} \tag{9}$$

$$x = \frac{x'}{w'}, y = \frac{y'}{w'}, a_{33} = 1 \tag{10}$$

$$\begin{cases} x = \frac{x'}{w'} = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}} \\ y = \frac{y'}{w'} = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}} \end{cases} \tag{11}$$

To obtain the $TR_{b,p}$, we choose four points from calibration board in state C and the corresponding four points in state B. Then we can construct eight equations composed of a_{ij} with the four chosen points. By solving the eight equations, we can get the each a_{ij}. Although the $TR_{b,p}$ is not a constant matrix which is correlated with the pitch angle θ_b between the driver's upper body and the horizontal line, we designed an adaptive way to keep the $TR_{b,p}$ available. From the angle between the backrest and the horizontal line at 45 degrees and ending at 135 degrees, $TR_{b,p}$ is calculated every 5 degrees then we could obtain 19 matrixes. With skeleton feature we can obtain the θ_b, thus by matching the angle interval that θ_b belongs to, a proper $TR_{b,p}$ can be obtained adaptively. For the $TR_{b,p}$, the pitch angle of the head hardly affects the mapping result, so when calculating the head turning probability there is no need to consider the influence of the pitch angle.

$TR_{b,a}$ is the affine transform matrix which change the plane from state B to state C. Denoting point in state C as $[u', v']$, and the corresponding point in state B as $[x, y]$, then according to the affine transform process, the $TR_{b,a}$ can be represented as follows:

$$[u', v', 1] = [x', y', 1] \times TR_{b,a} = [x', y', 1] \times \begin{bmatrix} a_1 & a_2 & 0 \\ b_1 & b_2 & 0 \\ c_1 & c_2 & 1 \end{bmatrix} \tag{12}$$

$$\begin{cases} u = a_1 x + b_1 y + c_1 \\ v = a_2 x + b_2 y + c_2 \end{cases} \tag{13}$$

Then we could choose at least 3 points in state B and the correspond points in state C. Given enough corresponding points, Eqs. (11) and (13) can be solved by least squares or SVD decomposition.

The raw head plane is composed of key points on the eyes and neck. The upper body plane is composed of the shoulders and the center of two hip joints. With the TR_h and TR_b, the key points of the head and upper body can be mapped into the same plane, thus the feature of the head and upper body are fused together.

(D) Arm dynamic features encoding

The second step of feature mapping is to project the arm features onto the same 2D plane. Compared with the rotation judgment and location of the head and upper body, the dynamic features of the arm are more complicated as they do not have a reference coordinate system. Therefore, for the 2D position and motion recognition of the driver's arm, we designed a gaussian mixture model (GMM) based skeleton feature dimensionality reduction method to mapping the skeleton feature of arm agents into a 2D plane. Firstly, the vector point composed of each arm agent's Part Affinity Fields [57] are sampled into Set S_{PAF} randomly. Generally, S_{PAF} obeys a normal distribution, so the GMM model is constructed to describe the coverage of each body agent as shown in Fig. 17(a). To further reduced the information dimension of the GMM, we center S_{PAF} to the origin point (0,0) as shown in Fig. 17(b).

Then covariance matrix of S_{PAF} are diagonalized to make the distribution of S_{PAF} on the x-axis and y-axis independent. The covariance matrix of S_{PAF} is expressed as follows:

$$\mathrm{cov}(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1} \tag{14}$$

In Eq. (14), (X_i, Y_i) is the location of each point in S_{PAF}. $\bar{X}$ and $\bar{Y}$ are the expectation of the sampled points. n is the size of S_{PAF}. $\mathrm{cov}(X, Y)$ is a real symmetric matrix and diagonalizable. As shown in Eqs. (15) and (16), Λ is the matrix after the diagonalization and $\boldsymbol{P}$ is transition matrix. S is the point set of the S_{PAF} in Fig. 17.(b)

$$\mathrm{cov}(X, Y) = \mathbf{P}\Lambda\mathbf{P}^{\mathbf{T}}, \mathbf{P}\mathbf{P}^{\mathbf{T}} = \boldsymbol{I}, \Lambda = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \tag{15}$$

$$\begin{aligned} \boldsymbol{\Lambda} &= \frac{1}{n-1} \mathbf{P}^{\mathrm{T}}(S - E(S))(S - E(S))^{\mathrm{T}} \mathbf{P} \\ &= \frac{1}{n-1} \left[\mathbf{P}^{\mathrm{T}}(S - E(S))\right] \left[\mathbf{P}^{\mathrm{T}}(S - E(S))\right]^{\mathbf{T}} \end{aligned} \tag{16}$$

The final normalized result S' is represented in the Eq. (17) and the normalization result is shown in Fig. 17(c).

$$S' = \mathrm{P}^{\mathrm{T}}(S - E(S)) \tag{17}$$

By normalization process, the arm agent's distribution feature on the x-axis and y-axis can be easily extracted. Here we chose seven parameters to indicate the posture features of the arm agent: diagonalized covariance matrix (α, β), expectations of PAF sampling points $(\mathrm{E(x)}, \mathrm{E(y)})$, coverage of arm agent $(\mathrm{L}_x, \mathrm{L}_y)$, and tilt angle of agent θ_a.

Then the features are input into the neural network to predict the arm agent location and feature distribution on a 2D plane, as shown in Fig. 18. The FCNN has three hidden layers and each layer contains 100 ReLU units. The output of the FCNN are the parameter of the arm agent's gaussian distribution on the new 2D plane. We give the mapping result of some distracted behavior in Fig. 19.

After mapping each body agent into the same 2D plane, the relative position relationship between arm and other agent can be inferred. Specifically, taking the right arm as an example, suppose gaussian distribution of the arm agent on the 2D fusion plane as P_t, and the position distribution corresponding to arm atomic action A_{7i} in Table 1 is Q_i. Then calculate the KL divergence between P_t and each Q_i, and make the state A_{7i} which minimizes $\mathrm{KL}(P_t \parallel Q_i)$ as the current relative positional relationship of the right arm. After determining the current arm location state A_{7i}, we represent the probability of A_{7i} as $1 - \mathrm{KL}(P_t \parallel Q_i)$.

$$\arg\min_i \mathrm{KL}(P_t \parallel Q_i) = \sum P_t(x, y) \log \frac{P_t(x, y)}{Q_i(x, y)} \tag{18}$$

The inferring process of the arm motion state is by comparing the center points of current forearm gaussian distribution $P_{a,t}$ and forearm gaussian distribution $P_{a,t-1}$ of last time interval. A threshold is adopted to judge the displacement of the center point so as to infer corresponding arm behavior listed in Table 1.

(E) Probability representation of the holding action

Accurate recognition of handheld objects is important for the recognition of distracted driving behaviors such as answering phone calls. In this paper, we use YOLOv4 to obtain the specific locations of some objects that may cause distracted driving, such as phones, water bottle, etc [58]. If the Euclidean displacement between detected objects' position and the wrist joint is less than a certain threshold τ_h, then we can infer the hand is holding the corresponding object. The threshold τ_h can be learned from the training data. The probability of the holding action depends on the detection probability of the holding object.

3.3. C-AOG graph for distracted driving behavior recognition

3.3.1. Event composition and event parse graph

Just like a fence is made up of multiple planks tied together, atomic actions are combined into basic events, and then the events are associated with the flow states of driving behaviors to form a C-AOG-based knowledge fence. To embody the event category, a method similar with the stochastic context sensitive grammar (SCSG) [59] is adopted to combine the atomic action into a single event. As shown in Fig. 9, the event is composed of the atomic action within the And-or Graph (AoG). With SCSG, the AoG for the event category can be

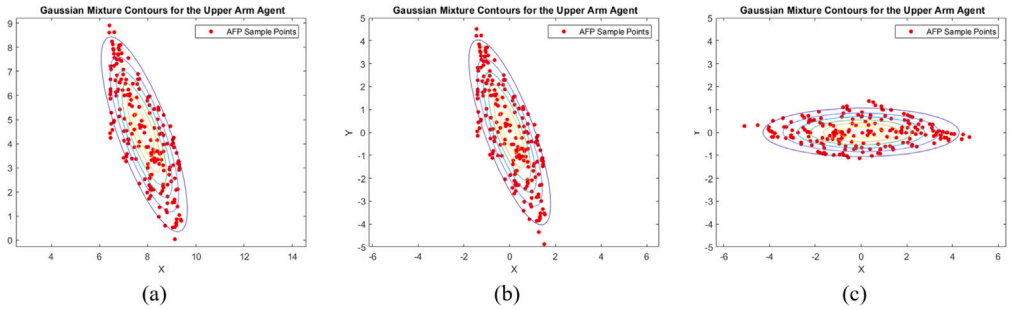


Fig. 17. Representation of upper limb features based on gaussian distribution.

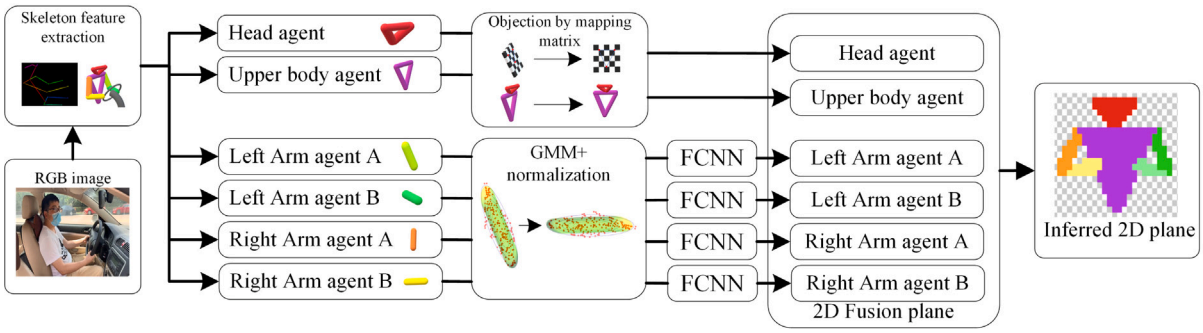


Fig. 18. Mapping process of upper body agent on 2D plane.

