Vision-based mobile robot navigation through deep convolutional neural networks and end-to-end learning

Yachu Zhang, Yuejin Zhao, Ming Liu, Liquan Dong, Lingqin Kong, et al.

Yachu Zhang, Yuejin Zhao, Ming Liu, Liquan Dong, Lingqin Kong, Lingling Liu, "Vision-based mobile robot navigation through deep convolutional neural networks and end-to-end learning," Proc. SPIE 10396, Applications of Digital Image Processing XL, 1039622 (19 September 2017); doi: 10.1117/12.2272648

SPIE.

Event: SPIE Optical Engineering + Applications, 2017, San Diego, California, United States

Vision-Based Mobile Robot Navigation through Deep Convolutional Neural Networks and End-to-End Learning

Yachu Zhang, Yuejin Zhao, Ming Liu*, Liquan Dong, Lingqin Kong, Lingling Liu
Beijing Key Laboratory of Precision Photoelectric Measuring Instrument and Technology,
School of Optoelectronics, Beijing Institute of Technology, Beijing, 100081, China

ABSTRACT

In contrast to humans, who use only visual information for navigation, many mobile robots use laser scanners and ultrasonic sensors along with vision cameras to navigate. This work proposes a vision-based robot control algorithm based on deep convolutional neural networks. We create a large 15-layer convolutional neural network learning system and achieve the advanced recognition performance. Our system is trained from end to end to map raw input images to direction in supervised mode. The images of data sets are collected in a wide variety of weather conditions and lighting conditions. Besides, the data sets are augmented by adding Gaussian noise and Salt-and-pepper noise to avoid over-fitting. The algorithm is verified by two experiments, which are line tracking and obstacle avoidance. The line tracking experiment is proceeded in order to track the desired path which is composed of straight and curved lines. The goal of obstacle avoidance experiment is to avoid the obstacles indoor. Finally, we get 3.29% error rate on the training set and 5.1% error rate on the test set in the line tracking experiment, 1.8% error rate on the training set and less than 5% error rate on the test set in the obstacle avoidance experiment. During the actual test, the robot can follow the runway centerline outdoor and avoid the obstacle in the room accurately. The result confirms the effectiveness of the algorithm and our improvement in the network structure and train parameters.
Keywords: vision-based, convolutional neural network, navigation, end-to-end learning

1. INTRODUCTION

Autonomous mobile robots have vast potential applications in a wide spectrum of domains such as exploration, search and rescue, transport of supplies, environmental management, and reconnaissance [1]. In contrast to humans, who use only visual information for navigation, many mobile robots use laser scanners and ultra-son sensors along with vision cameras to navigate [2]. But these sensors have its downside too, such as expensive and so on. With the development of CCD, CMOS and other visual sensor technology, many people focus on the research to develop a navigation algorithm for mobile robots using only visual information [3, 4]. Cameras are considerably less expensive, bulky, power hungry, and detectable than active sensors, allowing levels of miniaturizations that are not otherwise possible. More importantly, active sensors can be slow, limited in range, and easily confused by vegetation, despite rapid progress in the area.

Here also a problem that real-time stereo algorithms are considerably less reliable in camera input system. Mapping input images to possible steering angles as a single indivisible task through Convolutional neural network can help have an improvement of system.

Convolutional neural networks model animal visual perception, and can be applied to visual recognition tasks. Some famous CNN models (such as AlexNet [5], VGG-f [6], Google-Le Nets [7], VGG-verydeep-16 [8] and Residual Nets [9]) are widely used in many fields. Convolutional neural networks (CNNs) consist of multiple layers of receptive fields. These are small neuron collections which process portions of the input image. The outputs of these collections are then tiled so that their input regions overlap, to obtain a higher-resolution representation of the original image; this is repeated for every such layer. Tiling allows CNNs to tolerate translation of the input image. Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters. They also consist of various combinations of convolutional and fully connected layers, with pointwise nonlinearity applied at the end of or after each layer. A convolution operation on small regions of input is introduced to reduce the number of free parameters and improve generalization. One major advantage of networks is the use of shared weight in convolutional layers, which means that the same filter (weights bank) is used for each pixel in the layer; this both reduces memory footprint and improves performance.

