Convolutional Neural Network Behavioral Cloning in Self-Driving

Haolong Fu, Jin Zhao and Axing Xi

May 1, 2019

Convolutional Neural Network Behavioral Cloning in Self-Driving

FU Haolong, ZHAO Jin, XI A'Xing, LIU Dongjie, LIU Zihao

Department of Mechanical Engineering, Guizhou University, Guiyang 550025, China

Abstract: To reduce the cost of the intelligent driving technology of intelligent car in a specific environment. An unmanned system with behavior cloning method is designed, which based on convolutional neural network. This method is a process that mimics human learning, unmanned driving by predicting the state of the vehicle through convolutional neural networks, ordinary visual sensors are used to obtain environmental information. First, behavioral cloning method based on convolutional neural network is studied. Then, an intelligent vehicle based on Linux system Raspberry Pi is designed, two sets of experiments are designed to verify the feasibility of the method. Finally, the desired model is applied to the first "DIY Robocars KuaiKai" Race Categories. The results show that the behavioral cloning method can realize the self-driving technology of intelligent car in a specific environment.

Keyword: Behavioral Cloning, Convolutional Neural Network, Self-driving, Intelligent vehicle, Raspberry Pi, Vision sensor

0 Introduction

With the development of artificial intelligence, Self-driving cars technology has been favored by more and more universities and enterprises at home and abroad. Due to the complexity of the unknown environment and the high cost, Self-driving is difficult to popularize in urban environments. However, Self-driving can be achieved in some specific environments. such as self-driving of BRT in urban environments, self-monitoring in border-specific patrol environments, etc.

The traditional self-driving system mainly consists of four parts, as shown in Figure 1 (a): acquisition of environmental information, high-precision map establishment, path planning, path tracking controller [1-6]. The traditional method usually separate these parts to study. However, if there is a part of the error, it will accumulate the error, resulting in an inaccurate final results; in addition, the traditional self-driving cars is relatively expensive.

In order to avoid the drawbacks of the traditional methods, Chen S et al[7] studied the cognitive map-based model, as shown in Figure 1 (b), through the neural network to obtain the cognitive map of the location and the relationship of objects in complex traffic scenes. This method verifies the potential of self-driving based on an end-to-end learning. Rausch V et al[8] studied a deep neural network strategy for end-to-end control of self-driving car, and compared the effects of three optimizations on convolution neural network loss by simulation.

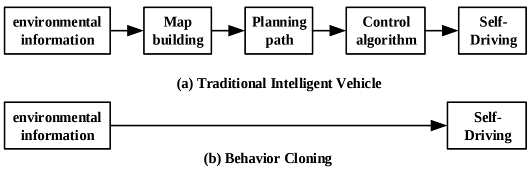


Figure 1 Comparison of traditional methods and behavioral cloning methods

Inspired by the method of end-to-end learning, this paper proposes a convolutional neural network of behavior cloning method to realize self-driving in a specific environment. In this environment, the vehicle surrounding is simple and fixed, and it recognizes the prescribed lane line and realizes self-driving by the behavioral cloning. Its advantages are: (1) reducing the accumulate error of traditional method; (2) convolutional neural network training model is self-optimization; (3) using visual sensors to obtain environmental information which saves costs greatly.

1 The Convolutional Neural Network Of Behavior Cloning

1.1 Behavior Cloning

Behavioral cloning is a method of machine learning. In this paper, the theory of behavioral cloning is used three parts to realize self-driving, as shown in Figure 2. The first part is the data acquisition. The operator controls the vehicle to obtain environmental information. Vehicle's status and environmental information will be recorded. The second part is training model, which uses the convolutional neural network to train the model. This stage is to learn the lane line information of environment image, “remembering” its speed and rotation angle at a certain position; The third part is feedback testing, which is also called the cloning stage. After the second part, the vehicle “remembered” its speed and corner at a certain moment. When the vehicle reaches the lane line next time, using the model to predict its speed and corner. thereby achieve the behavior cloning and complet the self-driving of the intelligent car in a specific environment.

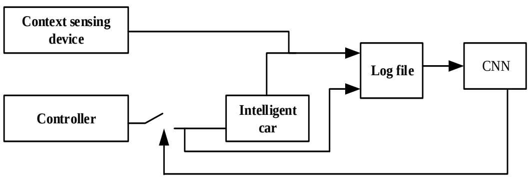


Figure 2 Behavioral cloning

1.2 Construction of the convolutional neural network behavioral cloning model

The Convolutional Neural Network (CNN) [9] mainly solves classification and regression problems. The typical convolutional neural network mainly includes input layer, convolution layer, and output layer.

