ISBN: 978-981-18-7950-0
2023 the 13th International Workshop on Computer Science and Engineering (WCSE 2023)
doi: 10.18178/wcse.2023.06.029

Algorithm for Mobile Robot Localization Based on Recurrent Convolutional Neural Networks

Li Shaowei [1], Xiang Cairong [1 +]

[1]School of Artificial Intelligence, Jianghan University, China

Abstract. Mobile robot localization has been considered to be an important task in the field of robotics research. Recurrent Convolution Neural Networks -based Mobile Robot Localization (RCNN-MRL) Algorithm is proposed in this paper. RCNN-MRL estimates self-position from the first person view captured by a camera on a robot using Recurrent Convolution Neural Networks (RCNN).It uses a regression model for localization using RCNN capable of processing consecutive images. We use simulated environments where a two-wheel robot moves randomly, and analyze the performance of localization. Our experiments show that RCNN model can estimate the self-position of the robot.

Keywords: Mobile Robot , Localization, Convolution Neural Networks , two-wheel robot, Time Series Image

1. Introduction

Many kinds of robots, such as cleaning robots and navigation robots, have greatly facilitated people's daily life. The key of these robots is to estimate the current position, which is also called robot localization, and it is also a research hotspot in the field of mobile robots[1].

Currently, there are two main classes of algorithms to estimate the machine position. The first algorithm is to estimate the robot position by using distance information, such as ultrasonic, Bluetooth and Wi-Fi ranging; the other algorithm is to estimate the robot position by using the camera to take images and extract position information from the images [2-3].

Recently, Convolution Neural Networks (CNN) have been widely used in computer vision and image processing [4]. Reference [5] proposes to use CNN to independently propose image features. The image is transformed into low-dimensional feature information by CNN. Based on these characteristic information, the position information can be estimated. In this paper, CNN is used as a regression model to estimate the position directly from the robot's First Person View Image (FPVI). Through this strategy, the robot position can be estimated in the unknown map information environment.

The main work of this paper is as follows: Firstly, the RCNN model is proposed by combining CNN with Recurrent Neural Networks (RNN). The RCNN model can provide continuous images. Then, the robot position is estimated based on the RCNN model. Then, a two-wheeled robot is used, and the robot is set to move randomly, and then the time series image information is obtained. Finally, the localization performance of the robot based on RCNN model is tested.

2. RCNN Model

This section details the RCNN model. Firstly, the structure of CNN and Recurrent Neural Networks (RNN) are analyzed, and then the hybrid model RCNN based on CNN and RNN is analyzed.

2.1. CNN

CNN is mainly composed of convolution layer, pooling layer, fully connected layer and output layer. Generally, the convolution layer and the pooling layer generally have multiple layers and appear alternately. The convolution layer is composed of multiple feature planes, and each feature plane is composed of

+ Corresponding author. Tel.: +86 18171385977.
E-mail address: 55440881@qq.com.

multiple neurons [6]. Each neuron is connected to the local region of the feature surface of the previous layer through a convolution kernel, where the convolution kernel is a weight matrix, for example, if it is a two-dimensional image, the convolution kernel is, or matrix [7-8].

Figure 1 shows the neuron model, which is a multiple-input single-output model. The relationship between output and input is shown in formula (1):

$$y_j = f\left(b_j + \sum_{i=1}^{n} x_i \omega_{ij}\right) \tag{1}$$

Where x_i is the input image data, and each input signal is input to the neuron j at the same time. ω_{ij} represents the weight value of the connected neuron j. b is the internal state of the nerve, i.e., the bias value, and $f(\ )$ is an activation function. In this paper, the Rectified Linear Unit (ReLU) [9] is selected as the excitation function.

