International Journal of Transportation Engineering and Technology
2023; 9(1): 19-26
http://www.sciencepublishinggroup.com/j/ijtet
doi: 10.11648/j.ijtet.20230901.13
ISSN: 2575-1743 (Print); ISSN: 2575-1751 (Online)

Driver Decision Space Inversion Method Based on DC-GAN

Shuanfeng Zhao*, Leping Li, Yang Li, Mengwei Wang, Shuaijun Wang, Mingyue Li

College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an, China

Email address:

zsf@xust.edu.cn (Shuanfeng Zhao), 1693522484@qq.com (Leping Li), 1123649987@qq.com (Yang Li), 529933419@qq.com (Mengwei Wang), 1517827981@qq (Shuaijun Wang), 1040914973@qq.com (Mingyue Li)

*Corresponding author

To cite this article:

Shuanfeng Zhao, Leping Li, Yang Li, Mengwei Wang, Shuaijun Wang, Mingyue Li. Driver Decision Space Inversion Method Based on DC-GAN. *International Journal of Transportation Engineering and Technology*. Vol. 9, No. 1, 2023, pp. 19-26.
doi: 10.11648/j.ijtet.20230901.13

Received: February 27, 2023; **Accepted**: April 11, 2023; **Published**: April 15, 2023

Abstract: In the driving process, drivers need to constantly perceive the surrounding environment to make decisions and perform operations. This cognitive space formed in the driver's mind is the driver's decision space. For the actual driving environment, factors such as complex road sign settings, unreasonable road planning, long-time fatigue driving, and reduced reaction ability of elderly drivers may interfere with the normal perception of the driver's decision space, resulting in reduced driving safety. In this paper, a driver decision space inversion method based on deep convolutional neural networks and generative adversarial networks is proposed to study driver perception in near-domain traffic scenarios. The steps of the method include: near-domain target element extraction, driving data collection, sample data generation, adversarial generative network model learning, data enhancement and decision space inversion. The experimental results show that the method in this paper can accurately identify the driver's decision space in both real and simulated driving scenarios by implementing real-time monitoring of driver perception. The method is important for studying the driver's decision space, which can promote the development of intelligent driving technology and the synergistic development of human-vehicle-road-loop. In the context of today's sustainable transportation and smart mobility, driver decision space research is not only an important basic research, but also an inevitable requirement to promote innovation and upgrading in the transportation field.

Keywords: Deep Convolutional Neural Networks, Generative Adversarial Networks, Driver Perception, Decision Space Inversion

1. Introduction

Driving is a complex human-computer interaction process in which the driver needs to constantly perceive the surrounding environment, make decisions and perform operations [1, 14]. The decision space is a cognitive space formed in the driver's mind, and the driver constantly adjusts and updates the decision space based on the perceived environmental information [15]. Due to the "black box" effect of the human brain, this paper cannot directly observe the process of change in the driver's decision space. However, by observing and analyzing the driver's operating behavior, this paper can infer changes in the driver's decision space [2, 13]. Therefore, in-depth research on the rules and characteristics of driver decision space changes is of great significance for optimizing driving environment and improving driving safety and efficiency. In the actual driving environment, factors such as complex road sign setting, unreasonable road planning, long-time tired driving, elderly drivers' reaction ability decline and other factors may interfere with drivers' normal perception of decision space, leading to reduced driving safety [3, 4, 11]. Therefore, this paper needs to study and improve these factors in order to achieve a positive interaction between drivers and driving environment. At the same time, by visualizing the changes of driver's decision space, this paper can better understand and master driver's decision process, help driver better adapt to different driving environments, and improve driving safety and efficiency.

Therefore, it is very important to study the evolution process of driver's decision space and the influencing factors for its application [5, 16]. Municipal authorities can use this to identify the rationality of road planning and the safety of traffic guidance Settings; Insurance companies can use this to analyze the cause of the accident and determine liability; The assisted driving system can evaluate the driving behavior by this method; The unmanned driving system can also refer to the driver perception process for human-like driving and so on. This is very significant to promote the progress of intelligent driving and the development of human-vehicle-road-loop synergy. In the context of sustainable transportation and smart travel, the research on driver decision space is not only an important basic research, but also an inevitable requirement for promoting innovation and upgrading in the field of transportation [5, 6, 12].