In order to study the feasibility of the convolutional neural network behavior cloning, the prediction of vehicle speed and corner information is realized by convolutional neural network. We established a convolutional neural network model. It consists of three parts: model building, model training and model testing, as shown in Table 1.

Table1 Structure of Convolutional Neural Networks

Model establishment	Model training			Model test
	Forward propagation	Backward propagation	Model update	

2 Intelligent Car Designing And Constructing

Because of the low cost and manipulation of the intelligent car, the car is designed to verify the feasibility of the convolution neural network behavior cloning, as shown in Figure 3. The car's hardware mainly includes: controller Raspberry-Pi, MicroSD memory card, vision sensor, drive motor, standard servo, TP-link wireless router, X-Box and its receiver, etc.

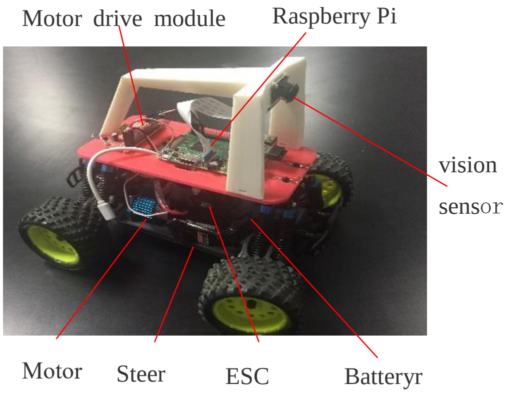


Figure 3 intelligent vehicle

2.1 Vision sensor

We selected a standard wide-angle raspberry-pie camera as the visual sensor, and its basic parameters are shown in Table 2.

Table 2 Main parameters of vision sensor

Pixel	Photosensitive chip	Size
5000000	OV5647	25mm*24mm

Drawing on the principle of Freescale's smart car camera group installing, and combining with the width of the experimental track, the installation principle of visual sensor is shown in Figure 4.

(1) H is the height of the visual sensor installation, in order to have the field of view wider, slightly increase the height of the visual sensor;

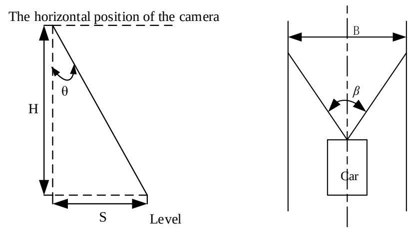
(2) **θ** is the pitch angle of the visual sensor. In order to avoid interference from other objects in the height direction, it is slightly appropriate to make **θ** smaller. The relationship between **H** and **θ** is as shown in (1):

$$\theta = \arctan \frac{S}{H} \qquad (1)$$

(3) β is the horizontal angle of view of the visual sensor. At the pre-point, the visual sensor has a certain width

and the minimum width is **B**. In order to obtain a more complete image confidence, β should slightly larger. The relationship between β , **H** and **S** is as shown in (2):

$$\beta = 2\arctan\frac{B}{2\sqrt{H^2+S^2}} \qquad (2)$$



(a) (b)

Figure 4 Schematic diagram of the installation position of the vision sensor

2.2 Other Major Originals Of The System

The intelligent car uses pulse width modulation (PWM) to control the motor and the steering gear, seting the PWM value of the vehicle speed between [0, 1] (0 means the motor starting, 1 means the maximum speed of the motor), and the PWM value of turn is between [-1, 1] (-1 means left turn maximum angle, 1 means right turn maximum angle), other major component models are shown in Table 3:

Table 3 Main components of the system

Project	Model
Raspberry Pi	3B+
MicroSD	32G
Motor	RS-380SH
Steering engine	E6001
TP-link	TL-WR847N
Wireless Controller	X-Box
Motor Drive Module	PCA9685 16
Battery	Ni-MH SC 7.2V/1100maH

2.3 The Communication Network Setting

By comparing the basic performance of short-range communication [10]. Considering Raspberry's communication method and data transmission rate, We adopted Wi-Fi with 2.4GHz frequency band as communication method in this paper, which realizes communication between various agents in Figure 5.