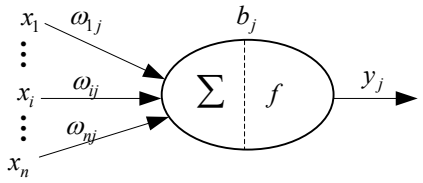


Fig. 1: Neuron Model

Immediately after the convolution layer is the pooling layer. The purpose of the pooling layer is to obtain the features of spatial invariance by reducing the resolution of the feature surface [10]. That is to say, the pooling layer serves the purpose of extracting image features for the second time. Each neuron in the pooling layer pools a local receptive field. In addition, this paper chooses maximum pooling as the pooling method. The so-called maximum pooling method is to take the point with the maximum value in the local acceptance region.

2.2. LSTM

After obtaining the features through CNN, RNN is used to extract the time series data of the features. Long Short Term Memory (LSTM) [11] as an RNN improvement. LSTM retains the advantages of RNN for modeling time series data. The model of RNN expansion in time is shown in Figure 2.

x_t means input vector of time t is shown in figure 2. Calculate the state value h_t of the hidden layer at time t through x_t and the state value h_{t-1} of the hidden layer at the previous moment t-1, as shown in formula (2):

$$h_t = f\left(x_t, h_{t-1}\right) = \sigma\left(W_{xh} x_t + W_{hh} h_{t-1} + b_h\right) \tag{2}$$

Where f denotes the RNN function. σ represents the sigmoid function , W_{xh}, W_{hh} respectively represent the weight matrices input to the hidden layer and from the hidden layer to the hidden layer. And b_h is the bias vector of the hidden layer. Repeat until you have read all the input.

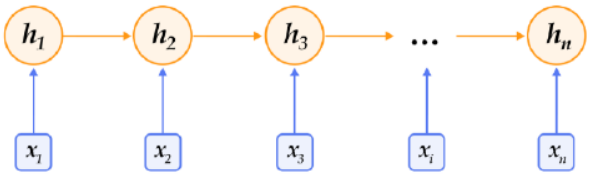


Fig. 2: RNN Expansion by Time

LSTM is based on the improvement of RNN, adding some units to avoid the problem of tonality disappearance in the training process [12]. The LSTM adds memory cells C, input gates i, output gates o, and forgetting gates f. By arrange that gates and the memory unit, the data processing capability of the RNN is improved. The functional relationship between these gates and memory cells is as follows:

$$i_t = \sigma\left(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i\right) \tag{3}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \tag{4}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh\sigma\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{5}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o\right) \tag{6}$$

$$h_t = o_t \cdot \tanh\left(c_t\right) \tag{7}$$

Where c_t, o_t, f_t and i_t denote the memory cell, output gate, forgetting gate, and input gate vectors, respectively. The W with subscripts represent the corresponding weights, and the b with subscripts represent the bias values. tanh represents the hyperbolic tangent function.

The input gate, the output gate, and the forget gate control the strength of the memory cell, but each uses a different control method, as shown in Figure 3.

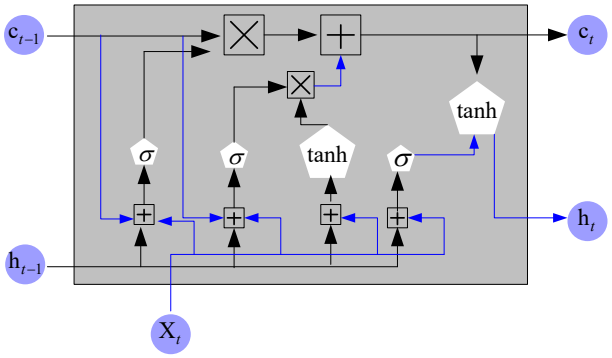


Fig. 3: Timing Diagram of LSTM

2.3. RCNN

In this paper, RNN and CNN are combined, and the proposed RCNN model is shown in Figure 4.

As shown in Figure 4, CNN is used to extract information from the image, and then the feature vector is output as the input of LSTM, and the robot position is estimated by LSTM. RCNN is a self-learning algorithm to estimate the robot position.