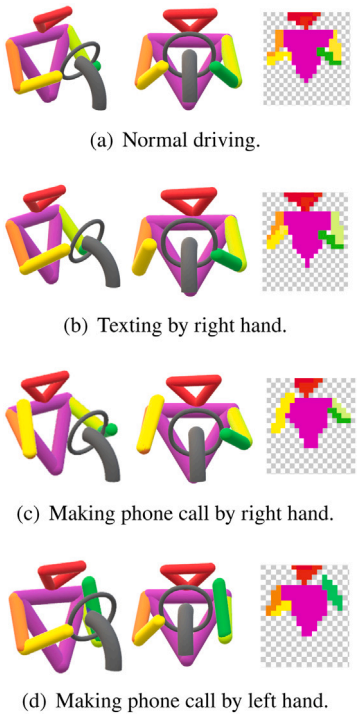


Fig. 19. Mapping result of four typical distracted behavior on 2D fusion plane.

described by 5-tuple parameter $\langle S_i, \mathrm{VE}, \mathrm{VA}, \Sigma, \mathrm{PA}\rangle$. S_i is the root node for the event groups which represent the specific behavior state. For each behavior Bi, the S_i has up to three state: prepare, start and end. VE represents the event nodes set and VA is the set of atomic actions. Σ is the possible composition form of the events. The selection of the Or-nodes is called a parse graph which is denoted as pg (the thicker paths in Fig. 9). Given a sample video I, an event sequence for certain behavior B_i is denoted as PG_i, which is shown in the following form:

$$PG_i = \left(pg_1, \ldots, pg_N\right) \tag{19}$$

If the parse graph PG_i matches the sample video I well, we could get the following posterior probability:

$$PG_i{}^* = \arg\max_{PG_i} p\left(PG_i \mid I\right) \tag{20}$$

With the optimal $PG_i{}^*$, the event of the sample video I could be inferred sequentially, naturally the state flow for each behavior can be predicted. With the training video, the event E_i could automatically composed by the detected atomic action and described by the SCGC.

3.3.2. The probability model of the C-AOG

As C-AOG obeys the Bayes's rule, the optimal behavior state fluent changing process F_B in the C-AOG can be achieved by maximizing a posterior (MAP). Concretely, the probability $P_{\mathrm{C-AOG}}$ is defined based on each parse graph pg_E which links the events and behavior state, so the probability model of the C-AOG conditioned on the pg_E. Besides, based on the results of a previous study on C-AOG [41], the probability model of the C-AOG can be represented as follows:

$$P_{\mathrm{C-AOG}}\left(pg_E\right) = p\left(pg_E \mid F_B\right) \propto \exp\left(-\mathcal{E}_{\mathrm{C-AOG}}\left(pg_E\right)\right) \tag{21}$$

where

$$\mathcal{E}_{\mathrm{C-AOG}}\left(pg_E\right) = \mathcal{E}_0\left(pg_E\right) + \sum_{E \in CR(pg_E)} \lambda_E(w(E)) \tag{22}$$

$\mathcal{E}_{\mathrm{C-AOG}}\left(pg_E\right)$ is a causality score which describes the matching degree between video stream and C-AOG. $\mathcal{E}_0\left(pg_E\right)$ is the energy from the original model which assumes the events has no link to the behavior state fluent change. $CR\left(pg_E\right)$ is all the or-node connecting the event and the state fluent change to form the causal relation. $w(E)$ is the event parse graphs representing how C-AOG choosing the Or-node. λ_E represents the switch probability on the Or-nodes which could lead to the behavior state changing. To learn the causal relation between the event and the behavior, we adopted an iteration learning method

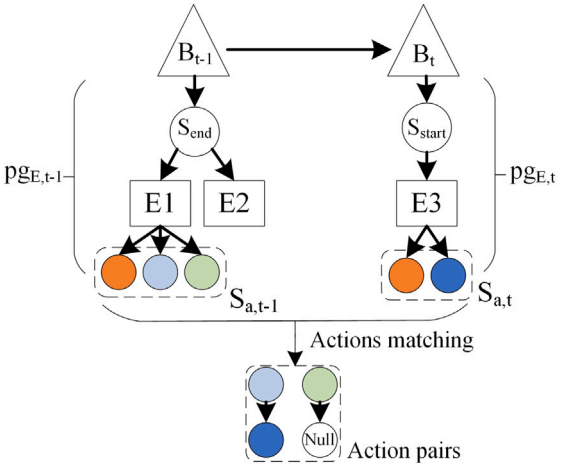


Fig. 20. Detection process for the atomic action changing pairs.

[41]. The iteration learning method contains two steps: First the best matched relation cr+ is choose by the Eq. (23).

$$\mathbf{cr}_{+} = \underset{\mathbf{cr}}{\operatorname{argmax}} \mathrm{KL}\left(p_{+} \parallel p\right) \tag{23}$$

p_{+} represents the new probability model after an event newly linked to a behavior fluent state within the C-AOG. In step two, we fit the selected causal relation with the real data by minimizing the KL-divergence between p_{+} and p. For each behavior B_i, the event selection will be end when the information gain of each candidate event is smaller than a threshold.

3.4. Counterfactual reasoning for enhancing the accuracy of the behavior detection

In Section 3.1, we obtained the real-time driver's distraction state through DL methods. However, DL-based detection method will be affected by the measurement noise which usually lead to the wrong distracted behavior detection. So, in this section, we will design a counterfactual reasoning mechanism by the C-AOG based causal reasoning knowledge fence to suppress the affection from the measurement noise and the agent occlusions.

Counterfactual reasoning infers the truthfulness of the distracted behavior changing by mining the logical consistency of the atomic actions. Logical consistency means the change of atomic action is a gradual process which conforms to kinematic limitation. In order to analysis the logical consistency, firstly, we need to extract the changed atomic action of each related agent. As shown in Fig. 20, assume in time t, the distracted behavior obtained from the TSD-DLN changed from B_{t-1} to B_t. The parse graphs for B_{t-1} and B_t are denoted as $pg_{E,t-1}$ and $pg_{E,t}$. The atomic action in $pg_{E,t-1}$ and $pg_{E,t}$ are denoted as $\mathrm{S}_{a,t-1}$ and $\mathrm{S}_{a,t}$. The same atomic action in $\mathrm{S}_{a,t}$ and $\mathrm{S}_{a,t-1}$ will be omitted, then remained actions in the two sets will be matched to find the action pairs belong to the same body agent. It needs to be noted that if an action cannot be paired with any other actions, then we will add a null node to form a pair. When the null node points to a certain action A_h, it means the occurrence of the action A_h, and if the action node A_d points to the null point, it means the action A_d terminated.

The action pairs can be obtained by checking the action encoding number in Table 1. For example, A_{1i} belongs to the right-hand state and A_{2i} belongs to the left-hand state. In fact, the action pairs reflect the state change of each agent which is the top to bottom mapping of the behavior changing. After the action pairs detection, the state changed agent can be fixed, then we will check whether the target agent obeys the kinematic limitation. The checking result is represented as the switch probability set of each action pair. When calculating the switch probability of the action pair, we will face three situations, agent action state change, agent action generation and agent action termination. For the agent action state change and agent action generation, the switch probability $\mathrm{P}_{sw,t}$ is expressed as follows:

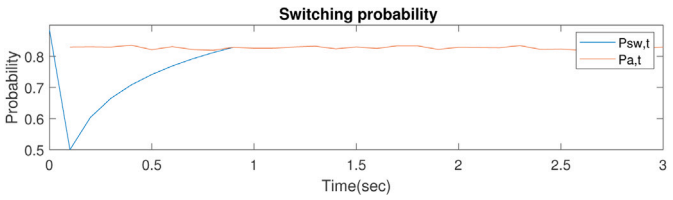


Fig. 21. Probabilistic suppression curve of abnormal atomic action. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\mathrm{P}_{sw,t} = \varphi_a \times \left(1 + \ln\left(\frac{t-\tau}{\delta}\right) \times \omega\right) \times \mathrm{P}_{a,t} \tag{24}$$

In Eq. (24), $\mathrm{P}_{a,t}$ is the probability of the newly detected or generated action. φ_a is the conversion factor which indicates the switching possibility among each action pair. ω is the time decay factor, which is used to modify the decay degree of the conversion factor over time. t represents the current time and τ indicates the timestamp when the behavior changed. δ represents the time interval. When action pair changing from null to new atomic action, the φ_a equals to 1 and ω equals to 0.01. When judging the switch probability $\mathrm{P}_{sw,t}$ of the state changed pair, φ_a and ω are taken as $\frac{0.5}{\mathrm{P}_{a,t}}$ and 3 respectively. As shown in Fig. 21, the red curve represents the probability of the detected new atomic behavior. When the old atomic action of a agent is transformed to a new atomic action, if the transformation is not logical or not directly connected in the temporal order, the probability of the new atomic behavior is replaced by the switch probability $\mathrm{P}_{sw,t}$. As shown in the blue curve in Fig. 21, the switch probability $\mathrm{P}_{sw,t}$ is first pulled down to 0.5 by the conversion index φ_a. And then since the new atomic action is continuously detected, the chance of misrecognition gradually decreases, so the $\mathrm{P}_{sw,t}$ will restore to the new atomic action probability after 5 to 10 time-steps under the effect of ω. Having no logical or temporal connection means the agent cannot switch between the two atomic action immediately or the two nodes in the action pair has no causal relationship. For example, sometimes the object detection method wrongly recognizes cell phone as bottle, in this case, the atomic action A_{11} in Table 1 may switch to action A_{13}. However, A_{11} and A_{13} usually have no directed link, so in this situation, we need to lower the confidence of new action with Eq. (24).

