Universe Scientific Publishing

CNN-Based End-to-End Unmanned Driving Simulation in Curved Scene

Chenglong Guan, Xinyu Zhang

Chongqing University of Posts and Telecommunications Chongqing, Chongqing 400000, China.

Abstract: Nowadays, the research on autonomous driving technology is more and more in-depth, and the purpose of the research is that autonomous vehicles can achieve safe driving in various driving scenarios. In order to improve the existing autonomous driving scene data set, this paper proposes an end-to-end scene reality simulation implementation method based on CNN. This method is used to train an end-to-end model based on vision sensors for cornering scenes, establishing a direct mapping relationship between vision sensors and driving behavior. In this experiment, the Udacity simulator autonomously collects the data set of the curve scene, and after cutting and normalizing the original data, an end-to-end model with a low loss value is trained, and then the image data is divided into boxes. , Gaussian, median and bilateral filtering operations, the results show that filtering the data can quickly reduce the network loss value. Finally, comprehensive analysis can prove that this method is a high-performance and efficient scene implementation method.

Keywords: Environment Perception; End-to-End; Curved Scene; Image Filtering

Introduction

At present, there are two main methods for realizing autonomous driving. One is to manually define and label the features of images, and explain them. In this traditional way, the errors that appear in the initial lane detection will accumulate continuously in the process of passing to the path planning and logic unit [1]. Therefore, the realization of autonomous driving today is mainly through deep learning, which establishes a direct mapping relationship between car sensors and driving behavior, and realizes an end-to-end driverless mode. Among them, one end is the input of the sensor, such as inputting various road conditions images in the image sensor, and the other end is the corresponding driving behavior output, such as deceleration, braking and other operations.

1. End to end system hierarchical framework

The end-to-end driving model based on deep learning is a very important part of autonomous driving research. The basic idea is very simple, that is, to control driving behavior through machine learning algorithms.

1.1 End to end driverless model

The end-to-end driverless model obtains the image data of the road through the camera, and at the same time obtains the corresponding vehicle control parameters, which are put into the model as the input of the deep model for training. After training the network model, when we input the vehicle real-time road condition image data obtained by the camera into the depth model again, we can obtain the corresponding vehicle control data, and then input these control data into the vehicle wire control system to finally control the automatic driving of the vehicle [2].

1.2 Convolutional Neural Network System

The convolutional neural network (CNN) used in this experiment is composed of four full connection layers, four convolution layers and four maximum pooling layers.

2. Image acquisition and preprocessing design

The problem of data imbalance is prone to occur in practical engineering, but this part of the data is very likely to contain very important and useful information [3]. There are two solutions today: one is to process the data, and the other is to optimize the algorithm [4]. This experiment mainly starts from the former, and balances the data through data preprocessing.

2.1 Collection of data sets

In this experiment, a data set was collected by using Udacity's self-driving car simulator. The data set includes a total of 24,726 images of the curve scene captured by the left, middle and right cameras, and recorded the user's corners. The steering wheel angle, accelerator force, brake force and current speed at the time.

2.2 Data set preprocessing

The image data of the initial data set collected in this experiment at 0 degrees is much larger than the image data of other angles. Such serious data imbalance will cause the final trained network model to be more inclined to drive in a straight line. Therefore, before network training, we must preprocess the data according to various known problems.

First, we randomly sample the data with steering wheel angle of 0 degrees and discard it. Next, we change the image range from [0,255] to [-0.5,0.5] by the formula X=X/255-255, by The normalization operation speeds up network convergence and facilitates fitting. At the same time, in order to reduce the training frequency and improve the accuracy of training, we decided to cut the image size from 320×160 pixels to 320×110 pixels.

The data distribution after preprocessing is much more balanced than before, and it is more suitable to be put into the network for training.

2.3 Image filtering processing

In this experiment, we have carried out four kinds of filtering on the image, namely box filtering, Gaussian filtering, median filtering and bilateral filtering. Among them, the box and Gaussian filtering belong to linear filtering, and the median and bilateral filtering belong to nonlinear filtering. We hope that after filtering the image, it can not only eliminate the interference of noise on the model [5], but also retain the characteristics of the image itself.

3. Training and Testing of End-to-End Models in Curved Scenarios

The purpose of this experiment is to verify the feasibility of the network model based on the end-to-end idea in the curve scene, and to verify whether the image filtering can optimize the end-to-end model.

3.1 Network training results

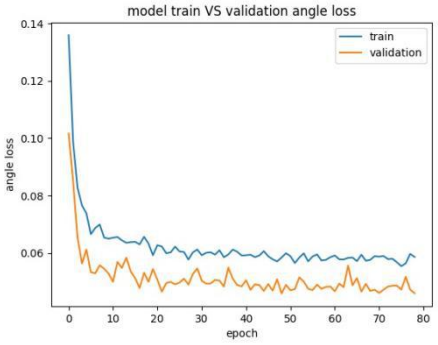

Figure 1 Original image network training results

Taking the training results of the original image network as an example to analyze, it can be seen that 26 local minimum points appear when the original data image has been iterated to 80 times, and the first minimum point is the result after the fourth iteration. The best point is probably in the 21st iteration, and the lowest error by the 80th iteration is 60% lower than the error at the first lowest point, but the training time is about 5.25 times the latter.

3.2 Simulation

After training the model, we put the model into Udacity's autopilot mode for testing. Each model was tested 50 times and placed at different starting points to observe the probability of errors in one lap. The result is as follows:

Table 1 Model simulation results

Model type	Number of successes	Number of failures	Success rate(%)
Original model	36	14	72
Box filtering + original model	39	11	78
Gaussian filter + original model	41	9	82
Median filter + original model	37	13	74
Bilateral filtering + original model	37	13	74

This is the same as the result we obtained by observing the Angle loss curve of the model. In the simulation experiment of the curve scene, the model that has undergone graphic filtering processing has a lower error rate in the automatic driving mode than the original model.

4. Experimental summary

In this experiment, the end-to-end network model is trained under the curve scene and various filtering processes, and finally the direct mapping between the camera image and the vehicle turning is obtained. The experimental results show that the end-to-end learning mode can be successfully used in the curve scene, and filtering the image data can further optimize the network model, and in this scene, linear filtering (Box, Gaussian

filtering) is better than nonlinear Filtering (Median, Bilateral filtering) improves network performance even more.

References

[1] Fu, H.D., Zhao, J., Xi, A.X., etc. An end-to-end method for the corner output of unmanned vehicles [J]. Science Technology and Engineering, 2019, 19(36): 207-211.

[2] Bojarski M , Testa D D , Dworakowski D , et al. End to End Learning for Self-Driving Cars[J]. 2016.

[3] Krawczyk B . Learning from imbalanced data: Open challenges and future directions[J]. Progress in Artificial Intelligence, 2016, 5(4).

[4] Abdi L, Hashemi S . To Combat Multi-Class Imbalanced Problems by Means of Over-Sampling Techniques[J]. Soft Computing, 2015, 19(12):3369-3385.

[5] Wang, T., Ship image filtering and denoising method under multi-layer threshold function [J]. Ship Science and Technology, 2020, 42(24): 73-75.