This paper presents a method of driver decision space inversion. In this method, near-area target elements, such as vehicles, roads, pedestrians and traffic lights, are extracted by inputting aerial view sequence of real environment, and these elements are coded. Then, the driving data acquisition module collects vehicle driving data such as steering wheel angle, brake and accelerator pedal at the moments corresponding to the bird's eye view sequence. The coded data of target elements and driving data are combined to form the sample data. The target elements extracted from the aerial view map correspond to the decision-making space of the driver, and the driving data correspond to the decision-making of the driver. These sample data will be input into the adversarial generation network model, and the network will perform data enhancement according to the characteristics of the input sample data. Each generated data is a combination of driving behavior and element coding, so for each generated data, this paper can launch a bird's eye view corresponding to it by driving data. The bird's-eye view of the generated driving data that is most similar to the input driving data is resolved as the driver's decision space in the current state. With the continuous input of real scene bird's eye image frames, this paper is able to observe how the driver's decision space changes with the environment.

Generating adversarial networks has an important role in the approach to driver decision space inversion [7]. It can learn features from sample data and enhance data from input samples to generate more realistic data [8]. In addition, by analyzing the generated data, this paper can infer the driver's decision space and observe how it changes as the environment changes.

2. Method

The extraction of the driver decision space needs to be based on a sequence of input aerial views of the driver's near domain environment, and the information about the target elements present in the driver's near domain in each frame is extracted through the input of successive aerial views of the driving scene. Since the bird's-eye view reflects all the information around the driver in the current driving state, which includes vehicles, pedestrians, traffic signs, etc. Therefore, this paper use a target detection and tracking network to extract the target elements present in the bird's eye view sequence (This part of work has been completed in article previous research content). The extracted target element sequence and driving data are combined into sample data, which is input into the generated antagonistic network for training.

Since it is impossible to include all the driving scenarios that a driver would encounter in a realistically collected driving scenario and a simulation-generated driving scenario, this paper use a generative adversarial network for data augmentation [9]. The generative adversarial network consists of a generator network and a discriminator network [9]. This paper improve the generator network by using an encoder-decoder network as the basis and introducing the attention mechanism module [10] and the residual module to allow the generator network to generate data selectively and prevent the model from crashing due to the generation of a large amount of irrelevant data. The improved generative adversarial network model learns based on the features of the input sample data to generate many realistic data. Because each set of sample data is composed of both the target element data extracted from the bird's eye view and the driving data, the network generates many data that also generate many driving scenarios.

Finally, this paper compare the similarity between the input driving data and the driving data part of the database by comparing the error analysis, and the bird's-eye view sequence corresponding to the data sequence with the smallest error with the input data is the driver's decision space sequence.

Figure 1 shows the framework of driver decision space inversion method.

2.1. Driver Decision Space Generation

Convolutional neural network (CNN) is a deep learning method, it has the characteristics of local perception domain, shared weight and space, which can not only reduce data connection parameters, but also optimize the network by using the local features hidden in the data to maintain a certain degree of deformation, such as displacement, scaling, rotation, etc. Generally, the convolutional neural network consists of alternating convolution layer and pooling layer, and then a fully connected layer is added to the output layer to form a complete convolutional neural network. The sample data in this paper is composed of multi-dimensional arrays, which can be transformed into a series of feature graphs by the convolution layer and the pooling layer. The pooling is for the purpose of feature extraction and feature dimension reduction.

The architecture of the generative adversarial network consists of a generator G and a discriminator D. The architecture of the generative adversarial network is shown in Figure 2. The generator generates dummy samples G(z) by inputting a random vector z, The purpose of the generator is to generate fake sample data that can be faked, and the discriminator makes a distinction between the real data and the generated fake data. The purpose of the discriminator is to better distinguish the real data from the generated fake data. G and D are usually nonlinear mapping functions, such as convolutional neural networks. The structure of the generated adversarial network is shown in the following figure.