For the atomic action termination, there exists two situations: the atomic action termination by the causality reasoning or by the occlusion of the atomic action subject. The atomic action termination by the causality reasoning is usually with high confidence, for example, the end of the body turning can be accurately identified by the body turning angle. However, for some atomic actions termination such as the hand holding action and the relation between agents, the failure of the object detection and the occlusion between agents can easily lead to the immediate termination of these behaviors. Therefore, in order to eliminate the potential behavior distortion caused by the invalid atomic action termination, we designed a behavioral mutation damping mechanism (MDM). The MDM like a spring, link the atomic action and the null node to form a soft action termination. The MDM is embodied as a probability function shown in Eq. (25).

$$\mathrm{P}_{a,t} = \left(1 - \frac{1}{1 + e^{\omega \times \left(\frac{-(t-\tau)}{\delta} + 5\right)}}\right) \times P_{a,\tau} \tag{25}$$

The Eq. (25) is designed based on the sigmoid function. $\mathrm{P}_{a,t}$ is the probability of the ending action α. τ is the time stamp for the termination start and δ is the time interval between video frames. ω is

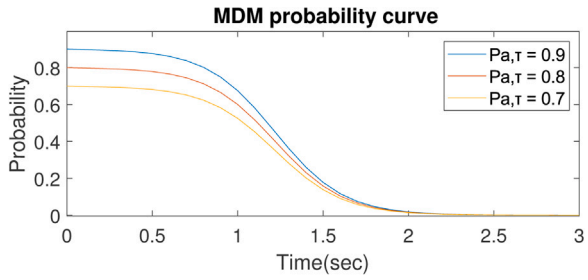


Fig. 22. Behavioral mutations damping probability curves.

the effect factor which determines the declining rate of the probability curve. As shown in Fig. 22, $P_{a,\tau=0}$ represents the probability value before the sudden termination of an atomic action. Then, under the action of MDM, the probability of action a will gradually decrease instead of falling to a minimal value or 0.

The MDM can suppress a steep drop of the atomic action probability if the atomic action is terminated without causality. Therefore, a certain recovery period can be reserved for the invalidly terminated atomic actions, thus avoiding the impact of short-time noise on the behavior fluent change in the C-AOG.

4. Experiment setup and validation dataset

4.1. Training process for the TSD-DLN

The training process is divided into several single stages which corresponding to the information flow of the recognition process. Both VGG-19 networks in the Fig. 2 are pretrained on the ImageNet [60] and COCO datasets [61] to get the initial weights. The FCNN module shown in Fig. 6 is trained on the GTEA Gaze+ [62] which is a specially collected open source dataset for the gaze transition prediction. Binary Cross-Entropy loss function is used to train the FCNN module.

The GRU module in Fig. 5 has three hidden layers and the size of each layer is 64. When training the GRU, we use mean squared error loss function to compare the predicted channel weight w_t and the real w'_t extracted from the labeled attention area of training images.

Unlike semantic segmentation has the fixed and accurate area distribution, the attention area corresponding to the key feature distribution. Therefore, as a person's gaze, attention area often quivers in a small range. The traditional loss function such as the Binary Cross-Entropy is difficult to effectively calculate the error between the quivered attention area and the ground truth [50]. So, when calculating the loss for the F_{fused}, we adopted a modified Binary Cross-Entropy loss function, as shown in Eq. (26). N is the number of the pixel point in F_{fused} and F_{fused}[i] is the ith pixel point in the attention saliency map. F'_{fused} is obtained by convolving an isotropic gaussian over the training image with labeled attention area.

$$L\left(F_{\text{fused}}, F'_{\text{fused}}\right) = -\frac{1}{N}\sum_{i=1}^{N}\left\{\left(1+w^{qs}_{i,\text{ norm}}\right)\cdot\left(F'_{\text{fused}}[i]\right.\right.$$
$$\left.\left.\cdot\log\left(F_{\text{fused}}[i]\right)+\left(1-F'_{\text{fused}}[i]\right)\cdot\log\left(1-F_{\text{fused}}[i]\right)\right)\right\} \tag{26}$$

In Eq. (26), a quiver influence suppression factor $w^{qs}_{i,norm}$ is added into the Binary Cross-Entropy loss function. $w^{qs}_{i,norm}$ is designed to rescale the loss weight of each pixels in F_{fused}. The basic principle of the $w^{qs}_{i,norm}$ is that loss weight of the pixels within small distance from the ground truth attention center C'_t will be enhanced and vice versa. Thus, assuming the distance from point i to center point C'_t as d_i, $w^{qs}_{i,norm}$ is denoted as a nonlinear function whose variable is d_i. We use the tanh function to construct the w^{qs}_i and then normalize w^{qs}_i to obtain $w^{qs}_{i,norm}$, the calculating process is shown in Eqs. (27) and (28).

$$w^{qs}_i = \left(\max(D)+\mathrm{d}_i\right)\times\left(1-\tanh\left(\frac{\mathrm{d}_i}{\max(D)}\right)\right), \mathrm{d}_i \in D \tag{27}$$

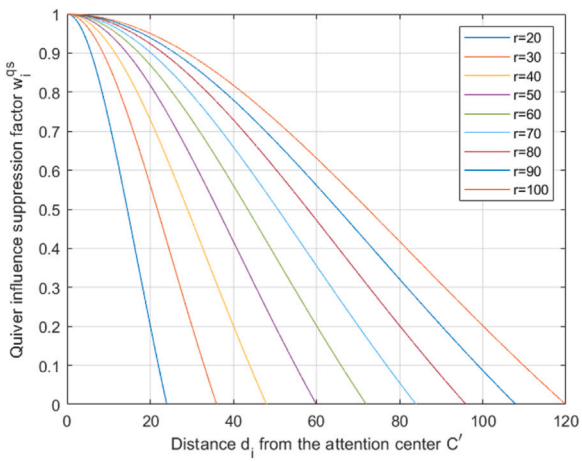


Fig. 23. Changing curve of the quiver influence suppression factor under different observed attention area.

$$w^{qs}_{i,norm} = \frac{w^{qs}_i - \max\left(w^{qs}\right)}{\max\left(W^{qs}\right)-\min\left(W^{qs}\right)}, w^{qs}_i \in W^{qs} \tag{28}$$

The function curve of the $w^{qs}_{i,norm}$ is shown in Fig. 23, r represents maximum coverage radius of the observed attention area. The attention area center C_t is obtained by finding the center of the $F_{attention}$. Since the learning process of the $F_{attention}$ is already included in the learning process of F_{fused}, there is no need to design an additional training process for the C_t.

ResNet-101 for the final behavior recognition in Fig. 8 are pretrained by the ImageNet and COCO. Then the conv1 and conv2_x layers are omitted to form the pruned network which could directly make the behavior feature with attention information F_{fused} as the input variable. As shown in Eq. (29), cross entropy loss function for multiple classifiers is adopted to evaluate the classification result when training the pruned ResNet-101 network.c represents the category of the distracted driving behavior, M_B is the number of distracted driving behavior category. N is the number of the observed samples, i means serial number of the observed sample. y_{ic} is a symbolic function when the behavior type of sample i is equal to c, the value of y_{ic} is 1, otherwise y_{ic} is 0. p_{ic} is the probability of observation sample i belonging to category c, p_{ic} is predicted by the last softmax layer of the pruned ResNet-101.

$$L = \frac{1}{N}\sum_i L_i = -\frac{1}{N}\sum_i\sum_{c=1}^{M_B} y_{ic}\log\left(p_{ic}\right) \tag{29}$$

In Table 2, we listed the training configuration of the different part in the attention-based deep learning network. All the detail implementation is based on the PyTorch which is an optimized tensors library for deep learning. The experiments were executed on a computing platform include one Intel Core i7-5930K CPU and four NVIDIA GeForce TITAN Xp GPUs.

4.2. Learning process for the C-AOG

As shown in Fig. 18, the arm agent feature is encoded by the gaussian mixture distribution, and seven characteristic parameters of the distribution are input into the five-layer FCNN. The FCNN for each arm agent are trained independently. The output of the FCNN is the arm agent's gaussian distribution on the fusion 2D plane. The loss function for each FCNN module is designed based on Kullback–Leibler divergence, as shown in Eq. (30), to evaluate difference between the predicted distribution $P(X,Y)$ of the agents and the distribution $P'(X,Y)$ of the ground truth data.

$$KL\left(P \mid P'\right) = \sum P(X,Y)\log\frac{P(X,Y)}{P_{\prime}(X,Y)} \tag{30}$$

Table 2
Training configuration of different modules in TSD-DLN.

DL module	VGG-19	NN	GRU	ResNet-101
Optimizer	Ensemble SGD [63]	SGD	ADM	Ensemble SGD
Learning rate or strategy	Cyclic cosine annealing [64]	0.001	0.001	Cyclic cosine annealing
Dropout rate	–	0.1	0.1	–
Batch size	64	32	32	64

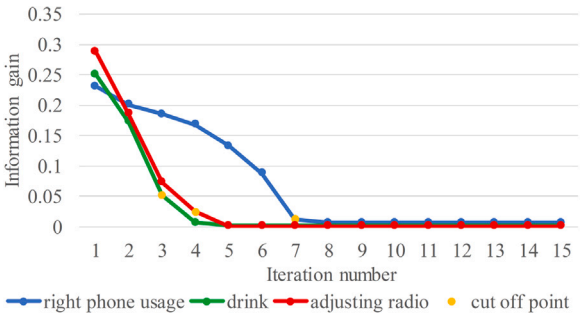


Fig. 24. The information gain of C-AOG during iterative learning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

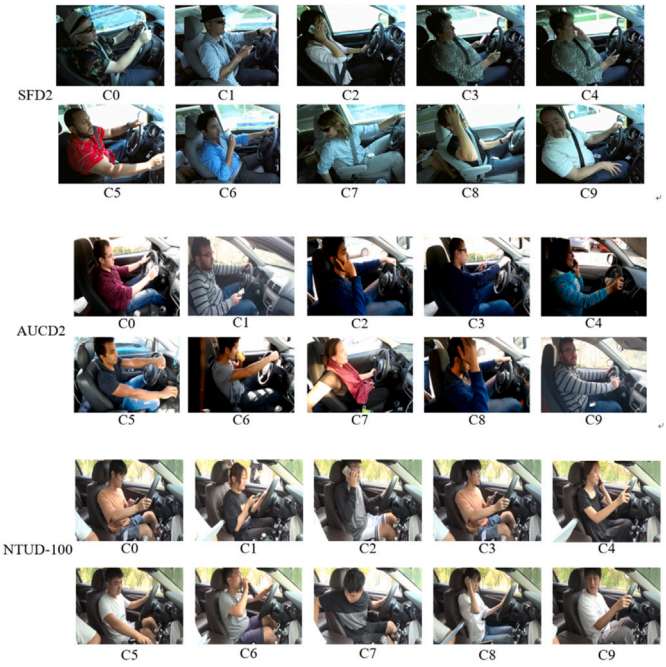


Fig. 25. Behavior samples in three benchmark datasets.

