A Novel Active Object Detection Network Based on Historical Scenes and Movements

Nan Ye, Ruogu Wang, and Ning Li

***Abstract*—Active object detection (AOD) aims at getting a better image for detection. It moves robot to get a best view of the object. This method is more effective than traditional object detection when facing following problems: the object in image is of tiny scale, the object is blocked by other irrelevant objects, the object is partially captured by camera, etc. We consider that the past images acquired by the robot are related to the problems mentioned above. So different from the state-of-the-art methods which mostly generate actions based on current image, a novel AOD network based on CNN and LSTM networks is proposed to advance detection performance in this paper. The AOD network uses current and historical scenes as well as movements are learned to explore and generate following robot actions. We train the Action Network through reinforcement learning. During training, phenomenon that robot stuck and repetitive in several specific situations usually occurred, which results in invalid training. To solve this, an effective training strategy is proposed to skip trapped or repetitive actions. Our proposed AOD network was evaluated on Active Vision Dataset and the experiment results showed its advantage of both accuracy and efficiency on AOD tasks.**

***Index Terms*—Active vision, active object detection, reinforcement learning, convolutional network (CNN), long short-term memory (LSTM).**

I. Introduction

Visual object detection aims at finding object in the visual scene, which is a core functionality for robots in daily environment. In recent years, computer vision has made great progress on object detection, especially with deep models, such as [1]-[3]. However, object detection task in the field of robotics is different from traditional object detection. Robots can move around in daily environment and collect a series of images. These images are usually taken under ambiguous intention and have such problems: the object in image is of tiny scale, the object is blocked by other irrelevant objects, the object is partially captured by camera, etc. These problems often lead to performance degradation in object detection task using traditional methods [4]. To solve these problems, a bunch of methods propose to use active vision which aims at actively moving the robot to get a best view of the object.

The early work of AOD focuses on view selection [5], [6]. In recent years, it mainly studies the next best view prediction [7]-[9]. For example, [10] uses entropy increasing of the current and future images to determine the best position at the next moment. This method needs to obtain the possible position at the next moment in advance so that it is difficult to implement in unknown scenarios.

[11] constructs a specific physical model for analysis. However, when the surrounding environment is too complicated, it is difficult and time-consuming to constructs such a model. In order to avoid constructing complex models while ensuring the performance of the method in unknown scenarios, previous work [12], [13] begun to consider deep reinforcement learning and have got better performance. [14] and [15] adopt deep reinforcement learning and build an action decision model based on ResNet-18 to decide the next best view. The model can generate the next action based on the current image.

Fig. 1. The robot moves in the scene, changes the view, and the classifier's score for the object changes. When the object gets high score, it can be well detected.

Though recent work of AOD have achieved better performance through reinforcement learning, we argue that this formulation is not enough. For instance, we observe two problems. The first is that these methods didn't take historical information into consideration. We consider that the robot's movements in the scenes is continuous, and thereby the past actions taken by the robot and the past images acquired by the robot are related to next best view prediction. And the second problem is that the robot would be trapped in one scene or repeat the circle in several scenes, which may lead to invalid training. To solve the first problem, we propose using Conv-LSTM [16] to add historical information to the model inspired by [14]. For the second problem, we improved the training strategy to make the model training more effective.

In this paper, we build our AOD network which can generate a series actions to achieve a best view according to historical actions, historical scenes and current scene. We train the model and evaluate it on the Active Vision Dataset [14], and achieves satisfying results. The main contributions of this paper are in three folds:

Manuscript received December 10, 2020; revised March 3, 2021.

Nan Ye, Ruogu Wang, and Ning Li are with Shanghai Jiao Tong University, Shanghai, 200240, China (e-mail: Nan-yean@sjtu.edu.cn, rgwang@sjtu.edu.cn, ning_li@sjtu.edu.cn).

DOI: 10.7763/IJCTE.2021.V13.1293

- Focus on AOD, we take historical scenes and movements into consideration, and construct an AOD network based on reinforcement learning. The network will move robot to a proper view to solve the bad object detection issues such as tiny scale, occlusion, truncation capture, etc.
- In order to avoid invalid training, we propose an effective training strategy to skip trapping situations and repetitive actions.
- We empirically evaluate our method on the Active Vision Dataset and it achieves state-of-the-art performance.

