E-DNet: An End-to-End Dual-Branch Network for Driver Steering Intention Detection

Youjia Fu 1, Huixia Xue 1,*, Junsong Fu 1 and Zihao Xu 2

1 College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China; youjia_fu@cqut.edu.cn (Y.F.); juns_fu@stu.cqut.edu.cn (J.F.)
2 Liangjiang Institute of Artificial Intelligence, Chongqing University of Technology, Chongqing 401135, China; xu139@stu.cqut.edu.cn
* Correspondence: xue_xue@stu.cqut.edu.cn

Abstract: An advanced driving assistant system (ADAS) is critical for improving traffic efficiency and ensuring driving safety. By anticipating the driver’s steering intentions in advance, the system can alert the driver in time to avoid a vehicle collision. This paper proposes a novel end-to-end dual-branch network (EDNet) that utilizes both in-cabin and out-of-cabin data. In this study, we designed an in-cabin driver intent feature extractor based on 3D residual networks and atrous convolution, which is applicable to video data and is capable of capturing a larger range of driver behavior. In order to capture the long-term dependency of temporal data, we designed the depthwise-separable max-pooling (DSMax) module and combined it with a convolutional LSTM to obtain the road environment feature extractor outside the cabin. In addition, to effectively fuse different features inside and outside the cockpit, we designed and propose the dynamic combined-feature attention fusion (D-CAF) module. EDNet employs a freeze-training method, which enables the creation of a lightweight model while simultaneously enhancing the final classification accuracy. Extensive experiments on the Brain4Cars dataset and the Zenodo dataset show that the proposed EDNet was able to recognize the driver’s steering intention up to 3 s in advance. It outperformed the existing state of the art in most driving scenarios.

Keywords: driver safety; maneuver prediction; ADAS; driver intention; recurrent neural network

1. Introduction

In recent years, as the number of vehicles continues to increase, traffic congestion on the roads has gradually worsened, and traffic accidents are frequent. Globally, the number of deaths due to road traffic accidents reached 1.35 million, and approximately 50 million people are injured in road traffic accidents every year [1]. These accidents mainly include collisions between vehicles, vehicle–pedestrian collisions, and vehicle–non-motorized vehicle collisions. The accidents are mainly due to speeding, fatigued driving, and distracted driving. There are also accidents caused by external factors such as adverse weather conditions and poorly maintained roads. In light of the alarming data on traffic accident casualties, engineers are starting to seek new solutions.

An advanced driver assistance system (ADAS) is considered one of the most crucial ways to potentially decrease the risk of accidents and enhance road safety. An ADAS is an in-vehicle intelligent technology that typically utilizes sensors and cameras to gather vehicle information and provide warnings and recommendations to the driver. However, the ADAS supplied by current vehicles offers alerts based only on the vehicle’s operational status. The lack of environmental contextual information and driver behavioral data can result in delayed warnings that do not provide drivers with sufficient time to react [2]. If an ADAS can predict the driver’s intentions a few seconds in advance, the system can prepare before the driver maneuvers the vehicle, assisting the driver in controlling the vehicle in time to avoid a collision.
In recent years, many studies on driving have emerged [3–5]. These studies involved modeling using a variety of data, including vehicle dynamics data, driver state data, and external road environment data [6–10]. Many studies used both driver state data and external road environment data [11–14]. They found that this method of combining the two types of data is more effective than using either type of data alone. Although these studies used multi-feature splicing and demonstrated its effectiveness in their studies, simple feature splicing does not handle the correlation between multiple features very well [15,16].

In addition, some methods try to use a 3D CNN to deal with time series data or data with a temporal dimension, such as video. However, this approach results in an excessive number of model parameters, which hinders its performance in practical applications.

Therefore, when conducting driver steering intention research, the number of model parameters needs to be reduced while maintaining accuracy in order to develop a more efficient and accurate driver steering intention recognition system. In this paper, an end-to-end dual-branch network (EDNet) is proposed, which integrates in-cockpit and out-of-cockpit data based on driver cognition. The main contributions of our paper are as follows:

1. For in-cockpit video data, we propose a novel driver intent feature extractor called Atrous-3DResNet 50 (A-3DResNet 50). It incorporates atrous convolution in the initial part of 3DResNet to lighten the model’s complexity while capturing a broader range of driver behaviors.

2. To effectively capture long-term dependencies in time series data, we designed the depthwise-separable max-pooling (DSMax) module and integrated it with ConvLSTM to form the ConvLSTM-4DSMax road environment feature extractor for vehicle operations. This enhancement improves the model’s processing of time series information, closely aligning with the requirements of the research task for such data.

3. Aiming to integrate the features of the video data, both inside and outside the cockpit, we propose a multi-feature fusion strategy and designed a feature fusion module called dynamic combined-feature attention fusion (D-CAF) based on the attention mechanism. This method effectively integrates various features both inside and outside the cockpit to enhance the accuracy of the model in classifying the driver’s steering intention.

The rest of the paper is organized as follows: Section 2 reviews previous related work. Section 3 explains the proposed methods and modules in detail. Section 4 presents the dataset and discusses the experimental results. Section 5 provides the conclusions of the paper, as well as suggestions for future work.

2. Related Work

Driver steering intention is a cognitively driven task that results from the interaction of the driver, the road environment, and the vehicle. The driver perceives information from the external road environment through vision and hearing, and evaluates incoming information before formulating intentions and making decisions. Figure 1 displays the cognitive process of forming the driver’s steering intention.