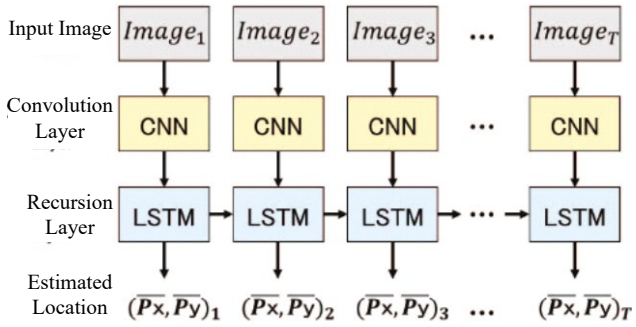


Fig. 4: Location Model based on RCNN

3. Robot Motivation Model

A two-wheeled robot is used to estimate the position of the robot by moving the robot in the surrounding environment, as shown in Figure 5.

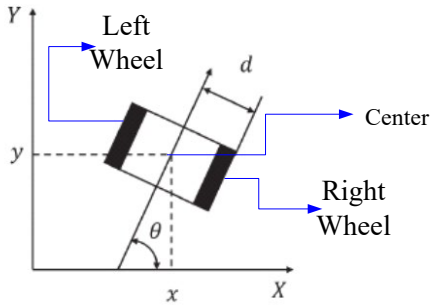


Fig. 5: Two-wheeled Robot Model

Assume θ_r and θ_l represent the speeds of the right and left wheels, respectively. Therefore, the translation velocity θ and angular velocity of the robot ω can be obtained as shown in equations (8) and (9):

$$\vartheta = \frac{\vartheta_r + \vartheta_l}{2} \tag{8}$$

$$w = \frac{\vartheta_r - \vartheta_l}{2d} \tag{9}$$

Where d represents the distance of the center of the robot from the wheel.

Once the translation and angular velocities are determined, the robot position coordinates and orientation are updated according to equations (10), (11), and (12):

$$\theta \leftarrow \theta + \Delta t w \tag{10}$$

$$x \leftarrow x + \Delta t \vartheta \cos\theta \tag{11}$$

$$y \leftarrow y + \Delta t \vartheta \sin\theta \tag{12}$$

Once the position is updated, the velocities of the right and left wheels are updated with probability P, as shown in equations (13), (14), and (15):

$$\vartheta_{\min} \leftarrow \alpha r_1 \tag{13}$$

$$\vartheta_r \leftarrow \beta r_2 + \vartheta_{\min} \tag{14}$$

$$\vartheta_l \leftarrow \beta r_3 + \vartheta_{\min} \tag{15}$$

Where, r_1, r_2 and r_3 are all random variables from 0 to 1. And, α, β are the minimum value ranges for setting the speeds of the left and right wheels.

In the simulation process of the next section, the relevant parameters of the robot mobility model are as follows: $d = 0.1\text{m}$ 、 $\Delta t = 0.001$ 、 $\alpha = 20.0$ 、 $\beta = 4.0$ 、 $P = 1.0\%$.

4. Experiment Analysis

4.1. Image Acquirement

The selected area(40m*40m) is used as the experimental environment, as shown in Figure 6 (B). The first image (FPVI) is taken at the initial position $(x,y,\theta) = (-15,15,-\pi/2)$, as shown in Figure 6 (a), and the image height and width are 480 and 480, while the channel width is 3, that is 480*480*3.

(a)FPVI (b) Simulation Environment

Fig. 6: Image Environment

Initially, the robot is placed in the initial position and the RCNN is initialized, and then the robot moves according to equations (8)- (15). In addition, the camera shoots at 4.0 FPS while recording the position, and a time series is defined as 10 consecutive pictures. After completing a sequence, the robot moves.