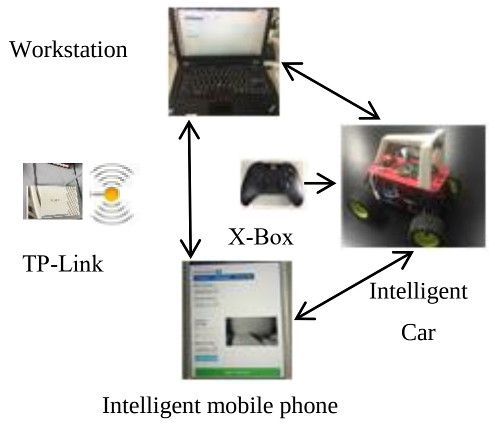


Figure 5 Wireless communication network

3 Convolutional neural network framework construction and experimental scheme

3.1 Convolutional Neural Network Framework

For the intelligent car, Our convolutional neural network framework is shown in Table 4: Input layer is RGB image and vehicle's state information, including the corner and speed.

The model structure of convolutional neural network behavioral cloning consists of five convolutional layers and 2 fully connected neural network layers. The ratio of the training set to the verification set is 8:2. Convolution layer parameter setting: the number of filters is gradually increased, the kernel_size is gradually reduced, and the strides are gradually reduced, so that the image feature information can be recognized accurately, and the activation function is the Relu function.

Table 4 Training model loss results

Project	Parameter
Train	Steps=100,Train_split=0.8 Verbose=1,Min_delta=.0005,Patience=5
Convolution layer	Filters=24,kernel_size=(5,5),strides=(2,2) Filters=32,kernel_size=(5,5),strides=(2,2) Filters=64,kernel_size=(5,5),strides=(2,2) Filters=64,kernel_size=(3,3),strides=(2,2) Filters=64,kernel_size=(3,3),strides=(1,1)
Fully connected layer	Flatten Dense(100, activation='relu') Dropout Dense(50, activation='relu')

	Dropout
Adam	Lr(learning rate)=0.001,Beta_1=0.9, Beta_2=0.999,Epsilon=1e-08

3.2 Experimental plan

The shape of the experimental lane is shown in Figure 6. There are more corners for actual track. The amount of angular loss and speed loss affects the total loss of the model. Therefore, We designed four loss weights to select the weight ratio suitable for this track: (1) corner loss: speed loss = 8:2; (2) corner loss: speed loss = 7:3; (3) corner loss: speed Loss = 6:4; (4) Corner loss: speed loss = 5:5. In addition, considering the number of iterations of training will also affect the total loss of the model, too little will lead to inaccurate training results, too much will produce over-fitting phenomenon, this paper chooses the number of iterations of each training is 50, 100.

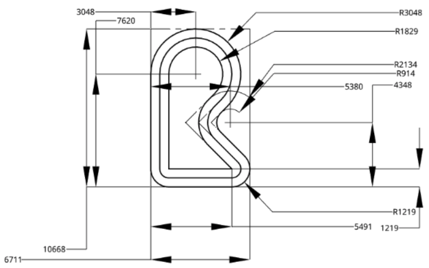


Figure 6 Experimental lane

This paper consists of two experiments: Experiment 1 is based on different weight ratios and iterations designed 8 trainable models; Experiment 2 is to select Experiment 1 to get a feasible training model, and verify the accuracy of the model in the actual track.

4 Experimental results

4.1 Receipt set collection and training model

The acquisition of experimental data is the first phase of the behavior cloning, collecting 7000 three-channel RGB images of 160*120 pixels. According to Experiment 1, the data will import into the constructed convolutional neural network to train model. The training results are shown in Table 5:

Table 5 Training model loss results

Model	Iteration times	Loss weight ratio	Velocity loss	Corner loss	Total loss
1	50	8:2	0.08	0.05	0.06
2	100	8:2	0.06	0.00	0.03
3	50	7:3	0.07	0.04	0.05
4	100	7:3	0.10	0.00	0.02
5	50	6:4	0.08	0.01	0.04
6	100	6:4	0.06	0.00	0.01
7	50	5:5	0.11	0.08	0.14
8	100	5:5	0.09	0.08	0.97

From the above table, we can draw conclusions: (1) As the number of training iterations increase, the speed loss, corner loss and total loss of the model decrease, and the model is more stable; (2) The weight loss and the angular loss weight The ratio is reduced, the total loss of the model is reduced, and the total loss is minimum when the ratio of the corner loss to the speed loss weight is 6:4. In summary, the theoretical corner loss and speed loss weight ratio = 6:4, and the number of training iterations is 100 to obtain the highest model accuracy.