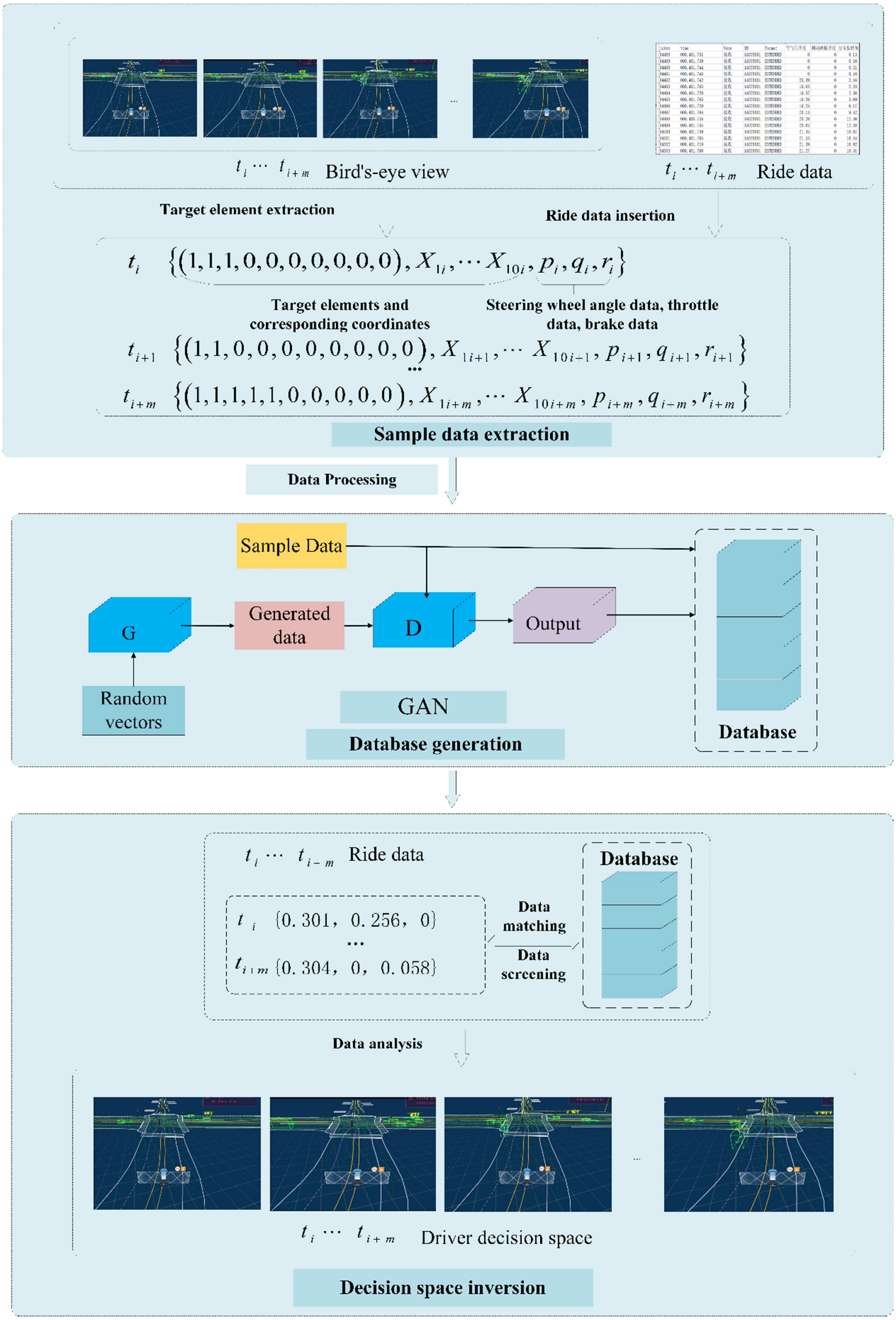

***Figure 1.** Driver decision space inversion model framework.*

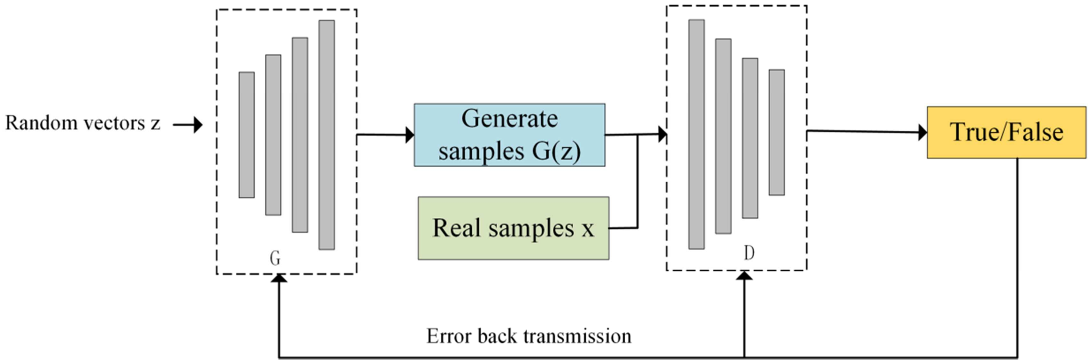

Figure 2. *Generating adversarial network framework.*

Based on the powerful data generation capability of generative adversarial networks, this paper uses a deep convolutional neural network-based generative adversarial network (DC-GAN) to generate a database of decision space samples of drivers. Then, by comparing the similarity between the driving data series and the driving data series in the sample database, the decision space of the driver is reconstructed.