![Figure 1. The cognitive process of forming the driver's steering intention.](image_url)
nizing driver steering intention is fundamentally a sequence prediction problem based on a time window. Driving behavior data within a specific time range is typically analyzed to predict and identify the driver’s steering intention. These driving behavior data cover the road environment information, vehicle motion state, and driver behavior.

Commonly used generative models include the hidden Markov model (HMM) and dynamic Bayesian network (DBN). Jain et al. [6] proposed an autoregressive input–output-based HMM (AIO-HMM) to predict the driver’s intention by utilizing driving behavior and road environmental information. He et al. [17] employed a dynamic Bayesian network to recognize the lane-changing behavior of surrounding vehicles in a highway scenario. They predicted the vehicle’s motion state and generated a trajectory based on this analysis. He et al. [18] designed a double-layer HMM structure, where the bottom layer multi-dimensional Gaussian HMM (MGHMM) integrates driver behavior with the vehicle motion state to identify driving behaviors. The recognition results are then forwarded to the upper layer multi-dimensional discrete HMM (MDHMM) to determine the final driving intention. Zabihi et al. [19] developed a maneuver prediction model using input–output HMM (IOHMM) to detect driver intentions by considering the vehicle’s motion state and driver behavior.

Discriminative models, such as an artificial neural network (ANN) and support vector machine (SVM), are also widely used in the field of driver steering intention recognition [20]. Kim et al. [21] utilized an ANN to analyze vehicle motion data, including the steering wheel angle, yaw rate, and throttle position, to classify the road conditions and predict the driver’s intention to change lanes. Leonhardt et al. [22] employed an ANN approach to predict the driver’s intention to change lanes based on the driver’s behavior and vehicle motion state. Morris et al. [23] developed a Bayesian extension of the popular SVM called a relevance vector machine (RVM) to distinguish between lane changes and lane keeping. On this basis, a real-time road prediction system was developed that is able to detect a driver’s intention to change lanes a few seconds before it occurs.

With the rapid development of deep learning technology, recurrent neural networks (RNNs) have become a research hotspot due to their advantages in dealing with time-dependent problems. A number of studies showed that long short-term memory (LSTM) networks outperform traditional machine learning and standard recurrent neural networks in terms of performance. They demonstrated their importance and superiority in modeling driver behavior [24–26]. Zyner et al. [27] proposed a recurrent neural network-based prediction method using LIDAR to acquire vehicle motion data to predict driver intent at roundabout intersections. Zhou et al. [28] designed a cognitive fusion recurrent neural network (CF-RNN) based on a cognitively driven model and data-driven model, which consists of two LSTM branches to fuse road environment information and driver behavior information to predict driver intent. Kim et al. [29] proposed a trajectory prediction method using LSTM that utilizes vehicle motion data collected from motorways to analyze temporal behavior and predict the location of surrounding vehicles. Transformer architectures also achieved significant results in areas such as image classification, trajectory prediction, and time series prediction [30]. Gao et al. [31] proposed a transformer-based integrated model that includes a lane change intention prediction model and a lane change trajectory prediction model. This model was designed to jointly predict lane change intentions and trajectories for vehicles operating in a mixed traffic environment, which involves both human-driven and autonomous vehicles. Chen et al. [32] also proposed an intention-aware non-autoregressive transformer model with multi-attention learning based on the Transformer architecture to accurately predict multimodal vehicle trajectories.

In summary, the research field of driver steering intention covers a variety of algorithms, including generative models, discriminative models, and deep learning. These studies typically analyze multifaceted data, such as road environment information, the vehicle motion state, and driver behavior information, to accurately recognize and predict driver steering intentions.
Several studies utilized vehicle motion states to analyze driver steering intentions. However, vehicle operating states primarily stem from driving behaviors, which reflect how driver steering intentions manifest in the vehicle. Therefore, vehicle motion states are not suitable for directly recognizing driver steering intentions, especially when advanced prediction is required. In addition, most of the previous studies were conducted based on manually encoded features, and many of the models have large parameter counts, which is inconsistent with the requirement for lightweight criteria in real vehicle applications.

Therefore, taking into account the driver’s cognitive perspective, this work utilized an attentional mechanism to incorporate both in-cockpit features related to driver behavior and out-of-cockpit features concerning the road environment, with the aim to accurately identify the driver’s steering intention.

3. Method

3.1. Overall Framework

This paper proposes a model that leverages both in-cockpit and out-of-cockpit video data to detect the driver’s steering intention prior to vehicle maneuvering. For the in-cockpit video, the driver intent feature extractor A-3DResNet 50 was constructed based on 3DResNet and atrous convolution. For the out-of-cockpit video, the video data are converted to optical flow images, and the DSMax module was designed based on depthwise separable convolution. The DSMax module combines ConvLSTM to construct the road environment feature extractor ConvLSTM-4DSMax. The D-CAF module is used to combine these two features. Subsequently, a classifier was built, consisting of a fully connected layer and a softmax layer, to identify and predict five types of vehicle maneuvers: straight ahead (straight), left lane change (Lchange), left turn (Lturn), right lane change (Rchange), and right turn (Rturn).

The structure of EDNet is shown in Figure 2. Considering the need for lightweighting the model and drawing inspiration from transfer learning, the freeze-training strategy was implemented for A-3DResNet 50 and ConvLSTM.