II. Problem Definition

The main idea of AOD is finding a best view from where the object is easily to be detected. A robot starts in an initial scene I_t (The subscript t indicates the current time t, and the following are the same), receiving an image with a bounding box of the object. However, the initial scene I_t is inadequate for detection because of several problems such as the object is of tiny scale, the object is blocked and the object is partially captured. So the robot needs to choose an action a_t and moves to the next scene I_{t+1} and repeats this process until the detection task is completed. The next scene I_{t+1} is determined by the current scene I_t , and action a_t. So the key is to generate an action policy of action a_t.

In this paper, we use a classifier to decide whether the current view is good for object detection or not. We build an AOD network to generate action policy. Note that our main work is choosing appropriate actions to reach the best view, so the bounding boxes of our target objects are already given, and the object detector is the same in [14].

III. AOD Network

Our AOD network is shown in the Fig. 2. It receives historical scenes $\{I_{t-1}, I_{t-2}, I_{t-3}\}$ and corresponding actions $\{a_{t-1}, a_{t-2}, a_{t-3}\}$, current scene $\{I_t\}$, a bounding box of the object $\{b_t\}$, current action space $\{s_t\}$, and then outputs scores for different actions: forward, backward, left, right, clockwise rotation and counter-clockwise rotation. Then the action with best score is the a_t which we need. Here we use π_* to represent the function mapping of each part of the AOD network. Thus, the AOD network consists of five parts $\{\pi_{lstm}, \pi_{img}, \pi_{box}, \pi_{space}, \pi_{act}\}$. π_{lstm} is an encoder for historical information:

$$Z_t^{lstm} = \pi_{lstm}((I_{t-3}, a_{t-3}) \dots (I_{t-1}, a_{t-1})) \quad (1)$$

It consists of the first nine layers of Resnet-18 and two {5x5 Conv, ReLU, 2x2 MaxPool} blocks [17], followed with LSTM layer. The input I_t is passed to π_{img}, which consists of the first nine layers of Resnet-18 and two convolution modules [14]. And π_{img} gives

$$Z_t^{img} = \pi_{img}(I_t) \quad (2)$$

We also consider the bounding box of object and the current action space:

$$Z_t^{box} = \pi_{box}(b_t) \quad (3)$$

$$Z_t^{space} = \pi_{space}(s_t) \quad (4)$$

Here the bounding box are normalized to have a good performance [15]:

$$\bar{x} = \frac{x}{I_w}, \bar{y} = \frac{y}{I_h}, \bar{w} = \frac{w}{I_w}, \bar{h} = \frac{h}{I_h} \quad (5)$$

where I_w, I_h denote the width and height of image. Finally, we use π_{act} which consists of fully connected layers to generate our action:

$$a_t = \max\{\pi_{act}(Z_t^{lstm}, Z_t^{img}, Z_t^{box}, Z_t^{space})\} \quad (6)$$

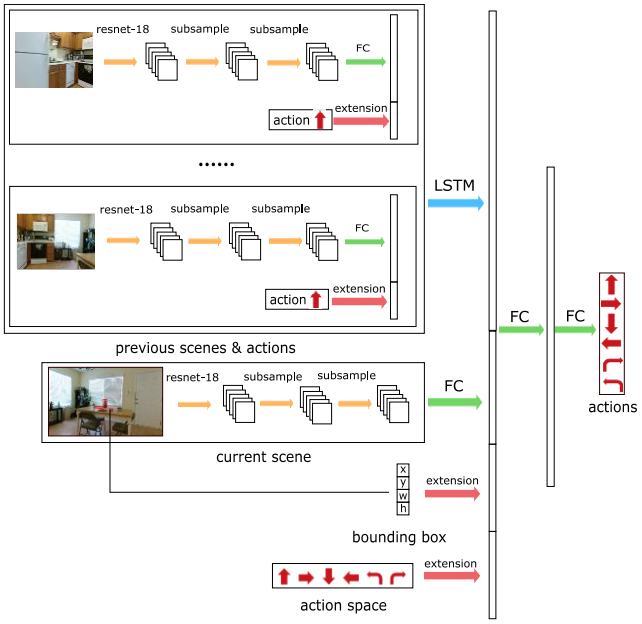


Fig. 2. The AOD network uses historical scenes and actions, current scene, bounding box of the object and action space, then outputs scores for different actions: forward, left, backward, right, clockwise rotation and counter-clockwise rotation.