Applications of Digital Image Processing XL, edited by Andrew G. Tescher, Proc. of SPIE
Vol. 10396, 1039622 · © 2017 SPIE · CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2272648

2. METHODOLOGY

2.1 Robot Hardware

We built a small, rugged and light-weight robot to collect data and validate our control system (as shown in Figure 1). Using a small, rugged and light-weight robot can follow us to facilitate data collection in different condition and test in a wide variety of environment. The downside of our robot is too small to load computing device necessary. Therefore, the robot is remotely controlled by a remote computer. A wireless link is used to transmit image, video and sensor readings to the remote computer. The robot is about 50cm length, with a forward-pointing wireless color camera, a wireless module and a driver module. The horizontal field of view of camera is about 90 degrees and the image captured by camera is 320×240 pixels. The typical speed of the robot during data collection and testing session was roughly 50cm per second.

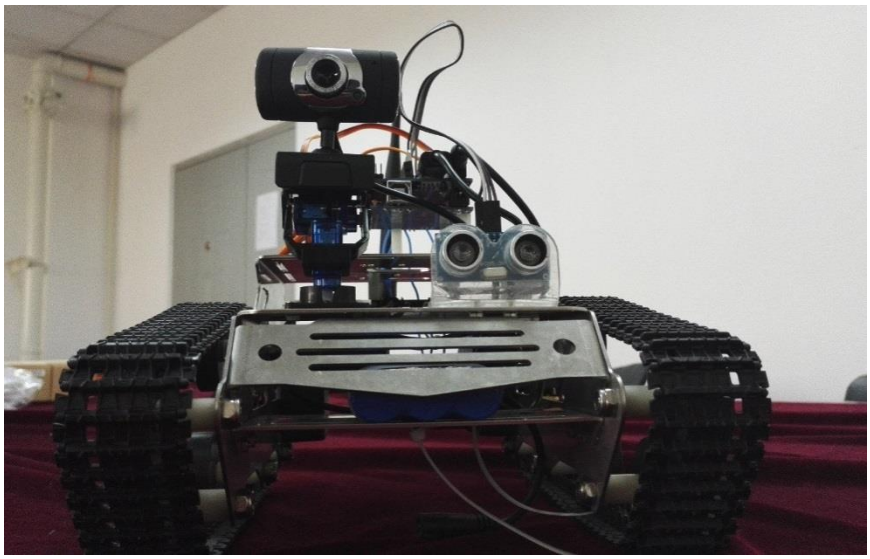

Figure.1 Hardware platform for the system

2.2 Data Collection

Data collection includes line tracking and obstacle avoidance two parts.

For the part of line tracking, the training data was collected by recording the actions of a human driver together with the video data. The human driver remotely drives the robot go straight to follow the straight lines in the ground. If the robot deviate the straight lines in a certain level, the driver should turn right or turn left to ensure robot can follow the lines. When control robot turn left or turn right to a certain extent, let robot go down the straight to follow the lines again and then repeat the above procedure. During each run, remote computer record the output of the video camera at 5 to 10 frames per second, together with the speed of wheels setting from the operator.

A crucially important requirement of the data collection process was to collect large amounts of data with enough diversity of type of lines, weather conditions, lighting conditions and different times of day. Besides, it is necessary for the human driver to adopt a consistent line tracking behavior. To ensure this, the driver should follow the same deviation level to adjust the direction of the robot.

We divide the driving direction into three conditions: go straight, turn left and turn right. The definition of the rate of deviation as follow: when the robot deviate to the right side to a certain level (image captured by camera as shown in Figure 2(a)) or beyond this level, we control robot to turn left. When the robot turn left to a certain level (image captured by camera as shown in Figure 2(b)), we control robot to go straight. The condition of left deviation is similar to the right deviation and the deviation level to turn right is shown in Figure 2(c).