The learning process for the C-AOG includes two steps. Firstly, the event is composed by the detected atomic actions. As shown in Fig. 10, when the driver is calling, the detected atomic actions are A_{11}, A_{22} and A_{31}, if there is no same combination in the event database, then a new event will be created to represent the detected atomic actions combination. From the static images of the Kaggle distracted driving dataset [65] and based on the atomic action defined in Table 1. Only the events causing the behavior fluent change will be included in the C-AOG, all the events not included will be encoded as E_0. Secondly, we causally correlate the events with the behavioral fluent to construct the C-AOG by the information gain-based iterative method [41]. The iterative training data of C-AOG is obtained from the self-collected dataset NTUD2 which contain 100 videos that reflect drivers' distracted driving behavior.

In Fig. 24, we give the information gain curves when the C-AOG learns the causality of three behaviors separately. The yellow points on the curve are cut-off points which indicates the C-AOG structure representing specific behavioral changes has reached an optimal state. As can be seen in Fig. 24, the addition of relevant events in each iteration of C-AOG makes the information gain of the model gradually shrink to a smaller threshold. When the information gain is less than a threshold represents that there are no more events affecting the current behavior fluent, thus the C-AOG reaches the optimum.

4.3. Experimental datasets

4.3.1. Standard datasets and data labeling

We used the following three benchmark datasets to verify recognition accuracy of the proposed method, AUC Distracted Driver (AUCD2) dataset [33], State Farm Distracted Driver (SFD2) dataset [65] and a self-collected dataset named NTU Distracted Driver (NTUD2). The data samples in AUCD2, SFD2 and NTUD2 are collected from the camera with a fixed perspective. Part samples in AUCD2, SFD2 and NTUD2 are shown in Fig. 25, all the samples from AUCD2 and SFD2 datasets are divided into ten distracted behavior categories C_i, which are safe driving, right phone usage, left phone usage, text right, text left, adjusting radio, drinking, hair or makeup, reaching behind, and talking to passenger.

AUCD2 and SFD2 rarely include the switching process between behaviors and only contain images with prominent behaviors feature. Therefore, by additionally collecting the complete distraction behavior process and switching process between behaviors, we create a new dataset NTUD2 which recorded the distracted driving behavior of 100 students and faculty members. 100 participants were divided into 10 groups and each group implement the distracting behaviors in a different order to obtain transitions between the multiple behaviors.

In order to label the data with behavioral attributes, we designed a continuous annotation tool for the video data of NTUD2. As shown in Fig. 26, the data annotation tool is based on a labeling software for traffic scene risk classification that we previously developed [25,66]. While annotators watching the collected videos, they press specified keys on the keyboard to annotate each video frame with behavioral attributes. There were ten buttons available on the keyboard (from Q to P) corresponding to the ten distracted behavior categories C_i. If no key is pressed, the video will be paused waiting for the annotation.

The labeling process of the training data for C-AOG is different from the process described in Fig. 26. As shown in Fig. 27, the labeling tool for the C-AOG's training data focuses on the state annotation of the specific behavior. When labeling the data, the input video is the entire process of a certain behavior from beginning to end. After marking, the complete behavior process will be divided into three states, prepare, start, and end.

4.3.2. The image noise and distortion considered in the experiment

To verify the robustness of the proposed method in noisy and occluded environments, based on noise type in TID2013 [67], we chose three kinds of noise that existed in the actual image collection process, as shown in Table 3, and some samples that contain the considered noise are given in Fig. 28.

Table 3
Type of noise added manually in the testing dataset.

No	Type of distortion	Practical situation	Effect	Generation method
1	Frame blur	Poor frame compression or camera shake	Global feature sparseness	Random pixel switching in the entire image domain
2	Local block-wise distortions of different intensity	Poor transmission or environment noise	Agent occlusion	Randomly mosaic by opencv
3	Random RGB mask	Artificial noise	Agent occlusion	Random area set to black

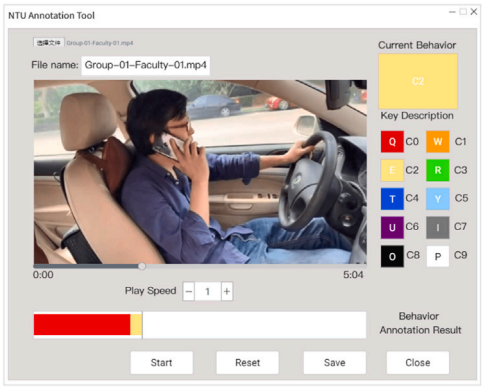


Fig. 26. Distracted behavior annotation tool for video data.

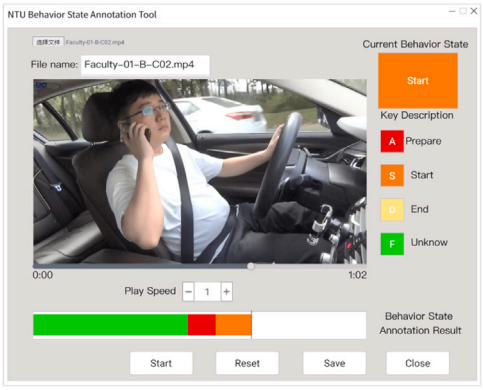


Fig. 27. Fluent state annotation tool for distracted driving behavior.

4.3.3. Dataset construction for training and testing

The training and testing data are extracted from the three datasets: AUCD2, SFD2 and NTUD2. The composition of the training data is shown in Table 4. 80% of the data in the training set is used for network training, and 20% is used for network validation.

As the proposed method depends on the stacked frames to obtain the temporal feature, so when training VGG-19 in the attention-based network, each image from AUCD2 and SFD2 is converted to a stack of ten identical pictures. The GRU module is trained only on data samples from NTUD2, for the AUCD2 and SFD2 lack sequent switching features between behaviors.

To evaluate the recognition and inferring performance of the proposed method, we designed three testing datasets. The first test dataset, denoting as static test dataset (STD), is composed of the static images from the aforementioned three datasets. STD contains 10,000 pictures, in detail, each behavior corresponds to 1000 pictures.

The second dataset, denoting as procedural test dataset (PTD), contains the complete process of the distracted driving behavior and the switching process between behaviors. As the distracted behavior is a combination of multiple procedural actions, PTD aims to verify whether each method can effectively identify the entire process of distracted behavior from beginning to end. In PTD, each data sample is a complete behavior video clip including 2 to 5 behaviors. By the labeling tools in Fig. 26, the label for each data sample is a data sequence corresponding to the behavior of each frames. Thus, we can get the exact start and end frame of the behaviors from the data samples in PTD. In PTD, there are 600 videos clips captured from the NTD2 dataset, reflecting the process feature of the distracted behavior.

The third dataset is denoted as procedural test dataset with frame distortion (PTD-D). Compared with PTD, part of the data in PTD-D is artificially added with specific noise. 120 data samples are chosen from the PTD, then each sample is injected with different noise. To maintain the balance of the testing data, the video clips having the noise categories in Table 4 are all 30. And the rest 30 video clips have the body agent occlusion situation.

4.4. Baselines

We used the following ten baselines for method performance comparison. The first baseline group contain five methods, VGG-19 [52], ResNet-101 [51], AlexNet [68], Inception V3 [69], and Densenet-201 [70], which are all very effective CNN-based deep learning framework. The second baseline group contains two CNN-RNN fusion deep learning framework, InceptionV3-BiLSTM [30] and InceptionV3-GRU [30]. The third baseline group contains D-HCNN [13], OWIPA [14], and CNN-GA [33] which fused different features to improve the classification accuracy.

4.5. Evaluation metric

We compared the recognition and inferring performance using Average Accuracy (AA), F1-Measure, and the area under the curve of receiver operating characteristics (AUC). AA evaluates the overall accuracy on the test dataset, regardless of data imbalance. AA's calculation formula is shown in Eq. (31), a_i is the correctly classified samples and m is the total number of samples.

$$A_{\text{avg}} = \frac{a_1 + a_2 + \cdots + a_m}{m} \tag{31}$$

F1-Measure is a comprehensive indicator that balances accuracy and recall. The calculation formula is as shown in Eq. (32).

$$F_\beta = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{32}$$

The ROC curve can reduce the interference caused by the imbalance of the testing dataset, thus ROC can objectively evaluate the performance of the classifier [71]. The abscissa of ROC is the False Positive Rate (FPR), and the ordinate is the True Positive Rate (TPR). AUC is the quantification of the ROC curve and can effectively evaluate the classifier's performance [72].

5. Experimental results and discussion

To verify the effectiveness of the attention module, we upsampled the attention feature $F_{attention}$ to a heat map and then fused attention feature with the original frame. From the fused results shown in Fig. 29, we can clearly see that most attention fields are distributed around the key features for the behavior type judgement.

We compared the recognition performance of multiple methods on STD, PTD, and PTD-D datasets. The recognition performance of baseline

Fig. 28. Example of noiseless and noisy driving behavior images.

Table 4
The composition of the training data.

