Control Method Based on Deep Reinforcement Learning for Robotic Follower with Monocular Vision

Dongdong Wang a, Feng Qiu b and Xiaobo Liu c

Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China 110819

Email: a 20163196@stu.neu.edu.cn, b 20162803@stu.neu.edu.cn, c20163229@stu.neu.edu.cn

Abstract: Robotic follower is receiving attention widely in recent years. Aiming at the problems of low sample collection efficiency, high training cost and difficult design of reward function in the real world, we propose a control method based on deep reinforcement learning. Different depth layers are adopted to attain the end-to-end control of the robotic follower through pre-trained. Then, we design a reward function mechanism to judge whether the robot follower follow falsely. Then the appropriate pre-trained network is transferred to reinforcement learning, and a deep reinforcement learning system for monocular vision robot following tasks is established. According to the experimental results, the proposed deep reinforcement learning method can efficiently collect a large number of data sets, shorten the training period and reduce the number of times that the robot follower loses its target.

1. Introduction

Robot follower has received widespread attention in recent years. Traditional improvements to existing following robotic systems are mostly sensor-based updates, which typically employ single sensor or multiple sensors. The former mostly uses monocular or binocular cameras or Kinect, etc. [1], which has cumbersome calibration steps and it is difficult to adapt to sudden changes in the real environment. The latter combines vision sensors with other methods such as laser ranging and ultrasonic navigation [2]. However, these methods also mean higher cost and complex data fusion.

With the improvement of computing resources, the application of deep learning is increasing rapidly. Wang N et al. [3] first proposed to use "offline pre-trained + online fine-tuning" method to solve the problem of insufficient training samples in target tracking. C. Ma [4] et al. used convolutional networks to extract depth features and achieved tracking through correlation filtering. Cui, Zhen [5] et al. use the multi-directional recurrent neural network to model and identify the reliable part that is useful for overall tracking. This application can reduce cost and avoid complex manual marking feature processes.

Reinforcement Learning (RL) is a strategy to learn the goals by maximizing accumulated future rewards. V. Mnih [6] et al. proposed the Deep Q Networks (DQN) algorithm, which demonstrates superior performance over deep learning with the goal of learning the state-action value function \( Q_A(s,a) \) given by deep networks by minimizing time difference errors. S. Yun [7] et al. used a deep network to classify a set of action sequences for pre-trained, and then updated the model through reinforcement learning. In this way, they achieved better control performance.

Existing deep reinforcement learning algorithms for visual target tracking are rarely used in...
real-world robotic following systems. To solve this problem, the paper proposes a deep reinforcement learning following robot system based on monocular vision. The system not only has efficient data set construction ability but also can control the robot end-to-end. Moreover, it shortens the deployment time of reinforcement learning in actual robots. Experiments prove its reliability in real-world applications.

2 System Architecture and Algorithm

2.1 End-to-end controlled VGG-11 network

In order to reduce the complexity of the algorithm network, we set the forward motion speed of the robot to a fixed value in the actual following task. Only by adjusting the steering of the robot allows the robot to always follow the specific target. Therefore, we constructed a deep learning pre-trained network for following the end-to-end direction control of the robot.

In this paper, a monocular color camera is used as the only input sensor, and the rotary motion commands of the robot are used as the output of the system. Considering the different relative position of person and robots, this paper maps the horizontal position information of the specific person in a single frame input image to the rotary motion command of the robot one by one:

- left of image → one-hot coding → [1,0,0] → rotary command → left turn
- middle of image → one-hot coding → [0,1,0] → rotary command → go forward
- right of image → one-hot coding → [0,0,1] → rotary command → right turn

This mapping is end-to-end. The positions of the specific person in the image are simply divided into three categories: left, middle, and right, and the classifier is pre-trained using the VGG-11 network proposed by Simonyan [8] et al. By classifying the positions of the specific person in the image, the network can output control commands end-to-end.

