CNN-Based Vision Model for Obstacle Avoidance of Mobile Robot

Canglong Liu[1,2], Bin Zheng[2]*, Chunyang Wang[1]*, Yongting Zhao[2], Shun Fu[2], Haochen Li[2]

[1] School of Electronic and Information Engineering, Changchun University of Science and Technology, Changchun, 130022, China
[2] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, 400714, China

Abstract. Exploration in a known or unknown environment for a mobile robot is an essential application. In the paper, we study the mobile robot obstacle avoidance problem in an indoor environment. We present an end-to-end learning model based Convolutional Neural Network (CNN), which takes the raw image obtained from camera as only input. And the method converts directly the raw pixels to steering commands including turn left, turn right and go straight. Training data was collected by a human remotely controlled mobile robot which was manipulated to explore in a structure environment without colliding into obstacles. Our neural network was trained under caffe framework and specific instructions are executed by the Robot Operating System (ROS). We analysis the effect of the datasets from different environments with some marks on training process and several real-time detect experiments were designed. The final test result shows that the accuracy can be improved by increase the marks in a structured environment and our model can get high accuracy on obstacle avoidance for mobile robots.

Keywords. End-to-End Learning, CNN; Obstacle Avoidance, ROS

1 Introduction

With the continuous development of science and technology, various mobile robots have been widely used in different fields, such as life services, industrial production, education, entertainment and military area, etc. The mobile robot technology include control theory, mechanical design, computer technology. The ability of mobile robots to navigate and avoid obstacles is an important indicator of the robot's intelligence.

Autonomous navigation and obstacle avoidance for mobile robot need to equip with some range sensors and depend on complex algorithms[1]. Some typical range sensors such laser sensors, ultrasonic sensors, and visual sensors, while these sensors have their own limitations. For example, lasers are more expensive and traditional algorithms based vision are relatively complex.

In recent years, with the development of machine learning, especially deep learning[2], it has became a study hotspot that the robot avoid obstacles by self-learning[3]. Deep learning is an end-to-end learning approach, which is a mapping relationship from input to output through a deep learning network. That is to say, the system automatically learns the characteristics of the data by pouring a lot of data into the algorithm. Levine et al.[4] demonstrate that end-to-end learning is superior to the traditional approach that fixed vision layers . They presented the end-to-end application of CNN for motion planning of robot. They trained a end-to-end learning model based convolutional neural network. The image obtained from a camera as input and output the command for control the robot arm.

Gaya et al. showed the application of automatic obstacle avoidance based CNN for Autonomous Underwater Vehicles (AUVs) [1]. While they don't test the model in real-time and the network is not end-to-end since it consist of requirement of the intermediate procedures. Nicolai et al. use deep learning approach for odometry estimates based lidar data[5]. while their approach need to obtain laser data from expensive LIDAR. Giust et al study a problem of quadrotor trail a forest based CNN[6] . They acquire the dataset by a hiker who equipped with three cameras in head . And they train a CNN model to output control commands for a quadrotor trail the forest following a special path. Ross et al. [7] proposed a method that learn left/right controller for Micro Aerial Vehicles(MAV). They use monocular vision as its sensor, and extracted features from the raw image. The MAV can autonomously navigate through a forest environment. But the system only aimed to the left and right motion, it has to be operated by a person when need to forward motion command.

Compared to the traditional mobile robot obstacle avoidance method, the method of obstacle avoidance based end-to-end learning greatly simplifies the calculation. The system generate steering commands directly by the original pixels[8]. For the problem of robot explore environment, traditional algorithms steps are sense-plan-act. While our main work is try to use the end-to-end model. The main difference between traditional algorithms and end-to-end model is in the

---

* Corresponding author: zhengbin@cigit.ac.cn   wangchunyang19@cust.edu.cn

path planning. Traditional algorithms are complicated, time-consuming and usually requires some sensors. By contrast, our method skips the traditional plan, that is sense-act[9].

In the paper, we present a mobile robot explore in an indoor environment, and the system can rapidly learn feature for avoid obstacles. We don't require high resolution camera, but only a cheap and common camera. The paper is organized as follows: Section 2 introduce the vision model of obstacle avoidance. Our experiment will present in Section 3, which include hardware platform, dataset, train model, a real-time test and other training details. Finally, conclusions and prospect are given in Section 4.