2.1.1. Network Improvement

The generated network in DC-GAN is an encoder-decoder structure. The network structure of the generator is designed, which is composed of convolutional block, deconvolution block and attention module. The main improvements are to introduce jump connections, attention mechanisms, and Dropout layers into the generator network structure to redesign the generator structure.

The convolutional block is responsible for extracting sequences in the input data and consists of a convolutional layer, a normalization layer and an activation layer. The principle of deconvolution block is opposite to that of convolutional block. Deconvolution block is used in generator for original size reduction, and its composition includes deconvolution, normalization layer and activation layer.

The focus of attention block is to set different weights for different features, and to focus on strongly correlated information while suppressing weakly correlated information. It consists of global average pooling layer, convolutional layer, ReLu activation layer and sigmoid activation layer. Figure 3 shows the structure of the attention mechanism, where the input feature map is first subjected to a global maximum pooling operation, and then the features are processed into 1×1×C features by the fully connected layer, Rule activation into, and fully connected layer. The weights are assigned for different features by setting threshold parameters. The Sigmoid activation layer is set to increase the nonlinearity of the model. The weights of each channel are represented by a real number fixed in the interval from 0 to 1. Finally, the weights of each channel are multiplied with the channel counterparts of the original feature map.

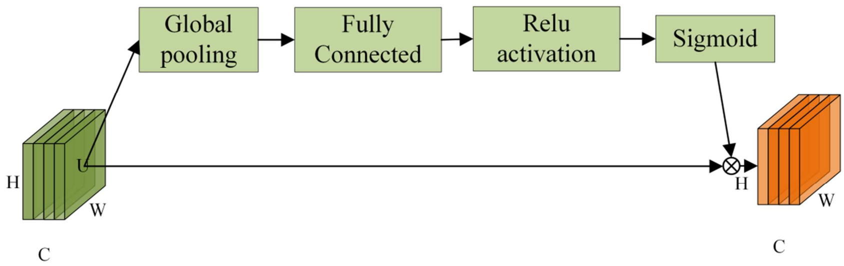

Figure 3. *Attention Module Structure.*

$$S_c = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} v_c(i, j) \quad (1)$$

$$e = \sigma(W_2 \delta(W_1 S)) \quad (2)$$

$$M_c = e_c M_c \quad (3)$$

In equations (1) to (3): S_c is the channel feature element of feature map C, $v_c(i, j)$ is the element in the C channel whose row and column coordinates are at positions i, j respectively. The width and height of the input C channel feature map is denoted by w, h. S is the output vector after global pooling, and the weight vector e is output after full connection W and two activation functions Rule and Sigmoid. Here, δ represents the Rule activation function, and σ represents the Sigmoid activation function. Finally, the feature vector $\widetilde{M}_c$ containing attention weight is calculated by weighting the weight vector sorted according to threshold segmentation and the input feature graph matrix M_c.

Jump connections are introduced to increase the nonlinear fit of the model by nonlinear superposition between the convolutional layers via the RULE nonlinear activation function. Neural networks lacking nonlinear activation do not have the ability to learn diverse parameters, while with the addition of a nonlinear activation function, multiple nonlinear activation layers are applied to increase the nonlinear expression capability of the model, thus enabling the fitting of arbitrary relationships. The Rule activation function is shown in equation (4), where the function has an output of 0 for a negative input and an output of the input itself for a positive input.

$$f(x) = \max(0, x) \tag{4}$$

The principle of residual jump connection is shown in equation (5):

$$H(x) = F(x) + x \tag{5}$$

In equation (5), x for input, $H(x)$ means output, $F(x)$ represents the convolution and activation of the input x. The presence of jump connections allows the covariance unit to not only convolve the input, but also fuse the unprocessed input x, enhancing the transfer of data information to the back-layer network.