![Figure 2. The overview of the proposed EDNet structure.](image)

3.2. Driver Intent Feature Extractor

The 3DResNet excels in video data processing, especially for human action recognition [33]. Its ability to handle video sequences of varying lengths and sizes makes it ideal for cockpit video data processing. In deep learning, several 3DResNet variants with different architectural depths are commonly used, including 3DResNet 18, 3DResNet 34, 3DResNet 50, 3DResNet 101, and 3DResNet 152. Each variant is tailored with distinct strengths and weaknesses, which become more significant depending on the task at hand.

The characterization and learning capabilities of the shallow network 3DResNet 18 are limited. Its effectiveness is not very satisfactory in handling complex and high-dimensional data due to its comparatively low learnable feature complexity. On the other hand, the computational resource needs and model complexity of the 3DResNet 152 deep network are much higher. This results in increased hardware resource demands and delays in the training and inference processes.
Hence, to strike a balance between the representation capacity, model complexity, and computational resource demands, we focused on the 3DResNet 34, 3DResNet 50, and 3DResNet 101 variants from the 3DResNet series. These networks offer significant representation capabilities while being relatively straightforward to train and requiring moderate computational resources.

Figure 3 displays the structures of 3DResNet 34, 3DResNet 50, and 3DResNet 101, which share a similar structure overall and are composed of five stages. Stage 1 incorporates convolutional and max-pooling layers for the spatial feature extraction and fusion of 3D data. Stages 2 to 5 include a varying number of residual blocks. The grey module represents the basic block, which is a residual block commonly employed in shallower ResNet. The blue module denotes the bottleneck block, which is utilized in deeper ResNet to reduce parameters, thereby effectively mitigating the model’s demand for graphics memory and computational resources.

Figure 3. The structures of 3DResNet 34, 3DResNet 50, and 3DResNet 101.

Although 3D ResNet works well for processing 3D data, it still has challenges with processing huge video frames and capturing long-term temporal information. Thus, we built the A-3DResNet family of networks for driver intent feature extraction inside the cockpit by introducing atrous convolution in stage 1 of 3D ResNet.

Atrous convolutional operations expand the receptive field without increasing the feature map size. This enhances the capture of temporal and spatial information, preventing information loss. Two key considerations underpin this design:

- Enhancing feature extraction: Atrous convolutions play a vital role in integrating both local and global information. Incorporating atrous convolutions in the initial stage of 3D ResNet enhances the feature extraction efficiency and reduces the information loss.
- Reducing model complexity: introducing atrous convolution in the early stages of the model can limit the number of network parameters and computational complexity, thus preventing the addition of excessive computational burden in the subsequent stages.

Applied to the task of driver steering intention recognition, we utilized Atrous-3DResNet 50 (A-3DResNet 50) as a driver intent feature extractor within the cockpit. The model takes in 16 frames per second that are uniformly sampled from the video, with a frame image size of 112 × 112. The structure of A-3DResNet 50 is shown on the left side of Figure 2. Table 1 presents the parameter information for each module in the model. In the output size column, the first number of every line indicates the feature map channel count, and the remaining ones indicate the feature map spatial dimensions.

Table 1. Parameter information for each module of the A-3DResNet 50.

<table>
<thead>
<tr>
<th>Module</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 1</td>
<td>64 × 8 × 28 × 28</td>
</tr>
<tr>
<td>Stage 2</td>
<td>256 × 8 × 28 × 28</td>
</tr>
<tr>
<td>Stage 3</td>
<td>512 × 4 × 14 × 14</td>
</tr>
<tr>
<td>Stage 4</td>
<td>1024 × 2 × 7 × 7</td>
</tr>
<tr>
<td>Stage 5</td>
<td>2048 × 1 × 4 × 4</td>
</tr>
</tbody>
</table>
3.3. Road Environment Feature Extractor

Convolutional LSTM (ConvLSTM) is a deep learning model that combines CNN and RNN [34]. It is frequently employed for handling spatiotemporal data in the format of data frames. We designed the DSMax module based on depthwise separable convolution. ConvLSTM and DSMax modules were combined to create ConvLSTM-4DSMax, which is a road environment feature extractor outside the cockpit.

3.3.1. Optical Flow Prediction

Optical flow characterizes the changes in the image and captures the motion information of the target in successive frames of the video. Therefore, we used FlowNet 2.0 [35] to generate optical flow maps containing information about the relative motion between vehicles and other traffic participants as input into ConvLSTM.

ConvLSTM combines the efficient high-dimensional data extraction capabilities of a CNN with the memory function of LSTM, making it suitable for processing complex temporal data, such as speech and video. Its operation is encapsulated by the following equations:

\begin{align}
    i_t &= \sigma([H_{t-1}, X_t, C_{t-1}] \ast W_i + b_i) \\
    f_t &= \sigma([H_{t-1}, X_t, C_{t-1}] \ast W_f + b_f) \\
    C_t &= f_t \odot C_{t-1} + i_t \odot \tanh([H_{t-1}, X_t] \ast W_c + b_c) \\
    o_t &= \sigma([H_{t-1}, X_t, C_{t-1}] \ast W_o + b_o) \\
    H_t &= o_t \odot \tanh(C_t)
\end{align}

where \(i_t\) represents the input gate, \(f_t\) represents the forget gate, \(C_t\) represents the cell output, \(o_t\) represents the output gate, \(H_t\) represents the hidden state, \(\sigma\) represents the sigmoid function, \(\ast\) represents the convolution operation, \(\odot\) represents bitwise multiplication, and \(\oplus\) represents bitwise addition.