2 Vision model of obstacle avoidance

Deep convolution neural network has made great progress with the development of large-scale computation and GPU. More and more researchers take advantage of CNNs to solver some problems for robot in machine learning.

We control the robot by CNN, which take RGB images as its input. And it has three classes output, include go straight, turn left and turn right. The network architecture is shown in Fig.1, which consists of 5 convolutional layers and 3 fully connected layers. In our environment, extracting and learning feature from our dataset by using Caffe[10] , which is a deep learning open source frame. And we designed a CNN network based AlexNet , which was a winner in ILSVRC-2012 and r presented by Alex[11]. The last fully connected layer was adjusted to 3 nodes, which is consistent with the system included three turning commands.
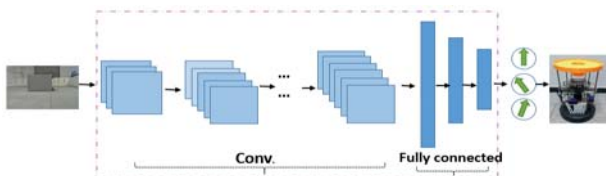

Figure 1. Structure of the CNN

The mobile robot system performed the instructions which was predicted with CNN based Robot Operating System (ROS). ROS is an open source framework for robot, which can provide a similar operating system for heterogeneous computer clusters[12].
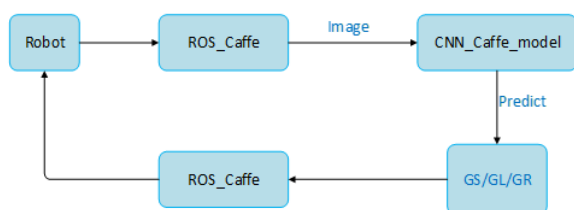

Figure 2. The flow chart of running

We use a package named ROS_caffe which provide a bridge between caffe and ROS. The flow chart is shown in Fig.2. The raw image was acquired by ROS from

robot, a trained CNN model classifies the images to decide which direction will go next for the robot, and specific instructions are executed by the ROS.

3 Experiments and analysis

3.1 Hardware platform and environment

We built a light-weight and small mobile robot based iRobot Roomba and ROS, which is an open source frame. ROS provides a variety of software packages that can be applied to robots, and it played an important role in the design of mobile robot control systems. In addition, the system was equipped with a common camera and a Mini computer.

(a)   (b)
Figure 3. Test environment

In order to demonstrate the problem more effectively, we constructed two types structured environment in indoor based KT foam board. The first type environment as shown in the Fig. 3(a), which was constructed using raw KT foam board. We put some black tape in another type environment, as shown in the Fig. 3(b). Besides, we placed a table in the center as a major obstacle.

3.2 Datasets

CNN need a massive dataset in order to train an effectively model. In these two environments, we operated the robot explore the environment and without colliding into obstacles. At the same time, we recorded the image and control commands by a tool of named rosbag in ROS.

Table 1. The number of training images

|	TL	TR	GS	Total
Dataset 1	1892	1955	1888	5735
Dataset 2	1901	1999	2038	5938

We set the label for each frame by matching the clock of control commands and the clock of images, and the label include Turn left (TL), Turn right (TR), and Go straight (GS). The dataset 1 and dataset 2 was collected respectively in the first and second type environment. We sampled 5735 images from dataset 1, and 5938 images from dataset 2. The special information is shown in table 1. The dataset 1 was spilt in disjoint 5057 images for training and 678 images for testing. The experience use 5178 images for training and 760 images for testing in the dataset 2.

3.3 Training results

We start training based our model, as shown in Fig.1. The model was trained on a workstation equipped with an NVIDA GTX 1080 GPU and NVIDIA cuDNN. And the network can directly generate TL/TR/GS commands from images of camera after the completion of the training.