Adding Dropout layer can avoid causing data overfitting. Because neural networks may perform well on the training set but not on the test set when processing large amounts of data, adding a Dropout layer can effectively avoid this phenomenon. It is based on the principle of randomly turning neurons into 0,1 vectors and stopping the work of neurons that multiply with 0 during the learning process of neurons. The emergence of this mechanism, which randomly combines the neurons used by the neural network for training and learning, increases the learning ability of the model, making the process of neuron parameter transfer within the convolutional layer dynamically variable and increasing the generalization ability of the model.

2.1.2. Network Processes

The input features of the network are the extracted target element sequence and the coordinates corresponding to the target sequence, steering wheel angle data, accelerator pedal opening data, and brake pedal torque data. The target element series data can be expressed by equation (6).

$$N = (n_1, n_2, \cdots, n_{10}) \tag{6}$$

Where n represents the extracted target elements, it is composed of 0 or 1, 0 represents no target elements, 1 represents target elements, considering the complexity of road traffic and the generation capacity of the model, the maximum capacity of the extracted target elements is set to 10. Then this paper need to know the exact location of the target element so that this paper can get the exact distribution of the target element, the location of the target element is expressed by the following equation.

$$X = \{X_{1i}, X_{2i}, \cdots, X_{10i}\} \tag{7}$$

In equation (7), X_{1i} represents the position coordinate of the first target element at moment i, which is represented by (x_{1i}, y_{1i}).

Then, the target element sequence N, position information X, steering wheel Angle data p, acceleration pedal opening data q, brake pedal torque data r, together constitute the input characteristics.

$$M = \{N, X, p, q, r\} \tag{8}$$

Encoder network encoding input features, information fusion sequence M obtained after information embedding is a high dimensional vector of fixed length, and then encoded by the encoder to obtain the encoded hidden state information m. And then using the attention mechanism to assign different weights to the information, Assuming that the attention vector is e^t, it can represent the most important information in the input fusion information. Therefore, based on the attention mechanism, this paper can get the correlation matrix of information at each moment, and finally decode it by the decoder.

$$m = Encoder(M; W_{encoder}) \tag{9}$$

$$e^t = \sigma(W_2 \delta(W_1 m)) \tag{10}$$

$$M = Decoder(m, e^t; W_{decoder}) \tag{11}$$

Then the driver decision space generation model based on generative adversarial network generates the processed information through the generator network and discriminator network for game generation, which makes the generated data distribution closer and closer. The process of generating the adversarial network is shown in the following table 1.

Table 1. *Generate adversarial network training procedures.*

Generate adversarial network training procedures
1 if the generator G does not produce the required data:
2 Select the amount of sample data to be used for training batch size: m
3 for i in range k:
4 Sampling m samples from the real data distribution $P(x)$, $\{x_{(1)}, \cdots, x_{(m)}\}$
Sampling m samples from random input data distribution $P(z)$, $\{z_{(1)}, \cdots, z_{(m)}\}$
5 Optimize the function using gradient descent:
6 $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [-\log(D(x_i)) - \log(1 - D(G(z)))]$

7 Sampling m samples from random input data distribution $P(z)$, $\{z_{(1)},\cdots,z_{(m)}\}$
Optimize the function using gradient descent:
8 $\nabla_{\theta g}\frac{1}{m}\sum_{i=1}^{m}[\log(1-D(G(z)))]$
9 else:
10 The output generator G produces data that meets the requirements

2.2. Driver Decision Space Inversion

The sample data together with the data generated by the generative adversarial network form a sample database, and when a sequence of driver driving data is acquired, this paper can resolve the corresponding driver decision space based on the similarity between the driving data part and the sample data in the database. The inversion method is shown in the figure below, first calculate the error of the sequences in this data and all data sequences in the sample database respectively, the sequence with the smallest error resolves the bird's eye view corresponding to the driver's decision space. Because the scenarios parsed from a single data may not be unique, the driver decision space inversion method proposed in this paper only predicts the driver decision space over time for the driving data sequence.