In order to better analyze the position estimation capability of RCNN-MRL in different environments, a normal environment and an obstacle environment are set, and 1200 sequences of data are collected in each environment. Figure 7 shows the normal and obstacle environments, respectively.

(a)Normal (b)Obstacle (c) Coverage for Obstacle

Fig. 7: Obstacle Environment

To test the effect of obstacles on robot localization, the reference obstacle coverage represents the number of obstacles in the environment. As shown in Figure 7 (C), using the background extraction algorithm [13], the obstacle coverage rate is 28.6%.

4.2. Location Performance Analysis

In this section, the image data obtained in Section 3 is used to train the RCNN model, in which the first 1000 of the 1200 sequences are used as training, the last 200 sequences are used as test data, and the number of iterations is 1000.

In addition, Euclidean Distance Loss (EDL) is selected as the performance index, which is defined as shown in formula (16):

$$Loss(I)=\sum_{t=1}^{T}\left(\left(\overline{P}_{x_t}-P_{x_t}\right)^2+\left(\overline{P}_{y_t}-P_{y_t}\right)^2\right) \tag{16}$$

Where $\left(\overline{P}_{x_t},\overline{P}_{y_t}\right)$ represents the estimated position and $\left(P_{x_t},P_{y_t}\right)$ is the true position of the robot. I represents the sequence data of the input image, and the sequence length is T.

Figure 8 shows the average EDL performance of robot localization with RCNN and CNN models under normal environment.

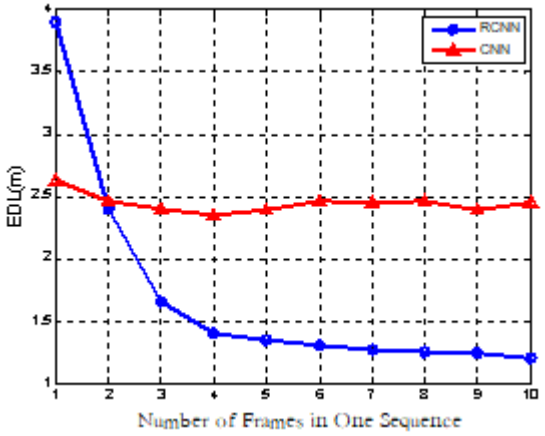
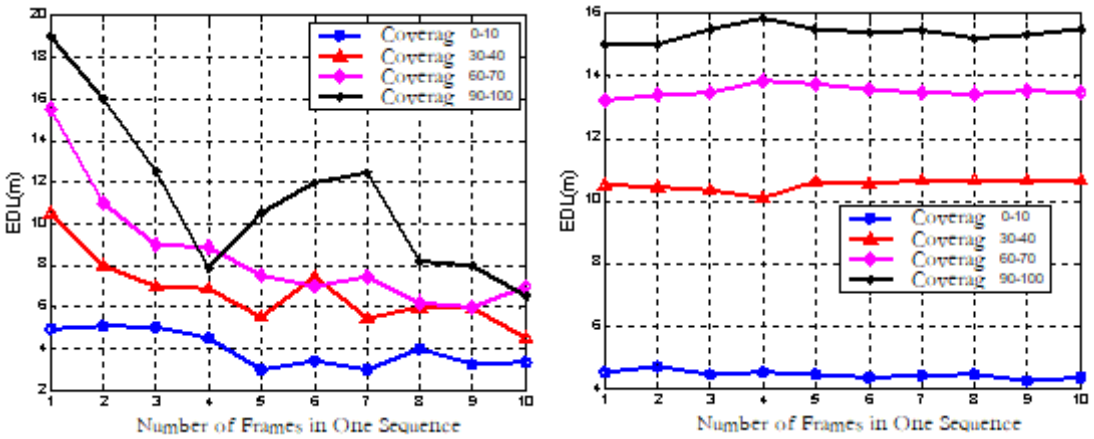


Fig. 8: Average EDL for RCNN and CNN Models(Normal Environment)

It can be seen from Figure 8 that the overall EDL performance based on the RCNN model is better than that of CNN. In the first two frames, the EDL based on RCNN model is lower than that based on CNN. For example, at the time of the first input image, the EDL based on the RCNN model is 3.9 meters, while the EDL of the CNN model is 2.5 meters. However, as the number of input images increases, the EDL based on

the RCNN model gradually decreases, and at the tenth image, the EDL reaches 1.2 meters. These data show that the proposed RCNN-based model can estimate the robot position using time-series data.