3.3.1 Training curves

The learning curves were plotted in the Fig. 4, which include test accuracy, train and test loss as the number of iterations. In the Fig. 4(a), the test accuracy eventually achieved 81.72%. And the model achieved a test accuracy 93.21% in the Fig. 4 (b).
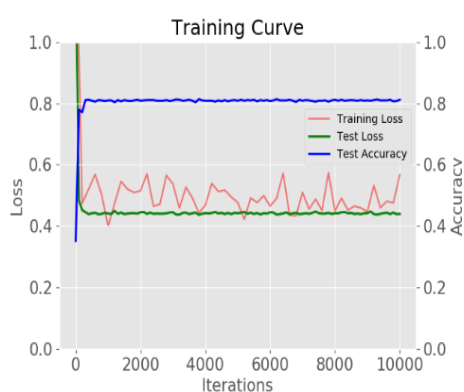
(a)   Train curve of dataset 1
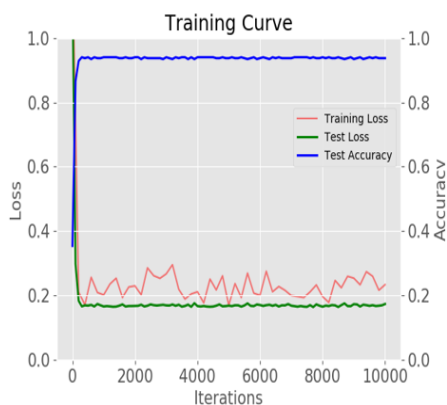
(b) Train curve of dataset 2

Figure 4. Training curves

We can see that the performance of dataset 2 outperform significantly dataset 1 from the training curves. Comparing two classes of the environment, we observe that the more easily the environment is identified, the higher the accuracy, which is similar to recognize the environment for human. The accuracy can be improve by change the environment mark in the constructed environment.

3.3.2 The confusion matrix

The confusion matrix of the dataset 1 classification result was shown in Fig.5. We can see that the overall accuracy

of the dataset 1 is 81.72%. The accuracy rate of the GS class is the highest, which account for 96.08%. The classes TL and TR are low in accuracy. In addition, the error classification of the GR class is often misclassified into the GL class.

	Output class			
	GL	**GR**	**GS**	
GL	20.65% 140	3.83% 26	2.21% 15	77.34%
GR	0.32% 70	24.93% 169	0.44% 3	69.83%
GS	1.47% 10	0.00% 0	36.14% 245	96.08%
				81.72%

Figure 5. The confusion matrix of the dataset 1 classification result

3.3.3 Test result for samples

We tested randomly some sample images with the trained model, shown in Fig. 6 .The average network prediction time of each frame is 4.03 ms. The raw input images are shown in the first line, and the second line are the corresponding predicted output which include turn left/right and go straight.

In the histograms, red mark represents the response probability values of three classes. We observe that CNN model is effectiveness, and it can extract valid information from raw input and predict the right commands.
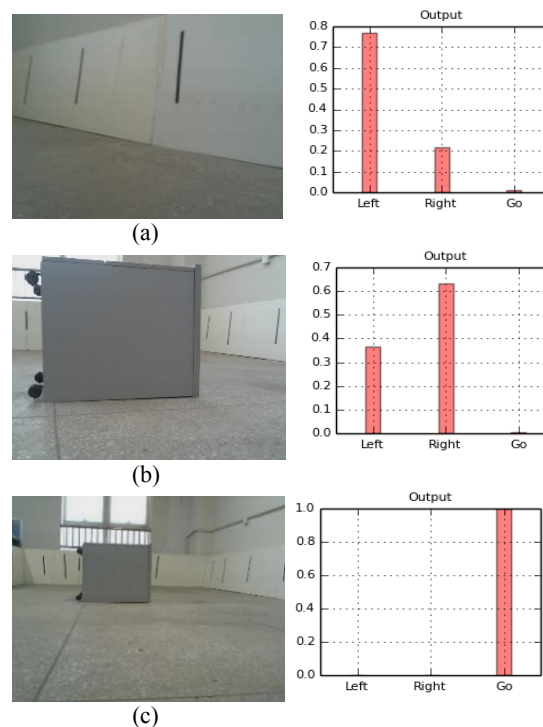

(a)

(b)

(c)

Figure 6. Test result for samples.