Figure.2 (a) : Driving deviation level to turn left; (b) Driving deviation level to go straight; (c) Driving deviation level to turn right

Three type of video (turn left, turn right and go straight) is recoded by following the runway center in different weather conditions and light conditions. The data was recorded and archived at a resolution of 320×240 pixels at 5 frames per second. A total of about 100 clips are collected with an average length of about 30 frames each. We get about 3000 pictures (684 pictures for left, 512 pictures for right and 1730 pictures for straight) and crop them into the right input size (149×58 pixels). To augment our data set and avoid over-fitting [10], Gaussian noise and Salt&Pepper noise is added to the cropped image (as shown in Figure 3(a) and Figure 3(b)). Finally, we obtained 11319 pictures (3585 pictures for left, 3585 pictures for right and 4149 pictures for straight) and about 9000 images is used for training, others for validation/testing.

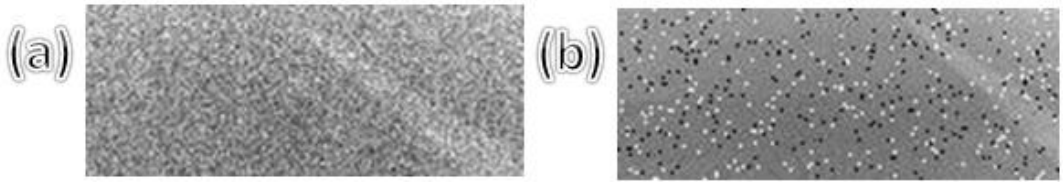

Figure.3 (a): Training image with Gaussian noise; (b) Training image with Salt&Pepper noise

For the obstacle avoidance part, we have the similar data collection strategy to the line tracking part. We use chair as obstacle in the room (as shown in Figure 4) and divide the driving direction into two conditions: go straight and turn right. To ensure the driver can adopt consistent obstacle avoidance behavior, we drive the robot straight ahead whenever no obstacle was present within a threatening distance. The robot was told to turn right at the same distance to the obstacle during each run. 1407 pictures (824 for right and 583 for straight) were made and after the data augmentation, we get about 3000 pictures finally.

Figure.4 Obstacle of the indoor environment

2.3 Learning system

A single 15-layer CNN learning system is proposed (as shown in Figure 5). The architecture of our convolutional net includes many different layers: convolutional layers, batch normalization layers, pooling layers, dropout layers, ReLU layer, full connection layer, softmax layer and so on.

Convolutional layer: Convolutional layer is the core block of the CNN. Local feathers of images are extracted through kernels (filters) and non-linear transform. The units in the convolutional layer connect with the local receptive fields of the previous layer to reduce the number of parameters. Each unit receptive the information of different area and considered in the higher layer to get the global information. To reduce the number of weights, an optimization method named weight-sharing is proposed which means using the same parameters in the same local area.

Batch normalization layer: Internal covariate shift is the change in the distribution of network activations due to the change in network parameters during training. To improve the training, batch normalization layer is used to reduce internal covariate shift and help address issues with gradient explore and gradient vanish to enable a higher learning rate [11]. It can regularize the model too.

Pooling layer: Although the feather captured through convolutional layer can be trained using classifier, there still have some problems like over-fitting and data redundancy. The pooling layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters and amount of computation in the network, and hence to also control overfitting. It is common to periodically insert a pooling layer between successive convolutional layers in the CNN architecture.

Dropout layer: Deep neural nets with a large number of parameters are very powerful machine learning systems. However, due to the limited training data and complicated relationship between inputs and outputs, overfitting is a serious problem in such networks [12]. Dropout layer is used to prevent complex co-adaptation on the training data to reduce the overfitting. The term "dropout" refers to dropping out units (hidden and visible) in a neural network [13]. Some units are temporarily removed from the network randomly with a probability of 0.5 usually. This prevents inter-dependencies from emerging between units and allows the network to learn more and more robust relationship.

ReLU layer: ReLU is the abbreviation of Rectified Linear Units. The rectifier function is an activation function (as equation (1) shows) which can be used by neurons just like any other activation function. Activation function is the relationship between the inputs and outputs of each unit and has a major impact to the convergence. Using sigmoid function and tanh function may cause gradient vanish and lead to a worse training performance [14]. To solve this problem, ReLU is used. Another advantage is ReLU can achieve convergence more quickly in the training procedure with higher accuracy.

$$f(x) = Max(0, x) \tag{1}$$

Full connection layer and softmax layer: Full connected layers are used to transfer the location related features into scores of different classes and a final softmax layer transforms the scores again into class likelihood probabilities.