(a) EDL for RCNN (b) EDL for CNN

Fig. 9: Average EDL for RCNN and CNN Models(Obstacle Environment)

Figure 9 shows the average EDL performance in the obstacle environment. It can be seen from figure 9 that the proposed RCNN model can effectively estimate the robot position. The EDL of figure 9 is much increased compared to figure 8. This shows that obstacles increase the difficulty of robot positioning. It is also easy to see from figure 9 that the larger the obstacle coverage ratio is, the larger the EDL is. However, compared with CNN, the RCNN model can effectively estimate the robot position.

5. Conclusion

In this paper, the RCNN model is proposed and used to estimate the robot position directly from the first person view image of the robot, which makes full use of the ability of RCNN to process continuous multiple image information. The experimental data show that the proposed RCNN model can accurately estimate the position of the mobile robot. At a lat stage, that robustness of the RCNN model is tested in a more complex real environment.

6. Acknowledgments

We would like to acknowledge the invaluable help of my colleagues, who have given my constant consultant in my paper writing, and have supported me throughout the whole process of the papers; and appreciate financial support from Jianghan University (Fund No. 2021yb053).

7. References

[1] Zheng JianSheng, Yang Wei. Attitude Adjustment of Wheeled Robot's Fixed-Point Motion[J]. Chinese Journal of Electron Devices, 2017,40(1):158-161.

[2] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs[C]. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015:34-42

[3] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description[J].in CVPR, 2015:43-46.

[4] Lv Guohao, Luo Siwei, Huang Yaping, Jiang Xinlan. A Novel Regularization Method Based on Convolution Neural Network[J]. Journal of Computer Research and Development, 2014,51(9):1891-1900.

[5] N. S ¨underhauf, F. Dayoub, S. Shirazi, B. Upcroft, M. Milford.On the performance of convnet features for place recognition[M].CoRR, vol. abs/1501.04158, 2015. [Online]. Available:http://arxiv.org/abs/1501.04158

[6] Zhou Feiyan, Jin Linpeng, Dong Jun. Review of Convolutional Network[J]. Chinese Journal of Computers,2017,40(6):1230-1253.

[7] LeCun Y, Bottou L, Bengio Y. Gradient-based learning applied to document recognition[J].Proceedings of the

IEEE, 2016,86(11):2278-2324.

[8] Gao-Li Gang, Chen Pai Yu, Yu Shi-Meng. Demonstration of convolution kernel operation on resistive cross-point array[J].IEEE Electron Device Letters, 2016,37(7):870-873.

[9] Nair V,Hinton G E, Farabet C. Rectified linear units improve restricted Boltzman machines[C].Proceedings of the 27th International Conference on Machine Learning, 2014:807-814.

[10] Gu Jiu-Xiang, Wang Zhen-Hua, Jason Kuen. Recent advances in convolution neural networks. arXiv:1512.07108v5,2017.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory[J].Neural Comput., 2017,9(8): 1735-1780.

[12] Khan W, Jiang P, Holton R. Word spotting in continuous speech using wavelet transform[C]//Proc. the 2014 IEEE International Conference on Electro Information Technology (EIT). Milwaukee: IEEE, 2014:275-279.

[13] Z. Zivkovic .Improved adaptive gaussian mixture model for background subtraction[C].in Pattern Recognition, 2014. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2. IEEE, 2014:28-31.