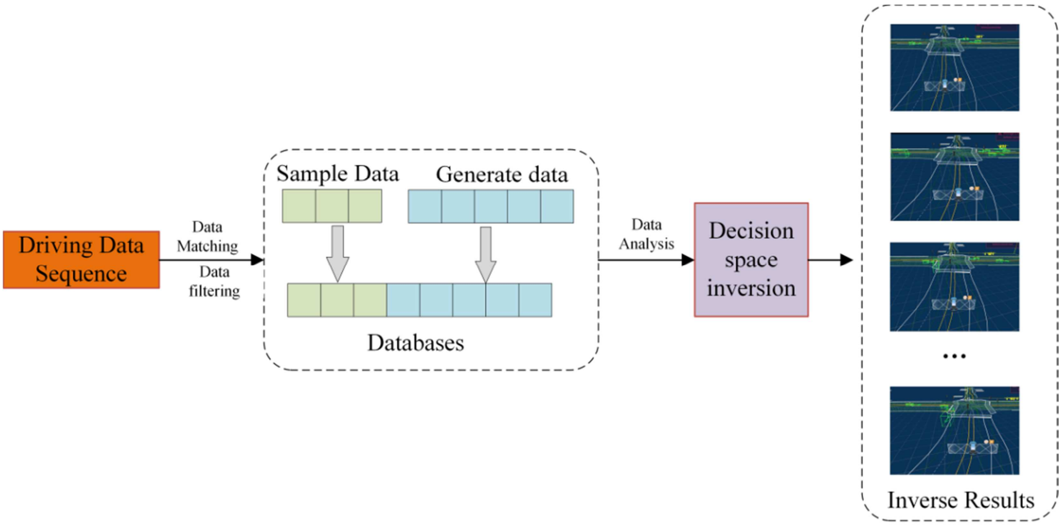

Figure 4. *Driver's decision space inversion method.*

3. Results and Discussion

This model was trained on a pytorch framework using a GTX3080 GPU and the Adam optimizer. Since generative adversarial networks need to be careful when setting the learning rate, otherwise the network is prone to crash. Therefore, this paper used the trial-and-error method to set the learning rate, and finally set the learning rate at 0.0001. The training results are as follows.

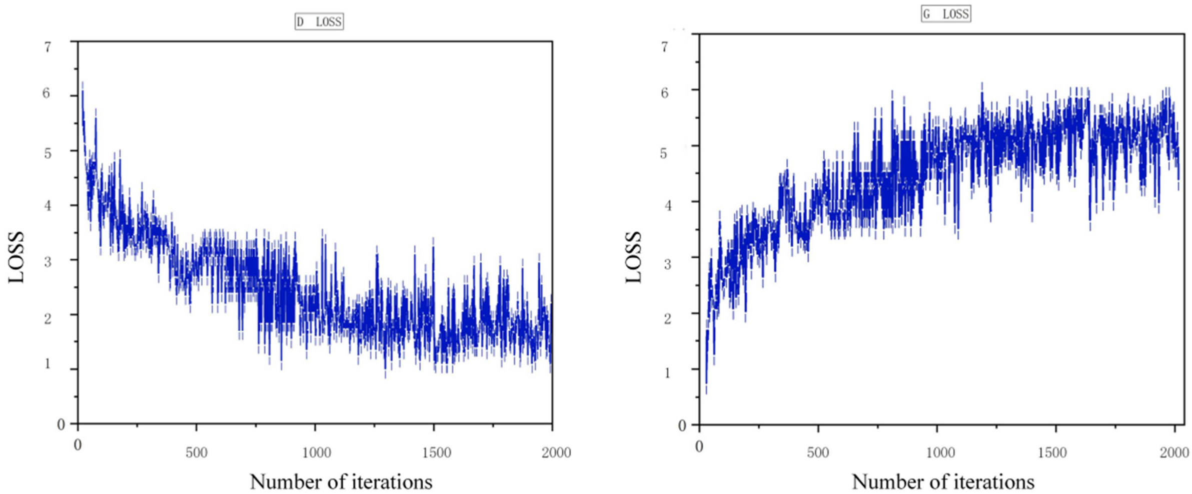

Figure 5. *Training loss graph.*

According to the previous analysis, this paper know that the value of $loss_D$ is obtained by summing the values of $loss_{D\_real}$ and $loss_{D-fake}$. Figures 5 represent the variation of discriminator loss function ($loss_D$) and generator loss function ($loss_G$) with increasing number of training on the dataset by DCGAN. From the figure, this paper can see that the generator and discriminator are relatively flat at the beginning of training, and as the number of training increases, the loss, loss value increases and the lossa loss value decreases, and the model gradually tends to stabilize and converge. From the figure, this paper can see that the generator and discriminator are relatively flat at the beginning of training, and as the number of training increases, the $loss_G$ value increases and the $loss_D$ value decreases, and the model gradually tends to stabilize and converge. The generator network and the discriminator network counterbalance each other, showing a large up and down fluctuation in the graph. The overall expression shows that the loss function of the discriminator gradually decreases and the loss function of the generator gradually increases.