Figure 4 displays the structure of ConvLSTM. Its input is five frames, which are obtained by uniformly sampling from the video data, and the input size of each frame is 112 × 176.

![Figure 4](image)

**Figure 4.** The structure of ConvLSTM. \(h_{i,j}\) denotes the hidden state and \(C_{i,j}\) denotes the cell state, where \(i\) denotes the time step and \(j\) denotes the layer number.
3.3.2. DSMax Module

To overcome the computational inefficiencies of standard convolutions in deep networks, this study introduced the depthwise-separable max-pooling (DSMax) module, which combines depthwise separable convolution (DSC) and max pooling. The module not only accelerates the model training and inference but also provides optical flow features predicted by ConvLSTM for subsequent feature fusion.

DSC consists of depthwise convolution (DWConv) and pointwise convolution (PWConv). In DWConv, a convolution kernel is applied to perform a convolution operation on each channel of the input feature map. The output of each channel is then spliced to obtain the final output. PWConv actually refers to a $1 \times 1$ convolution, which serves two roles in the DSC:

- Adjusting the number of channels: stand-alone DWConv cannot change the number of output channels; PWConv is used to adjust the number of output channels.
- Implementing channel fusion: PWConv is used to perform channel fusion operations on the feature maps output from DWConv so as to effectively integrate feature information.

The DSMax module applies max pooling to the output of the DSC. It reduces the feature map dimensions while retaining important features. The detailed structure of the DSMax module is illustrated in Figure 5.

![Figure 5. The structure of the DSMax module.](image)

3.3.3. ConvLSTM-4DSMax Model Implementation

The integration of ConvLSTM with four DSMax modules results in the ConvLSTM-4DSMax, which is an effective feature extractor for the road environment beyond the cockpit. ConvLSTM is employed to predict optical flow characteristics exterior to the cockpit, whereas the DSMax modules are responsible for the extraction and refinement of critical features from the data. The ConvLSTM-4DSMax can handle complex spatiotemporal data, enabling the successful extraction of deep feature correlations embedded within temporal sequences.

The structure of ConvLSTM-4DSMax is shown on the right side of Figure 2. The parameter information of each module in ConvLSTM-4DSMax is shown in Table 2. In the output size column, the first number of every line indicates the feature map channel count, and the remaining ones indicate the feature map dimensions.
Table 2. Parameter information of each module in ConvLSTM-4DSMax.

<table>
<thead>
<tr>
<th>Module</th>
<th>Layer</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvLSTM</td>
<td>Layer 0</td>
<td>128 × 112 × 176</td>
</tr>
<tr>
<td></td>
<td>Layer 1</td>
<td>64 × 112 × 176</td>
</tr>
<tr>
<td></td>
<td>Layer 2</td>
<td>64 × 112 × 176</td>
</tr>
<tr>
<td></td>
<td>Layer 3</td>
<td>32 × 112 × 176</td>
</tr>
<tr>
<td>DSMax</td>
<td>DSMax_0</td>
<td>64 × 37 × 59</td>
</tr>
<tr>
<td></td>
<td>DSMax_1</td>
<td>128 × 12 × 20</td>
</tr>
<tr>
<td></td>
<td>DSMax_2</td>
<td>256 × 4 × 7</td>
</tr>
<tr>
<td></td>
<td>DSMax_3</td>
<td>512 × 1 × 2</td>
</tr>
</tbody>
</table>

3.4. D-CAF Module

Driver intention features (in-features) inside the cockpit are obtained by A-3DResNet 50, and road environment features (out-features) outside the cockpit are obtained by ConvLSTM-4DSMax. For these two features, a dynamic combined-feature attention fusion (D-CAF) module was designed in this study, and Figure 6 shows the structure flow of the D-CAF module.

\[
AW = \text{sigmoid}(WX + b) = \frac{1}{1 + e^{-(WX+b)}}
\]  

where \(W\) is the weight of the linear layer, \(X\) is the input feature vector, and \(b\) is the bias.

Attention weights are obtained by applying a sigmoid activation function to map the values between 0 and 1. The weight matrix \(W\) and bias \(b\) are learned during the model-training process by continuously adjusting the values of the weight matrix to better model the input data. The combined features undergo an element-wise multiplication operation with the attention weights to obtain attention-weighted features (AW-Features). The D-CAF module achieves dynamic fusion of features both inside and outside the cockpit, enabling the model to adaptively focus on feature information from various parts and merge them.

3.5. Loss Function

The loss function is employed to quantify the disparity between the true intention of the driver and the intention recognized by the model. The smaller the difference, the more adept the model is at mapping inputs to outputs. In this study, the cross-entropy loss function was used in the in-cockpit experiments, and the mean squared error loss function was used in the out-of-cockpit experiments and the joint in-cockpit and out-cockpit experiments.
The cross-entropy loss function is commonly used in classification problems. In multi-classification problems, the expression of the cross-entropy loss function is shown below:

\[
CE(p, q) = - \sum_{i=1}^{C} p_i \log(q_i)
\]

(7)

where \(C\) represents the number of categories, \(p_i\) represents the value of the class, \(i\) is the true labeling, and \(q_i\) represents the predicted probability.

The mean squared error loss function is used to calculate the mean of the sum of squares of the errors between the predicted data and the original data. The expression of the mean squared error loss function is shown below:

\[
MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2
\]

(8)

where \(N\) represents the number of samples, \(\hat{y}_i\) represents the predicted value, and \(y_i\) represents the target value.