The network structure is shown in Figure 1. And cross-entropy loss function is used in the classifier:

$$L(\theta) = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} x^{(i)} \log \frac{e^{\theta^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta^T x^{(i)}}}$$  

(1)

Where $\theta$ is the parameter matrix of the network model; $k$ is the number of classifications; $m$ is the number of samples per training batch; $x^{(i)}$ is the i-th image sample; $y^{(i)}$ is the real label of the i-th sample.

![Figure 1. VGG-11 structure diagram](image_url)

In order to achieve the classification effect of the VGG network, it is necessary to collect images with various complex scenes. In addition, in order to reduce the time of collecting data set, we put specific people in the background of complex scene films. Using projection aids to collect data set in complex scenes. Then, we use KCF [9] algorithm to calibrate the target in real time and generate one-hot code according to the position of the calibration frame at the scene. Lastly, we further extend the data set by data enhancement. Figure 2 is a portion of a data-enhanced data set.
2.2 Update of the DQN model

To improve DQN, this paper embeds the pre-trained VGG into the deep network, which determines a suitable system for this task. Next, we combine the deep learning algorithm with the Q-learning [10] algorithm in the reinforcement learning algorithm to improve the DQN model.

2.2.1 Deep reinforcement learning. Reinforcement learning is a process in which an agent interacts with the environment and learns the optimal strategy \( \pi \) that obtains the maximum cumulative return by policy iteration method.

The cumulative return obtained by the agent following the strategy \( \pi \) at time \( t \) is defined as follows:

\[
G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}
\]

Where \( \gamma \) is the discount factor, and \( \gamma \in (0,1] \).

The state-action value function measures the degree of good or bad that the agent takes action \( a \) under state \( s \), which is defined as follows:

\[
Q_{\pi}(s, a) = E_{\pi}[G_t|s_t = s, a_t = a]
\]

The state-action-value function of the optimal strategy is defined as:

\[
Q^*(s, a) = \max_{\pi} Q_{\pi}(s, a)
\]

In the Deep Reinforcement Learning algorithm, a neural network is used to approximate the value function to find the optimal strategy.

2.2.2 Model Design of DQN. In the experimental system, the monocular camera completes the perception and recognition of the specific target. Therefore, the system takes the received visual field image as the state and obtains the positional information of the specific target through the visual field image.

DQN is a combination of the deep learning algorithm and the Q-learning [10] algorithm that belongs to the reinforcement learning algorithm. The objective function is defined as follows:

\[
L_i(\theta_i) = E[(y_i - Q(s, a, \theta_i))^2]
\]

Where \( y_i = r + \gamma \max_{a'} Q(s', a', \theta_{t-1}) | s, a \).

Updating the parameters of the objective function uses the gradient descent method:

\[
\nabla_{\theta_i} L_i(\theta_i) = E[(y_i - Q(s, a, \theta_i))\nabla_{\theta_i} Q(s, a, \theta_i)]
\]
The system uses 3 discrete actions \{left turn, go forward, right turn\} to ensure that the target is in the center of the robot's field of view. The difficulty of reinforcement learning is the setting of reward function. To solve this problem, we design a setting scheme of the reward function: during the experiment, the target gives a reward signal by judging whether the target deviates from the center of the robot's field of view. Considering the hysteresis of human judgment and the running speed of the robot we can set a time threshold \([T_1, T_2]\). When the target gives a negative reward signal, \(T_r\) is randomly selected and multiplied by the frame rate \(M\) of the camera. The number of frames with negative reward is \(N=T_r \times M\), and the reward function of the remaining frames is positive. (As Figure 3 shows).

2.2.3 Transfer Model of DQN.

Training a randomly initialized deep reinforcement learning model requires a lot of time, but the model transfer technique can transfer the prior knowledge learned by one model to another model, thus shortening the entire training cycle.

In this experiment, the deep reinforcement learning model outputs the value prediction of each action in each state, and the larger the predicted value, the greater the probability of choosing the action. In this paper, we use the VGG network without the Softmax layer as the deep learning part of DQN network, and transfer the parameters obtained from the pre-trained to the DQN model.

The interactive update process between the transfer model and the environment is shown in Figure 4.