Behavior type	No	Training data size			Behavior type	No	Training data size		
		AUCD2	SFD2	NTUD2			AUCD2	SFD2	NTUD2
Safe driving	C0	2489	700	2500	Adjusting radio	C5	2312	700	2500
Text right	C1	2267	700	2500	Drinking	C6	2325	700	2500
Left phone usage	C2	2317	700	2500	Hair or makeup	C7	2002	700	2500
Text left	C3	2346	700	2500	Reaching behind	C8	1911	700	2500
Left phone usage	C4	2326	700	2500	Talking to passenger	C9	2129	700	2500

Fig. 29. Attention features on the video frames of different distracted driving behaviors.

Table 5
The recognition results of multiple methods on the STD dataset.

Recognition model		VGG19	ResNetr-101	AlexNet	Inception V3	Densenet-201	InceptionV3-BiLSTM	InceptionV3-GRU	CNN-GA	D-HCNN	OWIPA	Ours without C-AOG	Ours with C-AOG
AA		0.812	0.852	0.751	0.860	0.836	0.872	0.867	0.861	0.830	**0.883**	0.846	0.848
F1		0.779	0.819	0.714	0.826	0.795	0.837	0.828	0.822	0.794	**0.843**	0.811	0.814
AUC	C0	0.837	0.866	0.749	0.861	0.830	**0.910**	0.901	0.872	0.842	0.899	0.843	0.839
	C1	0.829	**0.891**	0.801	0.864	0.832	0.875	0.867	0.885	0.802	0.879	0.869	0.866
	C2	0.816	0.883	0.756	0.882	0.843	0.885	0.890	0.871	0.831	**0.897**	0.870	0.873
	C3	0.820	0.852	0.779	0.850	0.821	0.856	0.860	0.862	0.851	**0.910**	0.862	0.867
	C4	0.807	0.870	0.730	0.890	0.822	0.871	0.853	0.871	0.812	**0.893**	0.874	0.882
	C5	0.810	0.792	0.710	**0.886**	0.845	0.823	0.847	0.851	0.799	0.855	0.802	0.799
	C6	0.834	0.878	0.801	0.872	0.872	**0.920**	0.902	0.864	0.825	0.916	0.846	0.827
	C7	0.782	0.811	0.690	0.833	0.827	0.834	0.827	0.812	0.835	**0.861**	0.801	0.814
	C8	0.788	0.835	0.758	0.840	0.825	**0.890**	0.876	0.862	0.855	0.832	0.855	0.858
	C9	0.799	0.842	0.740	0.831	0.819	0.855	0.842	**0.856**	0.845	0.855	0.843	0.850

methods and ours on the three datasets is evaluated by the classification accuracy and F1 score, as shown in the Tables 5 to 7.

As shown in Table 5, the behavior recognition performance of the CNN-based DL network is worse than that of the CNN-RNN fusion DL network on the STD dataset. Recognition methods that incorporate multiple types of body features as classification features show better performance than methods that use original single-frame images as input. In addition, we can find from the experimental results that simple feature dimensionality reduction, such as the extraction of HOG features [13], does not lead to significant performance improvement of the classification model. When handling the STD, the lack of optical flow information and the invalidation of C-AOG make the performance of the proposed method similar to the ResNet.

Compared with the classification performance on the STD, proposed method obtained relatively better classification performance on the PTD. As shown in Table 6, compared with other baseline methods,

Table 6
The recognition results of multiple methods on the PTD dataset.

Recognition model		VGG19	ResNet-101	AlexNet	Inception V3	Densenet-201	InceptionV3-BiLSTM	InceptionV3-GRU	CNN-GA	D-HCNN	OWIPA	Ours without C-AOG	Ours with C-AOG
AA		0.794	0.833	0.730	0.847	0.821	0.853	0.852	0.865	0.815	0.881	0.881	**0.895**
F1		0.762	0.795	0.691	0.812	0.785	0.823	0.813	0.826	0.778	0.845	0.844	**0.863**
AUC	C0	0.821	0.852	0.721	0.852	0.815	**0.899**	0.872	0.869	0.831	0.897	0.895	0.892
	C1	0.814	0.865	0.766	0.846	0.822	0.866	0.836	**0.889**	0.782	0.808	0.874	0.872
	C2	0.807	0.867	0.755	0.868	0.836	0.875	0.886	0.876	0.812	**0.893**	0.889	0.882
	C3	0.816	0.841	0.779	0.847	0.809	0.849	0.847	0.858	0.837	0.893	0.904	**0.912**
	C4	0.790	0.852	0.712	0.867	0.817	0.865	0.851	0.868	0.801	0.883	0.874	**0.919**
	C5	0.801	0.795	0.682	0.876	0.838	0.801	0.844	0.868	0.781	0.857	0.863	**0.885**
	C6	0.808	0.862	0.782	0.847	0.851	0.892	0.890	0.858	0.803	0.902	0.914	**0.939**
	C7	0.763	0.783	0.652	0.821	0.809	0.803	0.812	0.815	0.831	0.864	0.872	**0.893**
	C8	0.753	0.802	0.728	0.826	0.812	0.857	0.858	**0.869**	0.851	0.832	0.843	0.864
	C9	0.766	0.813	0.721	0.817	0.802	0.821	0.832	0.877	0.822	0.857	0.885	**0.902**

Table 7
The recognition results of multiple methods on the PTD-D dataset.

Recognition model		VGG19	ResNet-101	AlexNet	Inception V3	Densenet-201	InceptionV3-BiLSTM	InceptionV3-GRU	CNN-GA	D-HCNN	OWIPA	Ours without C-AOG	Ours with C-AOG
AA		0.759	0.805	0.688	0.812	0.791	0.821	0.826	0.833	0.763	0.834	0.832	**0.875**
F1		0.725	0.774	0.657	0.779	0.759	0.786	0.792	0.793	0.729	0.796	0.797	**0.840**
AUC	C0	0.790	0.821	0.693	0.807	0.794	0.859	0.857	0.816	0.799	0.823	0.823	**0.884**
	C1	0.779	0.829	0.712	0.835	0.782	0.824	0.818	0.845	0.722	0.867	0.854	**0.868**
	C2	0.765	0.822	0.734	0.828	0.803	0.825	0.859	0.843	0.736	0.812	0.855	**0.873**
	C3	0.768	0.805	0.739	0.814	0.785	0.816	0.806	0.824	0.797	0.835	0.824	**0.889**
	C4	0.764	0.817	0.667	0.833	0.779	0.819	0.813	0.858	0.773	0.867	0.835	**0.868**
	C5	0.789	0.770	0.623	0.821	0.801	0.783	0.813	0.833	0.732	0.801	0.821	**0.854**
	C6	0.766	0.834	0.704	0.814	0.812	0.867	0.873	0.839	0.756	0.851	0.878	**0.903**
	C7	0.721	0.766	0.632	0.799	0.786	0.785	0.788	0.802	0.721	0.818	0.812	**0.855**
	C8	0.730	0.789	0.684	0.794	0.793	0.826	0.828	0.845	0.803	0.802	0.811	**0.861**
	C9	0.721	0.801	0.689	0.774	0.772	0.805	0.811	0.823	0.795	0.824	0.809	**0.890**

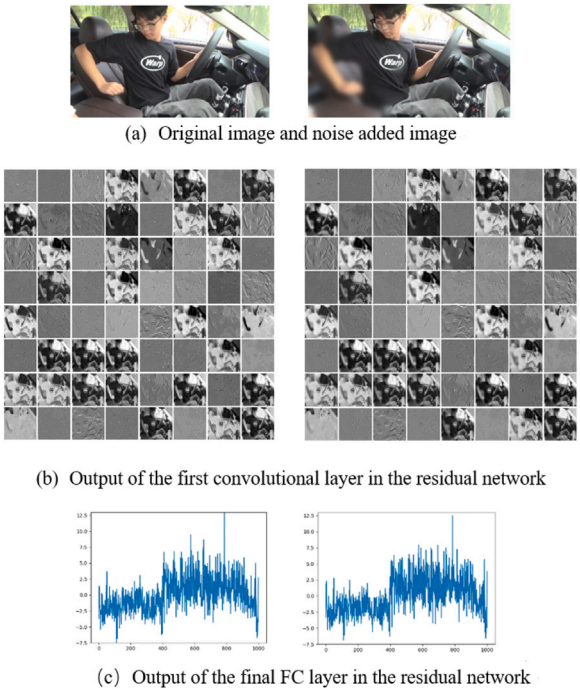
proposed method obtains a state-of-the-art result, and the average classification accuracy is 89.5%, and the F1 value is 0.863. Both indicators are higher than other methods.

We further verify the noise immunity and causal inference ability of various methods on the PTD-D. As shown in Table 7, when classifying the data samples in PTD-D, due to the rapid sparseness of image features or the occlusion of key features, most DL methods based on CNN or CNN-RNN will suffer performance degradation. In Fig. 30, we give the features from different layers of ResNet-101 when classifying a sample in PTD-D. In Fig. 30(a), we show two input images, one is the original image and the other one is picked out from the PTD-D. The Fig. 30(b) and (c) are the outputs of the two images passing through the first convolutional layer and the final fully connected layer of the residual network. By comparing the results in Fig. 30(b) and (c), it can be seen that noise causes significant sparsity in the image feature obtained by convolution and eventually leads to large deviations in the output of the fully connected layer.

In addition, the posture extraction method using single frame can hardly identify accurate human body agent features due to the sparseness of image features or image noise. In Fig. 31(a), feature blur in a small range has little effect on bone feature extraction. While in Fig. 31(b)–(d), affected by sparse features or the occlusion between limbs on a large scope, the skeleton features will be partially missed, which will directly affect the performance of the behavior recognition models which adopt body fusion features as input.

To better illustrate the robustness and noise immunity of our model in noisy environment as well as occluded scenes, we compare the recognition performance of the proposed method and ResNet-101 in four typical scenes.