3.6. Freeze-Training Strategy

In a multi-network model, each network is designed for a specific task, whereas the final model needs to consider all networks together to achieve the overall goal. In practice, each network performs well on individual tasks. However, when integrated into a unified framework, the results are not satisfactory. The main reason is that the features required for the joint task are overly complex. This complexity makes it challenging for the model to capture all the information during the learning process, ultimately impacting its performance.

To solve this problem, we employed the freezing strategy during training, which involves fixing the parameters of A-3DResNet 50 and ConvLSTM. This enables the model to concentrate its resources on training subsequent layers, leveraging knowledge from previous weight files. Compared with building the model from scratch, this method is more robust and mitigates overfitting risks, making it better suited for real-world applications.

4. Experiments

4.1. Dataset

In this study, the proposed method was evaluated using the Brain4cars and Zenodo datasets, and samples from both datasets are shown in Figure 7. Videos in both datasets recorded the process of generating driver steering intentions, but did not include the process by which the driver actually operates the steering of the vehicle. Therefore, they were suitable for evaluating the ability of the model to be accurate and predictive.
The Brain4Cars dataset [6]: Includes simultaneously recorded driver-facing (1920 px × 1088 px, 30 fps) and road-facing (720 px × 480 px, 30 fps) videos. Five types of maneuvers are defined: straight, Lchange, Lturn, Rchange, and Rturn, which encompass the driver’s driving behavior before the actual maneuver. In addition to the videos, the dataset contains supplementary information extracted from external cameras and GPS, such as the lane number, where the vehicle is located and the total number of road lanes. The Zenodo dataset [25]: Videos were provided from both driver-facing (1048 px × 810 px, 30 fps) and road-facing (1620 px × 1088 px, 30 fps) viewpoints, following the recording standards of the Brain4Cars dataset. In the laboratory, vehicles were simulated using a game simulator to replicate motorway and city driving conditions. This was achieved using a three-screen display system and driving equipment, including pedals, gears, and a force-feedback steering wheel. The collected data were annotated and processed to include 113 videos covering five maneuvers: straight, Lchange, Lturn, Rchange, and Rturn.

In the process of converting the video into frames, we found some data that did not meet the requirements of the study, such as blank data, videos with insufficient length, and data that did not align with the interior and exterior views of the cockpit. Therefore, these data were manually screened. Table 3 displays the specific information of the Brain4Cars and Zenodo datasets. In-car indicates internal cockpit data after excluding the unqualified data. Out-car indicates the external cockpit data after excluding the unqualified data. In-Out indicates the combined data inside and outside the cockpit after excluding the unqualified data.

Table 3. Specific information for the Brain4Cars and Zenodo datasets.

<table>
<thead>
<tr>
<th>Manoever</th>
<th>Brain4Cars</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Type</td>
<td>Straight</td>
<td>Lchange</td>
<td>Lturn</td>
<td>Rchange</td>
</tr>
<tr>
<td></td>
<td>In-car</td>
<td>234</td>
<td>124</td>
<td>58</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>Out-car</td>
<td>254</td>
<td>125</td>
<td>67</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>In-Out</td>
<td>215</td>
<td>103</td>
<td>49</td>
<td>99</td>
</tr>
<tr>
<td>Zenodo</td>
<td>In-car</td>
<td>32</td>
<td>24</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Out-car</td>
<td>32</td>
<td>21</td>
<td>24</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>In-Out</td>
<td>32</td>
<td>21</td>
<td>24</td>
<td>17</td>
</tr>
</tbody>
</table>

4.2. Implementation

In this study, our method was based on the PyTorch framework and was used to conduct driver steering intention recognition studies within the cockpit, outside the cockpit, and jointly within and outside the cockpit on a server equipped with an NVIDIA GeForce RTX 3060 (Santa Clara, CA, USA). This study used fivefold cross-validation and different training strategies for different experiments:

- In-cockpit experiments: The model was trained using 60 epochs and a batch size of 12. To ensure robustness and performance, a cross-entropy loss function was utilized, with stochastic gradient descent (SGD) as the optimizer. The momentum was set to 0.9, and the weight decay to 0.001. Additionally, a multistep learning rate scheduler was implemented, starting with an initial learning rate of 0.01. The learning rate was adjusted with a decay rate of 0.1 after the 30th and 50th epochs.

- Out-of-cockpit experiments: The model was trained using 80 epochs and a batch size of 8. The initial learning rate was set to 0.01 and decayed by a rate of 0.1 after the 30th and 60th epochs. The mean square error loss function was employed, while the remaining parameters remained consistent with those used in the in-cockpit experiments.

- Joint in-cockpit and out-of-cockpit experiments: the training strategy was the same as for the out-of-cockpit experiment.

4.3. Evaluation Metrics

In this study, the model performance was evaluated based on the accuracy, F1-score, and number of model parameters. The number of model parameters refers to the total count
of weights and biases that need to be learned in the network, indicating the complexity of the model. A lower number of parameters suggests that the model is more lightweight and efficient, making it more suitable for use in resource-limited settings.

4.4. Results and Discussion

This study focused on evaluating the model through in-cockpit experiments, out-of-cockpit experiments, and joint in-cockpit and out-of-cockpit experiments. Since this study utilized two datasets in the training process of the models, the Brain4Cars dataset was abbreviated as B and the Zenodo dataset was abbreviated as Z for the sake of concise expression.