Figure 6 shows the generation effect of driver decision space based on real driving scene and simulation driving scene. Several frames are randomly selected from the generated driver decision space sequence for display. In a real driving scenario, this paper label the aerial photographs taken by the UAV with information about the target elements in the decision space of the inverse performing driver. In figure a figure b, the driver can accurately perceive the vehicle in front of him and the traffic light at the intersection, and fails to perceive the red vehicle in the distance and the parked vehicle on the roadside. The reason for this analysis may be because the red vehicle is a roadside parked vehicle and is far away, which does not have an impact on the driver's decision, so this paper believe that the driver's decision space is generated accurately in this scenario. Again, the curb stop vehicle is not sensed in Figure c, but pedestrians and rear-end vehicles within the lane line are sensed. Figure d initially fails to detect the distant e-bike, but as the vehicle travels, this target element appears in the driver decision space.

In the simulated driving scenario, this paper drives the vehicle from the driver's perspective in the LGSVL simulator and inverts the driver's perception in the Dream3D bird's eye view. In the simulated driving scenario figure a, the driver perceives the vehicle during the approach of the rear side vehicle; in figure b, he slows down because he perceives the vehicle in front of him; in figure c, he perceives the vehicle in front of him on the right and chooses to drive away from the target; and in figure d, he chooses to accelerate straight ahead because he perceives the signal ahead. In summary, the driver decision space inversion method based on generative adversarial networks can accurately identify the driver's decision space in both real and simulated driving scenarios.

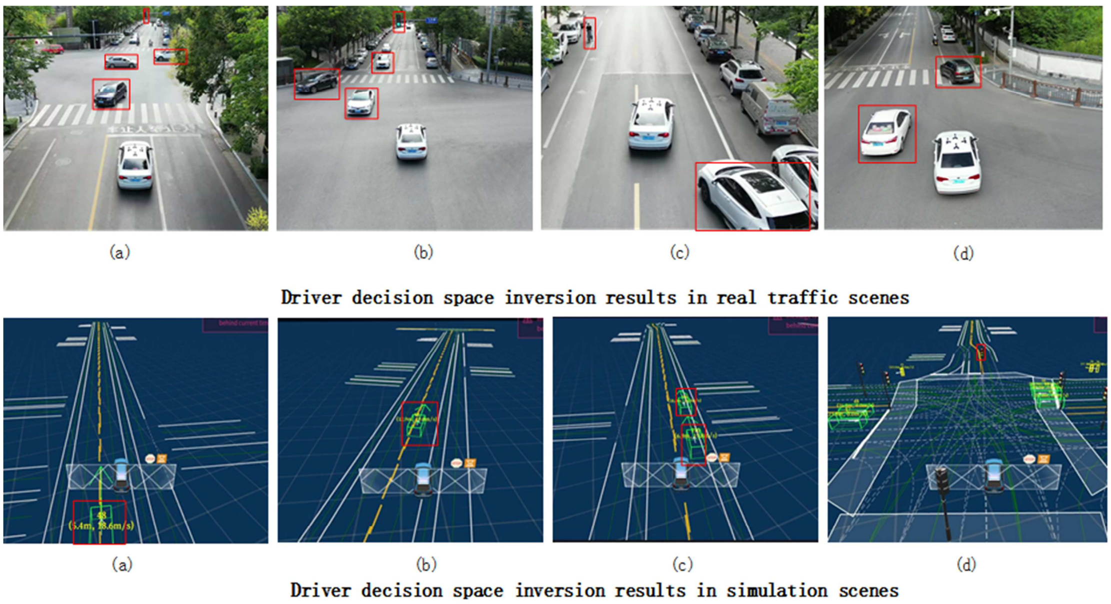

Figure 6. *Driver's decision space inversion results.*

4. Conclusion

In this paper, this paper proposes a DC-GAN method for inversion of driver decision space. The method first extracts the target elements of the driver's near domain as well as the driving data of the self-car, and after composing the sample data uses generative adversarial networks for data enhancement to generate a database of driver-perceived spatial scenes. Finally, this paper compare the input driving data sequences with the driving data sequences in the scene database to find out the sequence that is most similar to it, and the scene resolved by this sequence is the driver's decision space. This paper method can monitor driver perception in real time. In future work, this paper considers improving the generalization capability of the method and then investigates driver perception deficit and early warning methods based on the perception of driver decision space.