We compare the performance of the two methods in the scenario where the key features is blurred due to the image noise. The recognition probability curves of the two methods for distracting behaviors are shown in Fig. 32, the red curve represents the recognition probability of ResNet-101 for right-handed cell phone use behavior, the blue curve represents the recognition probability of proposed method, and the green dashed line represents the time scope of the image noise. When noise is added, the features in the cell phone region suffer short-time sparsity that significantly reduce the accuracy of ResNet-101 for distracted behavior detection. As for the proposed method, since the rapid loss of the phone is not accompanied by a hand drop, the MDM mechanism is triggered to suppress the behavior termination process



(a) Original image and noise added image

(b) Output of the first convolutional layer in the residual network

(c) Output of the final FC layer in the residual network

Fig. 30. The effect of image noise on the internal features of the residual network.

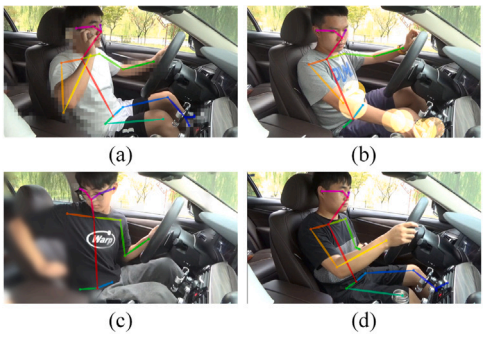


Fig. 31. The effect of noise and occlusion on skeleton feature recognition.

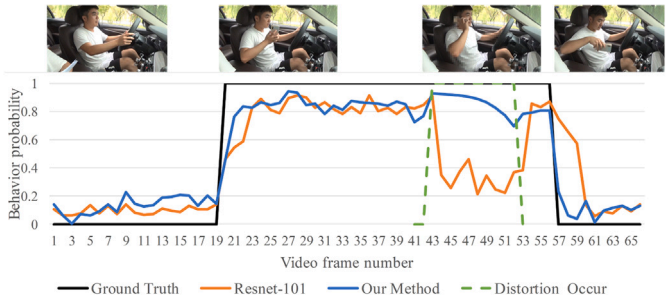


Fig. 32. Behavior recognition performance comparison between ResNet-101 and proposed method in the case of key features sparsity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

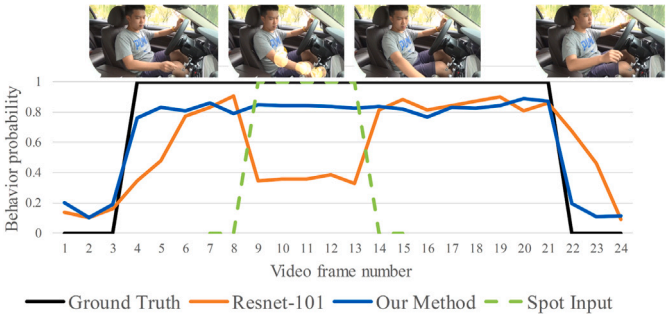


Fig. 33. Behavior recognition performance comparison between ResNet-101 and proposed method in the light spot noise environment.

with low confidence, so the probability of the true atomic action is maintained by Eq. (25) during the image noise addition process. As a result, the behavior fluent in the C-AOG will not changed from right phone usage to safe driving.

Similar to the results in Fig. 32, the comparison results in Figs. 33 and 34 also certified that proposed method has strong noise immunity in feature sparse scenes caused by light spot noise or body occlusion, while having better behavioral boundary recognition performance compared to ResNet-101.

Contextual misidentification, especially the object recognition failure, can make C-AOG inference erroneous and subsequently lead to logical contradictions in the behavioral analysis process. As shown in Fig. 35, we take drinking water as an example to illustrate the contextual misidentification suppression process based on counterfactual inference. When the water bottle is misidentified as cell phone, the model without counterfactual inference mechanism cannot accurately recognize the drinking behavior. While the model with noise suppression through the counterfactual inference mechanism can continuously and accurately acquire the real distracting behavior of the driver in a short period of time.

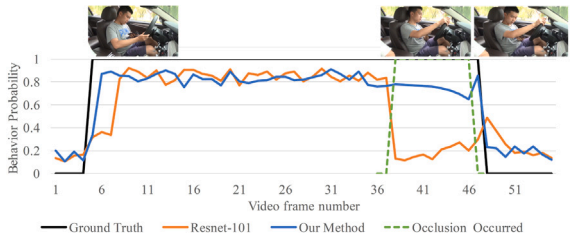


Fig. 34. Behavior recognition performance comparison between ResNet-101 and proposed method under body agent occlusion scene.

In Fig. 35, orange curve represents the probability of misidentified atomic action (holding cell phone), and the counterfactual reasoning mechanism suppresses probability of the wrong atomic action through the Eq. (25). On the other hand, probability changing of the real atomic action-right hand holds bottle or cup is taken over by the MDM mechanism since the bottle holding atomic action is abnormally terminated (no supporting condition such as right hand drop), thus MDM performing a slow probability curve decrease as shown in the purple curve of Fig. 35.

At last, we give the AUC values of various methods on different testing datasets in Fig. 36. From the AUC values, it can be seen that the recognition network consisting of CNN and RNN performs relatively well when dealing with static datasets consisting of single discrete video frame, while the classification performance of proposed method is significantly better when dealing with datasets consisting of continuous video frames, and the performance of proposed method remains optimal on the datasets with image noise or occluded situation. In addition, by comparing the C-AOG-DL fusion model with the method only based on DL model, it is clear that the addition of C-AOG plays an important role in improving the accuracy and robustness of the recognition model.

Compare to other SOTA method, by introducing the knowledge fence composed of C-AOG, the proposed method has short-term noise immunity and the ability to infer motion trend of the body agent. Even in the presence of noise and occlusion, as shown in Table 7, the classification accuracy of proposed method will not be greatly affected compare to the other SOTA method. In addition, compared with ResNet-101, proposed method is closer to the ground truth in terms of behavior boundary recognition, and ResNet-101 clearly shows time lag in the recognition of the beginning and end of the behavior. The advantages of the proposed method can be explained in three aspects: First, C-AOG predicts the behavior pattern by fusing multiple events and their temporal relation. So, compared with the DL method, the behavior pattern deduced by C-AOG is in line with common sense of causal logic and consistent in the time dimension. Secondly, even if the collected image affected by noise or local anomalies, skeleton features can still be effectively deduced to form the atomic behavior. Thus, the introduction of C-AOG can effectively improve the robustness of the recognition process. Thirdly, C-AOG has causal reasoning capabilities. Through counterfactual reasoning on the results of DL judgments, the logical contradictions in the judgment of behavior patterns can be detected, thereby C-AOG could further suppress the abnormal recognition results.

6. Summary and conclusion

In this paper, we proposed a distracted driving behavior recognition method by fusing deep learning model and C-AOG. The method firstly constructs a continuous recognition model for distracted driving behaviors based on a temporal–spatial double-line deep learning structure. The double-line model obtains the attention distribution features of distracting behaviors by the changing pattern of dynamic optical flow. An attention distribution features are then fused with the spatial features of

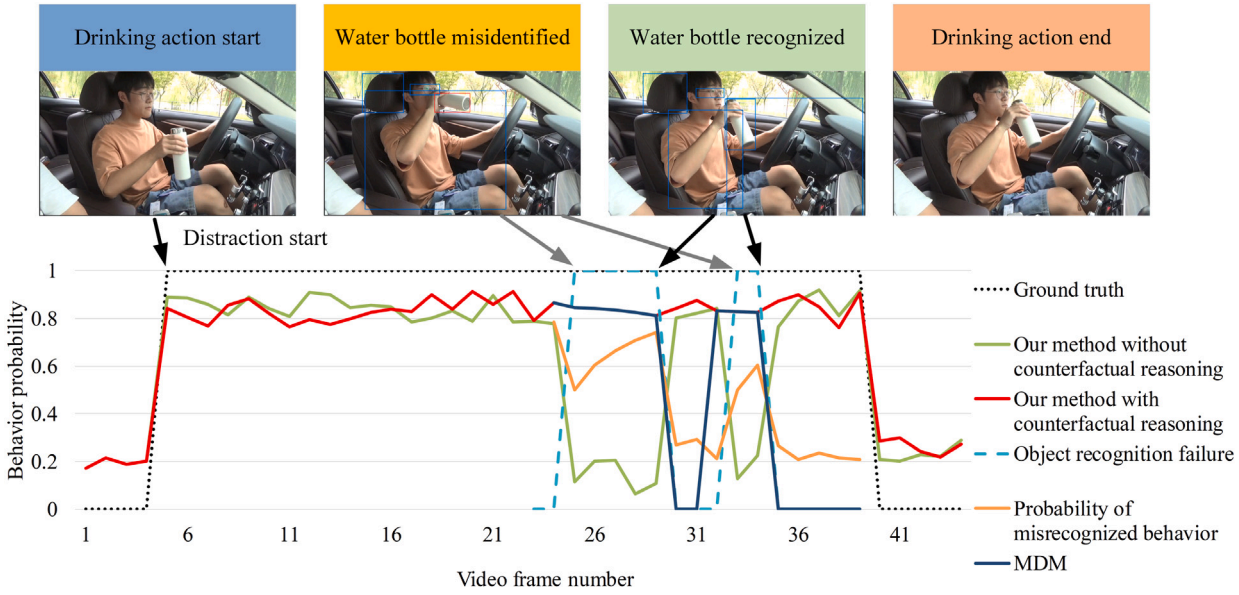


Fig. 35. Noise suppression by counterfactual reasoning under contextual misidentification. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

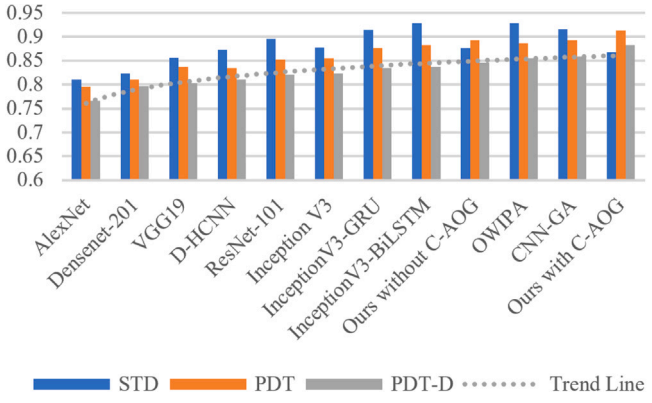


Fig. 36. AUC values of each type of behavior recognition model on different testing datasets.

single static frame. The double-line model finally recognizes the specific distracted driving behaviors reflected by the fused features through residual networks. The proposed double-line deep learning network can achieve better average recognition accuracy compared to traditional deep learning methods which rely on static image recognition.