4.4.1. In-Cockpit Experiments

In the cockpit interior experiment, the performance of 3DResNet 50 was compared with A-3DResNet 34, A-3DResNet 50, and A-3DResNet 101 on the Brain4Cars dataset, and the results are shown in Table 4.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc (%)</th>
<th>F1 (%)</th>
<th>Param (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DResNet 50 [11]</td>
<td>77.4</td>
<td>75.5</td>
<td>46.22</td>
</tr>
<tr>
<td>A-3DResNet 34 (ours)</td>
<td>75.3</td>
<td>76.0</td>
<td>33.15</td>
</tr>
<tr>
<td>A-3DResNet 50 (ours)</td>
<td>79.5</td>
<td>81.6</td>
<td>46.20</td>
</tr>
<tr>
<td>A-3DResNet 101 (ours)</td>
<td>79.1</td>
<td>79.6</td>
<td>55.26</td>
</tr>
</tbody>
</table>

Bold numbers indicate best performance.

Based on the evaluation results of each model on the Brain4Cars dataset, the A-3DResNet 50 achieved the highest accuracy of 79.5% and an F1-score of 81.6%, with a parameter count of 46.20 M. In contrast, the A-3DResNet 34 is a shallower network with fewer layers and parameters, having a parameter count of only 33.15 M. This design made the model faster in both training and inference, but it also resulted in a poorer performance. The accuracy of A-3DResNet 101 was 0.4% lower than that of A-3DResNet 50, and the F1-score was 2.0% lower than that of A-3DResNet 50. This was due to its deeper network structure and higher number of parameters, but this also led to longer model training and inference time, which ultimately led to lower performance.

Based on the evaluation results of 3DResNet 50 and A-3DResNet 50 on the Brain4Cars dataset, it was evident that A-3DResNet 50 outperformed 3DResNet 50. A-3DResNet 50 showed improvements of 2.1% and 6.1% in accuracy and F1-score, respectively, while reducing the number of parameters by 0.02 M. The enhanced performance could be attributed to the atrous convolution introduced in the initial stage of 3DResNet 50. This was because atrous convolution broadened the receptive field of the input data, capturing more detailed feature information. The atrous convolution minimized the amount of parameters that needed to be learned by spanning the correlation between nearby pixels to extract information.

4.4.2. Out-of-Cockpit Experiments

In the cockpit exterior experiments, ablation experiments were performed on the Brain4Cars dataset for both ConvLSTM and the DSMax module, and ConvLSTM-4DSMax was compared with other methods; the results are shown in Table 5.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc (%)</th>
<th>F1 (%)</th>
<th>Param (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-3DResNet 50</td>
<td>77.3</td>
<td>79.5</td>
<td>46.20</td>
</tr>
<tr>
<td>A-3DResNet 101 (ours)</td>
<td>79.1</td>
<td>79.6</td>
<td>55.26</td>
</tr>
</tbody>
</table>

Based on the evaluation results of each model on the Brain4Cars dataset, it was evident that A-3DResNet 50 outperformed 3DResNet 50. A-3DResNet 50 showed improvements of 2.1% and 6.1% in accuracy and F1-score, respectively, while reducing the number of parameters by 0.02 M. The enhanced performance could be attributed to the atrous convolution introduced in the initial stage of 3DResNet 50. This was because atrous convolution broadened the receptive field of the input data, capturing more detailed feature information. The atrous convolution minimized the amount of parameters that needed to be learned by spanning the correlation between nearby pixels to extract information.
Table 5. Experimental evaluation of out-of-cockpit experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc (%)</th>
<th>F1 (%)</th>
<th>Param (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rong [11]</td>
<td>60.9</td>
<td>66.4</td>
<td>5.41</td>
</tr>
<tr>
<td>Gebert [12]</td>
<td>53.2</td>
<td>43.4</td>
<td>240.26</td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>50.3</td>
<td>53.6</td>
<td>1.45</td>
</tr>
<tr>
<td>ConvLSTM-DSMax (ours)</td>
<td>55.6</td>
<td>58.1</td>
<td>2.43</td>
</tr>
<tr>
<td>ConvLSTM-2DSMax (ours)</td>
<td>57.8</td>
<td>59.9</td>
<td>3.40</td>
</tr>
<tr>
<td>ConvLSTM-3DSMax (ours)</td>
<td>60.1</td>
<td>62.7</td>
<td>4.38</td>
</tr>
<tr>
<td>ConvLSTM-4DSMax (ours)</td>
<td><strong>63.9</strong></td>
<td><strong>66.5</strong></td>
<td><strong>5.35</strong></td>
</tr>
<tr>
<td>ConvLSTM-5DSMax (ours)</td>
<td>63.2</td>
<td>65.7</td>
<td>6.33</td>
</tr>
</tbody>
</table>

Bold numbers indicate best performance.

Based on the comparison of ConvLSTM-4DSMax with Rong [11] and Gebert [12] on the Brain4Cars dataset, it can be seen that ConvLSTM-4DSMax had the best results. It achieved an accuracy of 63.9\%, an F1-score of 66.5\%, and a parameter count of only 5.35 M. This was due to the DSMax module in ConvLSTM-4DSMax, which included depthwise separable convolution and max pooling. Depthwise separable convolution can reduce the number of parameters and computational complexity of the model. Max pooling is a spatial dimensionality reduction strategy that can decrease the dimensionality of the feature map while preserving crucial features, thereby enhancing the efficiency of subsequent processing.