In order to test the feasibility of the model, we divide the training results into three categories: Volatility model, Relative volatility model, and Stable model. As shown in formula (3): where ***f*** is the type of model, and Table 5's classing results is shown in Table 6.

$$f = \begin{cases} volatility\ \mathrm{model} & total\ loss \geq 0.05 \\ relative\ volatility\ \mathrm{model} & 0.02 \leq totalloss \leq 0.05 \\ stable\ \mathrm{model} & tolal\ loss \leq 0.02 \end{cases} \quad (3)$$

Table 6 Model classification

Model type	Model
Volatility model	1、3、4、7、8
Relative volatility model	2、5
Stable model	6

4.2 Verify the accuracy of the model

In order to verify the feasibility and accuracy of the model which get in 4.1, loading the above model into the intelligent car, and the self-driving on the experimental track was carried out for 20 laps. We defined the success as self-driving on the specified track, and the failure was the opposite, such as: rushing out of the track, etc. the experimental results are shown in Table 7. Figure 7 is a screenshot of the intelligent car self-driving experimental video.

Third second　　Fifth second

Seventh second　　Seventeenth second

Figure 7 The experiment video screenshot

Table 7 Verification of model feasibility results

Model type	Success times	Failure times	Success rate
Volatility model 1	16	4	75.0%
Relative volatility model 5	18	2	88.9%
Stable model 6	19	1	94.7%

From the above table, we can conclude that the theoretical ratios of the speed loss: the corner loss = 6:4 and the Stable model obtained by the training iteration number of 100 can be applied in the actual environment. The convolutional neural network behavioral cloning method can realize the self-driving.

5 Conclusion

In this paper, the convolutional neural network behavioral cloning method is studied to realize self-driving in a specific environment. The method uses visual sensors to acquire environmental information, and learns the lane lines in specific environments through convolutional neural networks to simulate the process of human learning Predict the speed and corner of the vehicle. Designing two sets of experiments to verify the feasibility of the method. The method was successfully applied to the first “DIY Robocars KuaiKai” Race Categories. The experimental results and the results of the competition indicate that the convolutional neural network behavior cloning is feasible in a specific environment to achieve self-driving.

6 Acknowledgements

The authors would like to thank Project Supported by Guizhou Province [2017]2027、Project Supported by Guizhou Province [2017]5630 and Project Supported by Guizhou Province [2018] 2168.

References

[1] Shen Wei. Research on Intelligent Vehicle Visual Environment Perception Technology [D]. Nanjing University of Aeronautics and Astronautics, 2010

[2] Shi Q, Zhao J, Han L, et al. Dynamic lane tracking system based on multi-model fuzzy controller[C]// IEEE International Conference on Mechatronics and Automation. IEEE, 2016:873-877.

[3] Zhang Shuangxi. Research on obstacle detection technology of unmanned intelligent vehicle based on radar and camera [D]. Chang'an University, 2013.

[4] Zhang Qi-fei, Guo Tai-liang. Global Path Planning Algorithm of Robot Based on Multistage Decision[J]. Computer Engineering, 2016, 42(10):296-302.

[5] Zhao J, Oya M, Kamel A E. A safety spacing policy and its impact on highway traffic flow[C]// Intelligent Vehicles Symposium. IEEE, 2013:960-965.

[6] Ning Yongjian, Zhao Jin, Zhang Bingkun, et al. Intelligent Control Simulation Based on Vehicle Vehicle Follow Distance[J]. Computer Simulation, 2017, 34(9): 146-150.

[7] Chen S, Shang J, Zhang S, et al. Cognitive map-based model: Toward a developmental framework for self-driving cars[C]// IEEE, International Conference on Intelligent Transportation Systems. IEEE, 2017:1-8.

[8] Rausch V, Hansen A, Solowjow E, et al. Learning a deep neural net policy for end-to-end control of autonomous vehicles[C]// American Control Conference. IEEE, 2017:4914-4919.

[9] Chang Liang, Deng Xiaoming, Zhou Mingquan, et al. Convolutional neural networks in image understanding[J]. Acta Automatica Sinica, 2016, 42(9):1300-1312.

[10] Zhang Fangkui, Zhang Chunye. Research on Short-range Wireless Communication Technology and Its Convergence Development[J]. Journal of Electric Measurement & Instrumentation, 2007, 44(10): 48-52.