Resnet-18 has shown impressive performance in image classification, so we use its pre-trained models in [18]. We keep its weights fixed and learn the other weights of $\{\pi_{lstm}, \pi_{img}, \pi_{box}, \pi_{space}, \pi_{act}\}$ based on reinforcement learning. We use the classifier in section II to evaluate whether we choose the good actions to achieve an effective view. Thus, similar to [14], our reward is formulated as:

$$J(\theta) = E_{\pi((\phi(I_{t-3}),a_{t-3})\dots(\phi(I_{t-1}),a_{t-1}),\phi(I_t),b_t,s_t)}[R] \quad (7)$$

where $\Phi(I_{t-1})$ is the output of Resnet-18. And we assume the policy distribution to be independent at each time step. When we need to give the AOD network a positive reward signal, R is the score of the classifier, otherwise R = 0. To learn the policy, we use gradient with reinforcement learning approximate to [19], [20].

However, there remains an initialization problem. For the AOD network, we always consider historical scenes and actions in the last three time steps. Unfortunately, we do not have historical information at the beginning of the task, which leads to input dimension mismatch for LSTM layer. The common method for this is using zero to fill the blank in the input sequence. In this paper, we use another method: when there is a lack of historical information, we always use the information of closest time step to fill it.

IV. Experiments

We perform our method on active vision dataset [14], and then compare our method with several previous methods. The specific experimental detail is presented in this section.

A. Active Vision Dataset

Active vision dataset simulates the movement of a robot in indoor environments with real imagery. The dataset contains 9 independent indoor scenes. For example, you can find kitchen, living room, dining room, etc. in a single scene. It includes 20,000 + RGB-D images and 50,000 + 2D bounding boxes of object instances. The dataset is well organized and can be used to develop and benchmark active vision [14]. A typical robotic movement in the indoor environment simulated by the dataset is shown as Fig. 1.

B. Training

We use the idea of reinforcement learning to train our AOD network, and generate action at each time step.

In the training process, we take 5 steps as a set of movements, thus we set number of total movements T to 5. In each step, we use the classifier to grade the object in the scene. If the object gets a score more than 0.8, we believe that the robot has moved to a good view and the actions generated by the AOD network is effective. Then we will give the network a positive reward signal to encourage it to adopt similar strategies and end the training of this instance. Otherwise, we give the AOD network historical scenes and actions in previous 3 steps and scene, bounding box of the object, action space in current step. And then we generate the action from the AOD network and move the robot to the next position. This process will continue until the object gets more than 0.9 or the time step meets T. If the scores of objects within T time steps are all less than 0.9, and the final score is higher than the initial score, we also give the network a positive reward signal.

In the training process, we find that when the robot is in a position where the action space is limited, the AOD network will generate improper actions. For example, if robot is sandwiched between obstacles, left and right are improper actions. These actions will not change the view, thus the information would not be updated. According to [14], when the robot receives an improper action at time step t, it will stand still. In that case, the robot will use the information of time step t as the information of the next time step t+1. However, this makes robot trapped and induces invalid training. Considering such situation, we evaluate whether the action is legal or not then only perform the legal action. If the action is improper, one of the remaining actions which belong to {forward, left, right, backward, clockwise rotation, counter-clockwise rotation}is selected. This process will continue until the action is legal. And the forward is always the default selection.

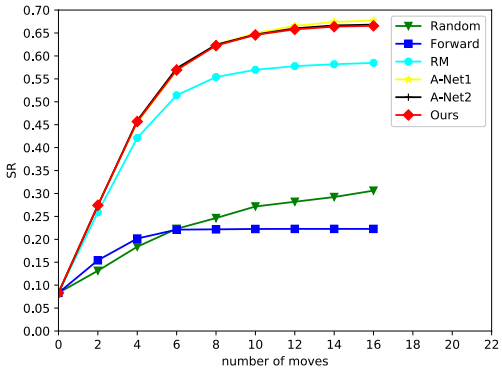
Reaching duplicate positions in the scene is inefficient and should be avoided. Therefore, we will record historical scenes during the training process. At any step within T = 5, if the robot reaches the same scene, the strategy is considered not efficient enough and the training of the current instance would be terminated.

We build three splits for training and testing and perform experiments on the training and testing sets in each split.