4.4.3. Joint In-Cockpit and Out-of-Cockpit Experiments

In EDNet, A-3DResNet 50 extracts driver intention features inside the cockpit, and ConvLSTM-4DSMax extracts road environment features outside the cockpit. Then, the two features are integrated using the D-CAF module, and the freeze-training strategy is applied to A-3DResNet 50 and ConvLSTM. To evaluate the impact of the D-CAF module and freeze-training strategy on the model performance, this study conducted experiments using four different combinations on both the Brain4Cars dataset and the Zenodo dataset. The results of the evaluations are presented in Table 6.

- Splicing: features inside and outside the cockpit were concatenated column-wise.
- Splicing + freeze: features inside and outside the cockpit were concatenated column-wise and the freeze-training strategy was used.
- D-CAF: D-CAF feature fusion method.
- D-CAF + freeze: D-CAF feature fusion method and freeze-training strategy.

Table 6. Performance of D-CAF module and freeze-training strategy on Brain4Cars and Zenodo datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc (%)</th>
<th>F1 (%)</th>
<th>Param (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Splicing</td>
<td>83.9</td>
<td>84.3</td>
<td>57.92</td>
</tr>
<tr>
<td>Splicing + freeze</td>
<td>84.4</td>
<td>85.2</td>
<td>7.80</td>
</tr>
<tr>
<td>D-CAF</td>
<td>85.3</td>
<td>86.5</td>
<td>59.63</td>
</tr>
<tr>
<td>D-CAF + freeze</td>
<td><strong>86.3</strong></td>
<td><strong>87.8</strong></td>
<td><strong>11.88</strong></td>
</tr>
</tbody>
</table>

Bold numbers indicate best performance.

From Table 6, it can be seen that after adding freezing to splicing and D-CAF, the number of parameters was significantly reduced, and the accuracy and F1-score were both improved to varying degrees. Replacing splicing with D-CAF resulted in a 1.4\% increase in accuracy, a 2.2\% increase in F1-score, and a 1.71 M increase in the number of parameters. This increase in parameters was necessary to capture and integrate the relationships between different features. The results show that D-CAF + freeze was the most effective, with 11.88 M parameters, which was only one-fifth that of D-CAF. Due to the utilization of the previous optimal weights, both the accuracy and F1-score improved, reaching 86.3\% and 87.8\%, respectively. This result demonstrates the effectiveness of the D-CAF module and the freezing strategy.
Figure 8 displays the confusion matrix, where each row represents the true label and each column represents the predicted category. The results show that the recognition of straight performed excellently, with an accuracy of 91.3%, while the recognition of Lchange had the lowest accuracy at 79.17%. As seen from the confusion matrix, 12.5% of the samples with a straight intention were misclassified as an Lchange intention. Observation of the dataset revealed that there were some ambiguous samples between straight and Lchange. There were some irregular behaviors of drivers in the straight data, such as drivers communicating with other people in the vehicle or being distracted by things outside the vehicle. These actions caused the driver’s head to rotate, leading the model to misclassify the straight intention as Lchange intention.

Table 7 displays the comparison with other methods. Camera denotes the video data oriented in and out of the cockpit. Other denotes the GPS information, head attitude information, vehicle speed, and lane number information.

Figure 8. Confusion matrix.

Table 7. Comparison with other methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Feature</th>
<th>Evaluate</th>
<th>Camera</th>
<th>Other</th>
<th>Acc (%)</th>
<th>FI (%)</th>
<th>Param (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO-HMM [6]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>72.7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>AIO-HMM [6]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>74.2</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>S-RNN [7]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>74.4</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>F-RNN-UL [7]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>78.9</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>F-RNN-EL [7]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>80.6</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>SDAE [8]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>73.3</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Deep CNN [8]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>77.7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>DBRNN [9]</td>
<td>*</td>
<td>✓</td>
<td>✓</td>
<td>80.0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Khairdoost [10]</td>
<td>*</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>83.5</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Rong [11]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>83.9</td>
<td>84.3</td>
<td>57.92</td>
<td>57.92</td>
<td>57.92</td>
</tr>
<tr>
<td>Gebert [12]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>75.5</td>
<td>73.2</td>
<td>170.52</td>
<td>170.52</td>
<td>170.52</td>
</tr>
<tr>
<td>CEMF_CC [13]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>77.5</td>
<td>80.4</td>
<td>87.30</td>
<td>87.30</td>
<td>87.30</td>
</tr>
<tr>
<td>TIFN [14]</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>87.9</td>
<td>12.30</td>
<td>12.30</td>
<td>12.30</td>
</tr>
<tr>
<td>EDNet (ours)</td>
<td>B</td>
<td>✓</td>
<td>✓</td>
<td>85.5</td>
<td>86.2</td>
<td>11.88</td>
<td>11.88</td>
<td>11.88</td>
</tr>
<tr>
<td>EDNet (ours)</td>
<td>Z</td>
<td>✓</td>
<td>✓</td>
<td>86.6</td>
<td>89.0</td>
<td>11.88</td>
<td>11.88</td>
<td>11.88</td>
</tr>
<tr>
<td>EDNet (ours)</td>
<td>B + Z</td>
<td>✓</td>
<td>✓</td>
<td>86.3</td>
<td>87.8</td>
<td>11.88</td>
<td>11.88</td>
<td>11.88</td>
</tr>
</tbody>
</table>

* Evaluation based on self-built dataset. Bold numbers indicate the best performances on the mixed dataset. Italicized numbers indicate the best performances on the Brain4Cars dataset.
It can be seen that the EDNet worked best on the Zenodo dataset, with an accuracy of 86.6%, an F1-score of 89.0%, and a parameter count of 11.88 M. This was due to the better data quality of the Zenodo dataset, which originated from a laboratory simulator with high video quality and clearer video information. In addition, drivers in the Zenodo dataset showed a greater range of movements and more regular driving behavior. On the Brain4Cars dataset, our model outperformed Rong [11], Gebert [12], and CEMF_CC [13]. On the mixed dataset, our model achieved an accuracy of 86.3% and an F1-score of 87.8%, further demonstrating the effectiveness of our model.