In addition, we designed a C-AOG based behavioral inference model to countervail the lack of behavioral inference capability of deep learning. The inference model established an inferable association between image skeleton semantics and distracted behavior fluent through the events composed of atomic action. By establishing the causal graph between the semantics of video frames and behavior flow, the behavior recognition model can maintain a better performance when the video frame is affected by multiple image noise or the occlusion between body agents. The proposed method outperforms mainstream deep learning methods and achieves SOTA recognition performance in both open-source datasets and self-collected dataset. For samples in PTD-D, the average accuracy of the fusion model is 0.875 and the AUC value is 0.883. Meanwhile behavioral boundaries can be delineated more accurately by the fused model. In this paper, we verify the noise suppression ability of causal inference during behavioral analysis, but we did not give a further study on the impact of noise intensity and time duration on causal inference and deep learning methods. Thus, one of the interesting yet difficult tasks in our future work is to apply the developed algorithms to infer the behavior fluent of the agents in more complex systems [73,74].

CRediT authorship contribution statement

Peng Ping: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – review & editing, Discussed the results, Contributed to the revisions. **Cong Huang:** Conceptualization, Validation, Writing – review & editing, Discussed the results, Contributed to the revisions. **Weiping Ding:** Conceptualization, Methodology, Review & editing, Supervision, Funding acquisition, Discussed the results, Contributed to the revisions. **Yongkang Liu:** Software, Validation, Writing – review & editing, Discussed the results, Contributed to the revisions. **Miyajima Chiyomi:** Supervision, Project administration, Data curation, Discussed the results, Contributed to the revisions. **Takeda Kazuya:** Methodology, Resources, Supervision, Funding acquisition, Discussed the results, Contributed to the revisions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

We would like to thank 100 students and faculties from Nantong University who joined in the data collection experiment. This work is supported in part by the National Natural Science Foundation of China [grant numbers 61872425, 61976120], in part by the Natural Science Foundation of Jiangsu Province [grant number BK20191445], the Jiangsu Provincial Grant [JSSCBS20211109], the Natural Science Foundation for Colleges and Universities in Jiangsu Province [approval. No 2022-539], the Natural Science Key Foundation of Jiangsu Education Department [grant number 21KJA510004] and Nantong social livelihood science and technology project [MS12020078, approval. No 2022-227].

References

[1] National Highway Traffic Safety Administration, Overview of motor vehicle crashes in 2019, US Department of Transportation, Washington, DC, USA, 2020.

[2] World Health Organization, Global status report on road safety 2018: summary, Technical repor, World Health Organization, 2018.

[3] Saurabh R Shrivastava, Prateek S Shrivastava, Jegadeesh Ramasamy, Global plan for the decade of action for road safety: Expectations from developing nations, Saudi J. Med. Med. Sci. 2 (1) (2014) 57.

[4] Yifang Ma, Zhenyu Wang, Hong Yang, Lin Yang, Artificial intelligence applications in the development of autonomous vehicles: a survey, IEEE/CAA J. Autom. Sin. 7 (2) (2020) 315–329.

[5] Duarte Fernandes, António Silva, Rafael Névoa, Cláudia Simões, Dibet Gonzalez, Miguel Guevara, Paulo Novais, João Monteiro, Pedro Melo-Pinto, Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy, Inf. Fusion 68 (2021) 161–191.

[6] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, Saber Fallah, A survey of deep learning applications to autonomous vehicle control, IEEE Trans. Intell. Transp. Syst. 22 (2) (2020) 712–733.

[7] Shan Bao, Ling Wu, Bo Yu, James R. Sayer, An examination of teen drivers' car-following behavior under naturalistic driving conditions: With and without an advanced driving assistance system, Accid. Anal. Prev. 147 (2020) 105762.

[8] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[9] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, Matti Pietikäinen, Deep learning for generic object detection: A survey, Int. J. Comput. Vis. 128 (2) (2020) 261–318.

[10] Fahad Lateef, Yassine Ruichek, Survey on semantic segmentation using deep learning techniques, Neurocomputing 338 (2019) 321–348.

[11] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, Lisha Hu, Deep learning for sensor-based activity recognition: A survey, Pattern Recognit. Lett. 119 (2019) 3–11.

[12] Judea Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Elsevier, 2014.

[13] Binbin Qin, Jiangbo Qian, Yu Xin, Baisong Liu, Yihong Dong, Distracted driver detection based on a CNN with decreasing filter size, IEEE Trans. Intell. Transp. Syst. (2021).

[14] Hong Vin Koay, Joon Huang Chuah, Chee-Onn Chow, Yang-Lang Chang, Bhuvendhraa Rudrusamy, Optimally-weighted image-pose approach (OWIPA) for distracted driver detection and classification, Sensors 21 (14) (2021) 4837.

[15] Karthika Mohan, Judea Pearl, Graphical models for processing missing data, J. Amer. Statist. Assoc. (2021) 1–16.

[16] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, Saeid Nahavandi, A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Inf. Fusion 76 (2021) 243–297.

[17] Fabio Tango, Marco Botta, Real-time detection system of driver distraction using machine learning, IEEE Trans. Intell. Transp. Syst. 14 (2) (2013) 894–905.

[18] Andrei Aksjonov, Pavel Nedoma, Valery Vodovozov, Eduard Petlenkov, Martin Herrmann, Detection and evaluation of driver distraction using machine learning and fuzzy logic, IEEE Trans. Intell. Transp. Syst. 20 (6) (2018) 2048–2059.

[19] Wenbo Sun, Matthew Aguirre, Jionghua Judy Jin, Fred Feng, Samer Rajab, Shigenobu Saigusa, Jovin Dsa, Shan Bao, Online distraction detection for naturalistic driving dataset using kinematic motion models and a multiple model algorithm, Transp. Res. C 130 (2021) 103317.

[20] Tianchi Liu, Yan Yang, Guang-Bin Huang, Yong Kiang Yeo, Zhiping Lin, Driver distraction detection using semi-supervised machine learning, IEEE Trans. Intell. Transp. Syst. 17 (4) (2015) 1108–1120.

[21] Rafał J. Doniec, Szymon Sieciński, Konrad M. Duraj, Natalia J. Piaseczna, Katarzyna Mocny-Pachońska, Ewaryst J. Tkacz, Recognition of drivers' activity based on 1D convolutional neural network, Electronics 9 (12) (2020) 2002.

[22] Yu-Kai Wang, Shi-An Chen, Chin-Teng Lin, An EEG based brain computer interface for dual task driving detection, Neurocomputing 129 (2014) 85–93.

[23] Guofa Li, Weiquan Yan, Shen Li, Xingda Qu, Wenbo Chu, Dongpu Cao, A temporal-spatial deep learning approach for driver distraction detection based on EEG signals, IEEE Trans. Autom. Sci. Eng. (2021).

[24] Omid Dehzangi, Vaishali Sahu, Vikas Rajendra, Mojtaba Taherisadr, GSR-based distracted driving identification using discrete & continuous decomposition and wavelet packet transform, Smart Health 14 (2019) 100085.

[25] Chiyomi Miyajima, Kazuya Takeda, Driver-behavior modeling using on-road driving data: A new application for behavior signal processing, IEEE Signal Process. Mag. 33 (6) (2016) 14–21.

[26] Kenneth L. Campbell, The SHRP 2 naturalistic driving study: Addressing driver performance and behavior in traffic safety, Tr News (282) (2012).

[27] Anaïs Halin, Jacques G. Verly, Marc Van Droogenbroeck, Survey and synthesis of state of the art in driver monitoring, Sensors 21 (16) (2021) 5558.

[28] Furkan Omerustaoglu, C. Okan Sakar, Gorkem Kar, Distracted driver detection by combining in-vehicle and image data using deep learning, Appl. Soft Comput. 96 (2020) 106657.

[29] Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang, Driver activity recognition for intelligent vehicles: A deep learning approach, IEEE Trans. Veh. Technol. 68 (6) (2019) 5379–5390.

[30] Jimiama Mafeni Mase, Peter Chapman, Grazziela P Figueredo, Mercedes Torres Torres, Benchmarking deep learning models for driver distraction detection, in: International Conference on Machine Learning, Optimization, and Data Science, Springer, 2020, pp. 103–117.

[31] Imen Jegham, Anouar Ben Khalifa, Ihsen Alouani, Mohamed Ali Mahjoub, Soft spatial attention-based multimodal driver action recognition using deep learning, IEEE Sens. J. 21 (2) (2020) 1918–1925.

[32] Arup Kanti Dey, Bharti Goel, Sriram Chellappan, Context-driven detection of distracted driving using images from in-car cameras, Internet Things 14 (2021) 100380.

[33] Hesham M. Eraqi, Yehya Abouelnaga, Mohamed H. Saad, Mohamed N. Moustafa, Driver distraction identification with an ensemble of convolutional neural networks, J. Adv. Transp. 2019 (2019).

[34] Yuxin Zhang, Yiqiang Chen, Chenlong Gao, Deep unsupervised multi-modal fusion network for detecting driver distraction, Neurocomputing 421 (2021) 26–38.

[35] Andreas Holzinger, Bernd Malle, Anna Saranti, Bastian Pfeifer, Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI, Inf. Fusion 71 (2021) 28–37.