The first layers of our structure is a convolutional layer of size 147 ×58 through 3 ×3 kernels and the next layer is a batch normalization layer (the same in 6-th layer and 9-th layer). The 3-rd layer is a pooling/subsampling layer of size 49 ×14 and the subsampling ratios are 3 horizontally and 4 vertically. The 4-th layer is a dropout layer with 0.5 dropout rate (the same as 11-th layer). The 5-th, 8-th, 12-th layers are convolutional layers of size 45 ×12 through 5 ×3 kernels, size 6 ×2 through 4 ×3 kernels, size 1 ×1 through 2 ×2 kernels respectively. The 7-th, 10-th layers are pooling/subsampling layers of size 9 ×4 with 5 ×3 subsampling ratios, size 2 ×2 with 3 ×1 subsampling ratios respectively. The 13-th layer, 14-th layer and 15-th layer is ReLU layer, full connection layer and softmax layer respectively. The three outputs respectively code for "go straight", "turn left" and "turn right".

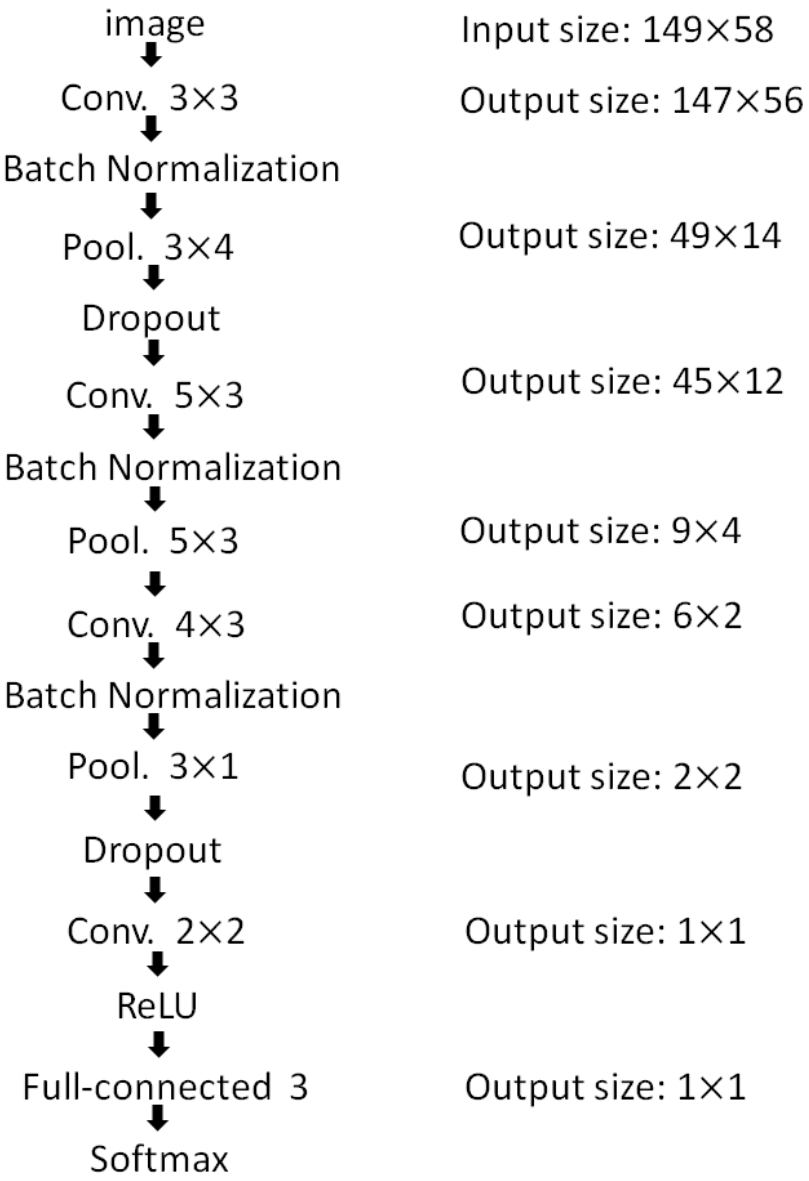

Figure.5 CNN structure

3. EXPERIMENT AND RESULT

The complete experiment workflows as follow: The images are captured by the camera in the robot and saved in a specific folder several times per second. The image is sent to the pre-trained network after cropped to the appropriate input size and get the class of the image. Then the command of computer control robot to change direction according to the class of the input image.