3.3.4 Comparison

We compare the result with some others in order to show the effectiveness of the method. As is shown in table 2, Lei Tai et al. tested deep-network solution towards obstacle avoidance in an indoor environment, their overall accuracy is 80.2%[13]. Comparing their method and result, our test was designed in the indoor environment with mark and without mark, respectively. The final test result shows that our model can get high accuracy on obstacle avoidance for mobile robots and the accuracy can be improved by increase the marks in the environment.

Table 2. Accuracy of obstacle avoidance

	Dataset 1	Dataset 2	[13]
Accuracy	81.72%	93.21%	80.2%

3.4 Real-time test

In our real-time-test experiment, we predicted the commands based the model that was trained with the dataset 2. The robot implements these outputs as constant rotational and/or translational velocities. We used a package named ROS_caffe which provide a bridge between caffe and ROS. The system performed the instructions which was predicted with CNN based ROS.

In the test, the robot predicted the instructions of the corresponding action by the trained model and perform the actions by ROS. Throughout the whole experiment, the robot successfully avoid square table in the center of the environment, and it didn't hit around the KT board. In real time testing, the network forecast outputs are shown in the Fig.7. The experiment achieve the goal of the mobile robot obstacle avoidance in explore the environment.
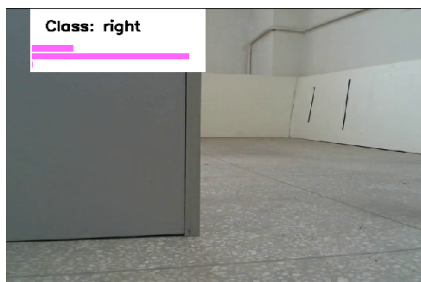

Figure 7. Real-time predict of the network

4 Conclusions and prospect

In the paper, we presented an approach to the mobile robot explore environment and obstacle avoidance using methods of end-to-end learning based CNN. A deep neural network was trained with the dataset and it converts directly the RGB images to steering commands. We also discussed how to improve the accuracy by change the environment mark in a constructed environment. The real-time test experiment our approach can get high accuracy on obstacle avoidance of mobile robots.

In the next work, we will attempt to do it in a more complex setting, include dynamic and non-structure environment. We will solver the tasks of robot navigation based CNN model and provide a significant contribution for the development of intelligence mobile robotic navigation.

References

1. Kovács L. Visual Monocular Obstacle Avoidance for Small Unmanned Vehicles[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016: 59-66.
2. Deng L, Yu D. Deep learning: methods and applications[J]. Foundations and Trends® in Signal Processing, 2014, 7(3–4): 197-387.
3. Jia B, Feng W, Zhu M. Obstacle detection in single images with deep neural networks[J]. Signal, Image and Video Processing, 2016, 10(6): 1033-1040.
4. Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies[J]. arXiv preprint arXiv:1504.00702, 2015.
5. Nicolai A, Skeele R, Eriksen C, et al. Deep Learning for Laser Based Odometry Estimation[J].
6. Giusti A, Guzzi J, Cireşan D C, et al. A machine learning approach to visual perception of forest trails for mobile robots[J]. IEEE Robotics and Automation Letters, 2016, 1(2): 661-667.
7. Ross S, Melik-Barkhudarov N, Shankar K S, et al. Learning monocular reactive uav control in cluttered natural environments[C]//Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013: 1765-1772.
8. LeCun Y, Muller U, Ben J, et al. Off-road obstacle avoidance through end-to-end learning[C]//NIPS. 2005: 739-746.
9. Pfeiffer M, Schaeuble M, Nieto J, et al. From Perception to Decision: A Data-driven Approach to End-to-end Motion Planning for Autonomous Ground Robots[J]. arXiv preprint arXiv:1609.07910, 2016.
10. Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.
11. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
12. Quigley M, Conley K, Gerkey B, et al. ROS: an open-source robot operating system In: ICRA workshop on open source software[J]. IEEE, Kobe, Japan, 2009.
13. Tai L, Li S, Liu M. A deep-network solution towards model-less obstacle avoidance[C]//Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016: 2759-2764.