3 Experimental results and analysis

3.1 Establishment of pre-trained model

We collected 59,475 data sets through the projection technology designed in this paper. Although only
41,984 data sets were used for training, which was very few for deep learning, the network we designed showed a strong classification effect. In the process of data collection, we added some background with similar colors to clothes worn by human body to enhance the robustness of classification network.

In the experiment, we use ADNets-Model [6], VGG-11[7], VGG-13[7] and VGG-16[7] models respectively to analyze the effect influence of different network depths on the classification of the collected data sets, its accuracy on the test set during the train is shown in Figure 5.

![Figure 5. Accuracy of test set in training process of each model](image)

From Figure 5, we can see that as the number of network layers deepens, the pre-trained network will have a large change in the classification ability of the data set.

Considering speed, we chose the VGG-11 network. To verify the effect of pre-trained on reinforcement learning model, we trained the DQN model with transfer model and random initialization. Figure 6 shows the maximum state-action-value Q of the two models in real world applications after the second interactive update. DQN with pre-trained and transfer model has larger state-action values. This shows that DQN after pre-trained and transfer model has faster convergence speed.

![Figure 6. Comparison between pre-trained model and random initialization model](image)
3.2 Experimental results of robotic following system

After the pre-trained, we apply it to the actual following robot to determine the effect of the pre-trained model on the end-to-end control. It is not difficult to see from Figure 7 that the color of clothes worn by the follower is very similar to the color of the door, but the following robot can well distinguish the follower and make correct movements.

![Figure 7. Direct application of pre-trained model](image)

The transfer model makes reinforcement learning unnecessary to carry out complicated initialization process, greatly shortens the period of reinforcement learning, and enhances the robustness of the following robot to complex scenes, especially light changes. As shown in Figure 8, the four images in [a] show the following effect of the following robot without reinforcement learning. Due to the influence of light on people, the following robot mistakenly took the gray door as the follower and made the wrong action. However, the four images in [b] are the results after two rounds of intensive learning and training. The following robot can overcome the influence of light well and achieve effective following.

![Figure 8. Contrast chart before and after reinforcement learning](image)

4 Conclusion

This paper presents a set of deep reinforcement learning schemes for the robot follower. We designed the classification network to realize the end-to-end control. Through the network comparison experiment of ADNets-Model and VGG with different convolution layers, we have found a more suitable pre-network depth to achieve better classification with less data. Then, we transferred the pre-trained network to the reinforcement learning network DQN and built a complete deep reinforcement learning system suitable for the task of monocular vision following robots. In addition, the system gives reward and punishment signals according to the frame rate of the camera and the lag time of human judgment and designs a reward function suitable for this task. Finally, we constructed an intelligent robot system based on deep reinforcement learning algorithm for following target in the real world.

Acknowledgments

(1) Project 201810145225 supported by National Training Program of Innovation and Entrepreneurship for Undergraduates.
(2) Supported by the Fundamental Research Funds for the Central Universities (N182410001).
References

[1] Yoon Y, Yun W, Yoon H, et al. Real-Time Visual Target Tracking in RGB-D Data for Person-Following Robots[C]// International Conference on Pattern Recognition, 2014.

[2] Ilias B, Nagarajan R, Murugappan M, et al. Hospital nurse following robot: hardware development and sensor integration[J]. International Journal of Medical Engineering and Informatics, 2014, 6(1): 1-10.

[3] Wang N, Yeung D Y. Learning a deep compact image representation for visual tracking[C]// International Conference on Neural Information Processing Systems. 2013.

[4] Ma C, Huang J B, Yang X, et al. Hierarchical Convolutional Features for Visual Tracking[C]// 2015 IEEE International Conference on Computer Vision (ICCV), 2015.

[5] Cui Z, Xiao S, Feng J, et al. Recurrently Target-Attending Tracking[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016:1449-1458.

[6] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013,11(19):1312-5602.

[7] Yun S, Choi J, Yoo Y, et al. Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[8] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014,12(4):1409-1556.

[9] Henriques J F, Caseiro R, Martins P, et al. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.

[10] Mentzer J T, Williams L R. The Role of Logistics Leverage in Marketing Strategy[J]. Journal of Marketing Channels, 2001, 8(3-4): 29-47.