Different experiment use different pre-trained net. The pre-trained net is trained using each own data set with the same CNN structure mentioned above.

3.1 Line Tracking

For the line tracking experiment, we train the network with all 11319 pictures (3585 pictures for left, 3585 pictures for right and 4149 pictures for straight). Different kind of structures are selected to get the optimal structure and we obtain about 3.29% error rate on the training set and 5.1% error rate on the test set finally. We test system both in the indoor environment and outdoor environment. In the outdoor environment, the robot can follow the runway centerline accuracy in the playground (as shown in Figure 6). During the indoor environment test, a tin foil is cut into thin strips and lay on the ground of the room (as shown in Figure 7). The robot can track the tin foil accuracy, which means the system have a wide applicability.

Figure.6 Line tracking experiment in outdoor environment

Figure.7 Line tracking experiment in indoor exnvironment

3.2 Obstacle avoidance

For the obstacle avoidance experiment, we train the network with about 3000 pictures. Similar to the line tracking experiment, we test different structure and get 1.8% error rate on the training set and less than 5% error rate on the test set. In the actual test, the robot can avoidance irregularly placed obstacles accuracy in the room (as shown in Figure 8).

Figure.8 Indoor environment obstacle avoidance experiment

4. CONCLUSION

A proposal of mobile robot navigation method through DCNN and end to end learning is presented. A large 15-layer convolutional neural network was trained with limited data and we use dropout layer, batch normalization layer and data augment to help improve training performance and avoid overfitting. The main advantage of the system is its robustness to the diversity of situations in test environments and its high recognition rate. The learning system is trained from raw image to directly direction which can essentially eliminate the need for complex image pre-processing and other computational procedure. The actual test result in line tracking experiment and obstacle avoidance experiment yield a solid evidence that our learning system is reliable. Robot can find the optimal running route in different conditions with high recognition rate. The overall results demonstrate the effectiveness and the efficiency of the proposal to autonomous mobile navigation.

FUNDING

National Natural Science Foundation of China (NSFC) (61301190, 61475018).

REFERENCE

[1] Y. Lecun, et al. "Off-road obstacle avoidance through end-to-end learning." in *International Conference on Neural Information Processing Systems* MIT Press, 739-746 (2005).

[2] J. M. Choi, S. J. Lee, and M. Won. "Self-learning navigation algorithm for vision-based mobile robots using machine learning algorithms." in *Journal of Mechanical Science & Technology*, 25(1), 247-254 (2011).

[3] E. R. Fossum, "Cmos image sensors: electronic camera-on-a-chip." in *Electron Devices Meeting. iedm.technical Digest.international*, 44(10)**,** 1689-1698 (1997).
[4] R. Lange, P. Seitz. "Solid-state time-of-flight range camera." in *IEEE Journal of Quantum Electronics* 37(3), 390-397 (2002).
[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Conference and Workshop on Neural Information Processing Systems*, pp. 84-90 (2012)
[6] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," in *British Machine Vision Conference*, pp. 1-12 (2014)
[7] C. Szegedy, W. Liu, YQ. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9 (2015)
[8]K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, "Veryeep Convolutional Networks for Large-Scale Image Recognition," in *the international conference on learning representations*, (2014)
[9] K. He, X Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778 (2016)
[10] P. Y. Simard, D. Steinkraus, and J. C. Platt. "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis." in *International Conference on Document Analysis and Recognition, Proceedings* IEEE, 958 (2003)
[11] S. Ioffe, C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." in *Computer Science*, (2015)
[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," in *Journal of Machine Learning Search*, 15, 1929-1958 (2014)
[13] Hinton, Geoffrey E, et al. "Improving neural networks by preventing co-adaptation of feature detectors." in *Computer Science*, 212-223 (2012)
[14] V. Nair, G. E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines." in *International Conference on Machine Learning*, 807-814 (2010)