As shown in Table 7, the F1-score of the EDNet was smaller than TIFN [14] on the Brain4Cars dataset. Therefore, in order to compare the two methods in detail, the advanced prediction ability of the different methods on the Brain4cars dataset is presented in Table 8. The advanced prediction ability of the model was evaluated by inputting varying numbers of frames into the model. The standard video length in the dataset was 5 s. Ending moments of the video represented the end of the driver’s steering intention generation process. We use 0 to denote the end moment of the driver’s steering intention generation process, and thus, $-5$ denotes the beginning moment of the driver’s steering intention generation process. Therefore, we use $[-5, 0]$, $[-5, -1]$, $[-5, -2]$, $[-5, -3]$, and $[-5, -4]$ to denote the prediction times of the model. $[-5, 0]$ indicates that the input to the model was 5 s of video data. $[-5, -1]$ indicates that the input to the model was 4 s of video data. $[-5, -2]$ indicates that the input to the model was 3 s of video data. $[-5, -3]$ indicates that the input to the model was 2 s of video data. $[-5, -4]$ indicates that the input to the model was 1 s of video data. The results presented in Table 8 demonstrate that our EDNet outperformed Rong [11] in terms of accuracy across all time ranges. In the comparison of the F1-scores, our EDNet outperformed other methods except $[-5, 0]$. As the prediction time range increased, both the accuracy and F1-scores gradually increased. It can be observed that there was a significant correlation between the time range and the accuracy, as well as the F1-score of the prediction. With a larger time range, the model could learn more information, resulting in improved recognition of the driver’s steering intention.

In addition, the accuracy and F1-scores of these three methods were low for $[-5, -3]$ and $[-5, -4]$. This was due to the fact that at the beginning of the video, the driver kept driving straight ahead, which made the distinction challenging. Notably, our method showed a significant improvement from $[-5, -3]$ to $[-5, -2]$, with both the accuracy and F1-score increasing by 7.9% and 10.2%, respectively. The accuracy reached 73.7%, and the F1-score reached 73.0%, indicating that our method could identify the driver’s steering intention 3 s in advance.

### 5. Conclusions

In this paper, we propose a novel end-to-end dual-branch network (EDNet). EDNet uses A-3DResNet 50 as the driver intent feature extractor inside the cockpit, combines ConvLSTM and the depthwise-separable max-pooling (DSMax) module as the road environment feature extractor outside the cockpit, and then integrates the two types of features
based on the dynamic combined-feature attention fusion (D-CAF) module. During the training of the networks, a freeze-training strategy is used for A-3DResNet and ConvLSTM. The experiments show that combining information from inside and outside the cockpit improved the driver steering prediction. Our method achieved impressive results: 85.6% accuracy and 86.2% F1-scores on the Brain4Cars dataset, 86.6% accuracy and 89.0% F1-score on the Zenodo dataset, and 86.3% accuracy and 87.8% F1-score on a combination of the two datasets, with a model parameter count of 11.88 M. The results show that the comprehensive performance of our method outperformed other methods in its class.

Despite the significant results achieved by the driver intention recognition method proposed in this paper, it still has some limitations. There is a relative lack of research data, with little publicly available video data recorded both inside and outside the cockpit. Additionally, real driving scenarios frequently involve improper behaviors, like driver distraction and conversing with others, which can confuse the recognition of the driver’s steering intention. Therefore, to better adapt the model to real driving scenarios and enhance its application in vehicle driving safety, we plan to collect data in future studies that are more representative of real driving environments, including driving behaviors such as driver distraction, conversation, and eating. We will analyze this data and refine the model to effectively distinguish these improper behaviors. We will also enhance the model’s accuracy in detecting drivers’ lane changing and turning intentions to meet the needs of real-world applications, and we are committed to making the model more lightweight for practical deployment. With these enhancements, driver behavior will be standardized, which is expected to reduce traffic accidents caused by driver error, including those resulting from improper driving behavior and misinterpretation of road conditions. Moreover, these improvements will be highly beneficial for autonomous vehicles, significantly enhancing their adaptability and driving safety.

Author Contributions: Conceptualization, Y.F.; funding acquisition, Y.F.; investigation, Y.F. and H.X.; methodology, Y.F. and H.X.; project administration, Y.F.; software, H.X.; supervision, J.F.; validation, H.X.; visualization, H.X.; writing—original draft, H.X.; writing—review and editing, Y.F., H.X., J.F. and Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Chongqing Basic Research and Frontier Exploration Project (Chongqing Natural Science Foundation) (CSTB2022NSCQ-MSX0918), Science and Technology Research Project of Chongqing Education Commission (Youth) (KJQN202101149), and Chongqing University of Technology Graduate Education Hight-Quality Development Action Plan Funding Results (gjcz20243183).

Data Availability Statement: The relevant algorithm of this study is available at https://github.com/XueIs100/EDNet, accessed on 15 May 2024.

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

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.