[36] Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, Heimo Müller, Causability and explainability of artificial intelligence in medicine, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 9 (4) (2019) e1312.

[37] Peng Cui, Susan Athey, Stable learning establishes some common ground between causal inference and machine learning, Nat. Mach. Intell. 4 (2) (2022) 110–115.

[38] Judea Pearl, Models, Reasoning and Inference, Vol. 19, CambridgeUniversity-Press, Cambridge, UK, 2000.

[39] Yuanlu Xu, Liang Lin, Wei-Shi Zheng, Xiaobai Liu, Human re-identification by matching compositional template with cluster sampling, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3152–3159.

[40] Wenguan Wang, Yuanlu Xu, Jianbing Shen, Song-Chun Zhu, Attentive fashion grammar network for fashion landmark detection and clothing category classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4271–4280.

[41] Amy Fire, Song-Chun Zhu, Learning perceptual causality from video, ACM Trans. Intell. Syst. Technol. (TIST) 7 (2) (2015) 1–22.

[42] Caiming Xiong, Nishant Shukla, Wenlong Xiong, Song-Chun Zhu, Robot learning with a spatial, temporal, and causal and-or graph, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 2144–2151.

[43] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, Léon Bottou, Discovering causal signals in images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6979–6987.

[44] Amy Fire, Song-Chun Zhu, Using causal induction in humans to learn and infer causality from video, in: Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 35, 2013.

[45] Yuanlu Xu, Lei Qin, Xiaobai Liu, Jianwen Xie, Song-Chun Zhu, A causal and-or graph model for visibility fluent reasoning in tracking interacting objects, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2178–2187.

[46] Xuanpeng Li, Qifan Xue, Jingwen Zhao, Dong Wang, Causal reasoning in multi-object interaction on the traffic scene: Occlusion-aware prediction of visibility fluent, IEEE Access 8 (2020) 80527–80535.

[47] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[48] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.

[49] Quanshi Zhang, Jie Ren, Ge Huang, Ruiming Cao, YingNian Wu, Song-Chun Zhu, Mining interpretable AOG representations from convolutional networks via active question answering, IEEE Trans. Pattern Anal. Mach. Intell. (2020).

[50] Yifei Huang, Minjie Cai, Zhenqiang Li, Yoichi Sato, Predicting gaze in egocentric video by learning task-dependent attention transition, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 754–769.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[52] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[53] Thomas Brox, Andrés Bruhn, Nils Papenberg, Joachim Weickert, High accuracy optical flow estimation based on a theory for warping, in: European Conference on Computer Vision, Springer, 2004, pp. 25–36.

[54] Haoran Wang, Yue Fan, Zexin Wang, Licheng Jiao, Bernt Schiele, Parameter-free spatial attention network for person re-identification, 2018, arXiv preprint arXiv:1811.12150.
[55] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, Tat-Seng Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5659–5667.
[56] Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.
[57] Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7291–7299.
[58] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, Yolov4: Optimal speed and accuracy of object detection, 2020, arXiv preprint arXiv:2004.10934.
[59] Mingtao Pei, Yunde Jia, Song-Chun Zhu, Parsing video events with goal inference and intent prediction, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 487–494.
[60] Deng Jia, Dong Wei, Socher Richard, Li-Jia Li, Kai Li, Fei-Fei Li, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[61] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, C Lawrence Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[62] Yin Li, Alireza Fathi, James M. Rehg, Learning to predict gaze in egocentric video, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3216–3223.
[63] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, Kilian Q Weinberger, Snapshot ensembles: Train 1, get m for free, 2017, arXiv preprint arXiv:1704.00109.
[64] Ilya Loshchilov, Frank Hutter, Sgdr: Stochastic gradient descent with warm restarts, 2016, arXiv preprint arXiv:1608.03983.
[65] Munif Alotaibi, Bandar Alotaibi, Distracted driver classification using deep learning, Signal Image Video Process. 14 (3) (2020) 617–624.
[66] Peng Ping, Yuan Sheng, Wenhu Qin, Chiyomi Miyajima, Kazuya Takeda, Modeling driver risk perception on city roads using deep learning, IEEE Access 6 (2018) 68850–68866.
[67] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, Image database TID2013: Peculiarities, results and perspectives, Signal Process., Image Commun. 30 (2015) 57–77.
[68] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[69] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[70] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, Kilian Q Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[71] James A. Hanley, Barbara J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1) (1982) 29–36.
[72] André M. Carrington, Paul W. Fieguth, Hammad Qazi, Andreas Holzinger, Helen H. Chen, Franz Mayr, Douglas G. Manuel, A new concordant partial AUC and partial c statistic for imbalanced data in the evaluation of machine learning algorithms, BMC Med. Inf. Decis. Making 20 (1) (2020) 1–12.
[73] Xingzhen Bai, Zidong Wang, Lei Zou, Fuad E. Alsaadi, Collaborative fusion estimation over wireless sensor networks for monitoring CO2 concentration in a greenhouse, Inf. Fusion 42 (2018) 119–126.
[74] Jian Ding, Shuli Sun, Jing Ma, Na Li, Fusion estimation for multi-sensor networked systems with packet loss compensation, Inf. Fusion 45 (2019) 138–149.

PENG PING received the B.S. degree in automation from the Beijing University of Chemical Technology, Beijing, China, in 2010, and the M.S.degree in automation from the Nanjing University of Science and Technology, Nanjing, China, in 2013. In 2020, he received the Ph.D degree in instrument science and technology from the Southeast University, Nanjing, China. In 2017, he went to Nagoya University as a joint Ph.D. Student. From 2013 to 2015, he was a Research and Development Engineer as part of the Cloud switch Group, Huawei Technologies Co., Ltd. Since 2020, he has been a Assistant Professor with the School of Transportation and Civil Engineering, Nantong University, Nantong, China. His research interests include vehicle safety, data-mining, and driving behavior modeling.

CONG HUANG received his B. Eng. degree in electrical engineering and automation from Nanjing Forestry University, Nanjing, China, in 2016 and the Ph.D. degree in control science and engineering from Donghua University, Shanghai, China, in 2021. He is currently a lecturer with the School of Transportation and Civil Engineering, Nantong University, Nantong, China. From 2019 to 2020, he was a Visiting Ph.D. Student in the Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy. His current research interests include fusion estimation, recursive filtering, network communication and autonomous vehicles. He is a very active reviewer for many international journals.

WEIPING DING (M'16-SM'19) received the Ph.D. degree in Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. From 2014 to 2015, he is a Postdoctoral Researcher at the Brain Research Center, National Chiao Tung University , Hsinchu, Taiwan. In 2016, He was a Visiting Scholar at National University of Singapore, Singapore. From 2017 to 2018, he was a Visiting Professor at University of Technology Sydney, Ultimo, NSW, Australia. He is currently a professor with the School of Information Science and Technology, Nantong University, Nantong, China. His research interests include deep neural networks, multimodal machine learning, granular data mining, uncertainty modeling in big data, co-evolutionary algorithm, and medical images analysis. He has published more than 100 journal papers. Dr. Ding is vigorously involved in editorial activities. He servedserves on the Editorial Advisory Board of Knowledge-Based Systems (Elsevier) and Editorial Board of Information Fusion (Elsevier), Engineering Applications of Artificial Intelligence (Elsevier) and Applied Soft Computing (Elsevier). He served/serves as an Associate Editor of IEEE Transactions on Neural Network and Learning System, IEEE Transactions on Fuzzy Systems, IEEECAA Journal of Automatica Sinica, Information Sciences (Elsevier), Neurocomputing (Elsevier), Swarm and Evolutionary Computation (Elsevier), IEEE Access and Journal of Intelligent & Fuzzy Systems, and Co-Editor-in-Chief of Journal of Artificial Intelligence and System.

YONGKANG LIU received the Ph.D. and M.S. degrees in electrical engineering from the University of Texas at Dallas in 2021 and 2017, and the B.S. degree in electronic information engineering from Shandong Normal University in 2015, respectively. His research interests focus on in-vehicle systems, advancements in smart vehicle technologies, and driver behavior modeling.

CHIYOMI MIYAJIMA received the B.E., M.E., and Ph.D. degrees in computer science from the Nagoya Institute of Technology, Japan, in 1996, 1998, and 2001, respectively. From 2001 to 2003, she was a Research Associate with the Department of Computer Science, Nagoya Institute of Technology. She was an Assistant Professor with the Graduate School of Information Science, Nagoya University, Japan, from 2003 to 2016. She was an Associate Professor with the Institutes of Innovation for Future Society, Nagoya University, from 2016 to 2018. From 2018 to 2020, she was an Associate Professor with Daido University, Nagoya. Since 2020, she has been a Professor with Daido University, Nagoya Japan. Her research interests include the analysis and the modeling of driver behavior.

KAZUYA TAKEDA (SM'09) received the B.E. and M.E. degree in electrical engineering and the Dr.Eng. degree from Nagoya University, Nagoya, Japan, in 1983, 1985, and 1994, respectively. From 1986 to 1989, he was with the Advanced Telecommunication Research Laboratories, Osaka, Japan. His research interest at ATR was corpus-based speech synthesis. He was a Visiting Scientist with the Massachusetts Institute of Technology, Cambridge, from November 1987 to April 1988. From 1989 to 1995, he was a Researcher and Research Supervisor with KDD Research and Development Laboratories, Kamifukuoka, Japan. From 1995 to 2003, he was an Associate Professor with the Faculty of Engineering, Nagoya University. Since 2003, he has been a Professor with Graduate School of Information Science, Nagoya University. And he is also the vice president of the Nagoya University. His main focus is investigating driving behavior using data centric approaches, utilizing signal corpora of real-driving behavior. He is a member of the Board of Governors of the IEEE Intelligent Transportation Systems Society. He is an author or coauthor of more than 100 journal papers, six books, and more than 100 conference proceeding papers. His current research interests are media signal processing and its applications, including spatial audio, robust speech recognition, and driving behavior modeling.