Funding

This work was supported by Shaanxi Provincial Key Research and Development Program Project No. 2020ZDLGY04-06, Shaanxi Provincial Key Research and Development Program Project No. 2020ZDLGY04-05.

References

[1] Zhang Zhi, Guo Yingshi, Yuan Wei, Wang Chang. The Impact of Cognitive Distraction on Driver Perception Response Time Under Different Levels of Situational Urgency [J]. IEEE Access, 2019, 7.

[2] Zang Jing, Jeon Myounghoon. The Effects of Transparency and Reliability of In-Vehicle Intelligent Agents on Driver Perception, Takeover Performance, Workload and Situation Awareness in Conditionally Automated Vehicles [J]. Multimodal Technologies and Interaction, 2022, 6 (9).

[3] Cheng Peiyao, Meng Fangang, Yao Jie, Wang Yiran. Driving With Agents: Investigating the Influences of Anthropomorphism Level and Physicality of Agents on Drivers' Perceived Control, Trust, and Driving Performance [J]. Frontiers in Psychology, 2022, 13.

[4] Riexinger Luke E, Fortenbaugh David M. A methodology for assessing driver perception-response time during unanticipated cross-centerline events. [J]. Traffic injury prevention, 2021, 22 (S1).

[5] Quanzhen Guan, Hong Bao, Zuxing Xuan. The research of prediction model on intelligent vehicle based on driver’s perception [J]. Cluster Computing, 2017, 20 (4).

[6] Zhanji Zheng, Zhigang Du, Qiaojun Xiang, Guojun Chen. Influence of multiscale visual information on driver’s perceived speed in highway tunnels [J]. Advances in Mechanical Engineering, 2018, 10 (12).

[7] Ding Hongwei, Sun Yu, Huang Nana, Shen Zhidong, Wang Zhenyu, Iftekhar Adnan, Cui Xiaohui. RVGAN-TL: A generative adversarial networks and transfer learning-based hybrid approach for imbalanced data classification [J]. Information Sciences, 2023, 629.

[8] Wang Shihong, Guo Jiayi, Zhang Yueting, Wu Yirong. Multi-baseline SAR 3D reconstruction of vehicle from very sparse aspects: A generative adversarial network based approach [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2023, 197.

[9] Ma Liling, Guo Jian, Li Jiehao, Wang Junzheng. A Noise-Excitation Generative Adversarial Network for Actuator Fault Diagnosis of Multi-legged Robot [J]. Unmanned Systems, 2023, 11 (02).

[10] Zhang Yucun, Li Tao, Li Qun, Fu Xianbin, Kong Tao. Image motion deblurring via attention generative adversarial network [J]. Computers & Graphics, 2023, 111.

[11] Walker G, Mendes LMN, Lenne M, et al. Modelling driver decision-making at railway level crossings using the abstraction decomposition space. Cognition, Technology & Work. 2021; 23 (2).

[12] Zhao N, Wang B, Xiong Y, Su R. Safety Prompt Advanced Driver-Assistance System with Lane-Change Prediction and Free Space Detection. 2022 IEEE 17th International Conference on Control & Automation (ICCA), Control & Automation (ICCA), 2022 IEEE 17th International Conference on. June 2022: 734-739.

[13] Kong H, Fang Y. The Car-Following Model Based on the Drivers’ Psychological Characteristics. International Journal of Intelligent Transportation Systems Research. June 2022: 1-9.

[14] Khaliq A, der Waerden P van, Janssens D, Wets G. A Conceptual Framework for Forecasting Car Driver’s On-Street Parking Decisions. Transportation Research Procedia. 2019; 37: 131-138.

[15] Bennajeh A, Bechikh S, Said LB, Aknine S. Bi-level Decision-making Modeling for an Autonomous Driver Agent: Application in the Car-following Driving Behavior. Applied artificial intelligence. 2019; 33 (13): 1157-1178.

[16] Liwei Hu, Zheng Chen, Ting Zhang, et al. Driver’s Dynamic Decision Car-following Model and Speed Limit in Plateau Extra-long Tunnel. Journal of Transportation Systems Engineering and Information Technology. 2018; 18 (5): 211-240.