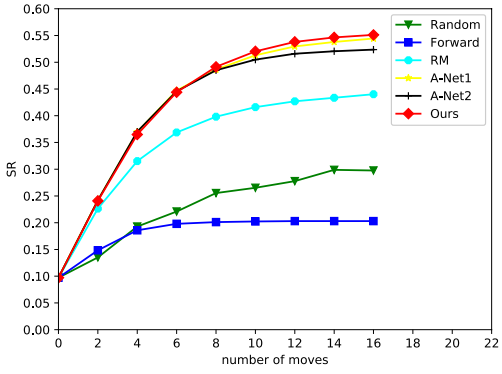
C. Experimental Results and Analysis

In order to evaluate the effectiveness of our method, we compare it with the method in [14] and two baselines. We also explore the impact of different filling methods on the performance of the AOD network. Experiments are set as follows:

- Forward: The robot takes forward action at each time step.
- Random: The robot randomly takes an action from {forward, left, right, backward, clockwise rotation, counter-clockwise rotation} at each time step.
- RM [14]: The policy is provided by the authors of the active vision dataset.
- Ours1(A-Net1): It uses the AOD network in Fig. 2, but using zero to fill the blank in the input sequence when there is a lack of historical information.
- Ours2(A-Net2): It uses the AOD network in Fig. 2 with another filling method: We assume that a series of fictional actions make the robot be at current state. And these actions and corresponding scenes are used as the missing historical information. When we consider these series of actions here, we also prefer forward action.
- Ours3: The method we declare in section III.



(a)known scenes



(b)unknown scenes

Fig. 3. Success rate curves of different methods in known scenes and unknown scenes. It can be observed that success rate improves much more as the number of actions increases using our method, especially in unknown scenes.

We use success rate (SR) and applied path length (APL) [15] to evaluate the efficiency of different methods:

$$SR = \frac{|N_S|}{|N|} \tag{8}$$

where N_s is a collection of successful detection task and N is a collection of all detection task.

$$APL = \frac{\sum_{i \in N} L_i}{|N|} \quad (8)$$

where L_i denotes the path length of each detection task. In our experiment, we regard the detection task to be success when the classifier gets more than 0.8. And we set a limitation to time step T. If the robot does not finish the detection task within time step T, we set its applied path length to T.

We evaluate each method on the test set, as shown in Table I. It can be seen that our method achieves a higher success rate, which is about 10% higher than the RM method. For three different filling methods, our method performs well on Home_003_1, Home_003_2, Home_014_1, and Home_014_2, and the average success rate is the best.

TABLE I: SUCCESS RATE OF DIFFERENT METHODS IN TEST SCANS WHEN TIME STEP T=16

T=16		Random	Forward	RM	Ours1	Ours2	Ours3
Split1	Home_001_1	0.242	0.169	0.323	**0.406**	0.375	0.384
	Home_001_2	0.259	0.145	0.463	**0.695**	0.545	0.628
	Home_008_1	0.289	0.239	0.540	**0.688**	0.666	0.667
Split2	Home_003_1	0.284	0.178	0.379	0.475	0.477	**0.489**
	Home_003_2	0.267	0.163	0.403	0.446	0.459	**0.520**
	Office_001_1	0.249	0.174	0.332	0.332	**0.339**	0.334
Split3	Home_002_1	0.314	0.285	0.563	0.776	**0.789**	0.728
	Home_014_1	0.331	0.207	0.306	0.417	0.449	**0.538**
	Home_014_2	0.444	0.267	0.653	0.665	0.613	**0.672**
avg		0.298	0.203	0.440	0.544	0.526	**0.551**

TABLE II: THE AVERAGE APL AND AVERAGE SCORE OF DIFFERENT METHODS IN TRAINING AND TESTING SCANS WHEN TIME STEP T=10

T=10	Known Scenes		Unknown Scenes	
	APL	Score	APL	Score
Random	8.262	0.9278	8.333	0.9302
Forward	8.135	0.9295	8.531	0.9306
RM	6.252	0.9289	7.220	0.9319
Ours1	5.847	**0.9322**	**6.575**	0.9294
Ours2	5.826	0.9300	6.828	0.9339
Ours3	**5.810**	0.9317	6.828	**0.9350**

We also observe the change of success rate of each method as the number of actions increases. As is shown in Fig. 3(a) and Fig. 3(b), it can be observed that success rate improves much more as the number of actions increases using our method. Regardless of whether it is in a known scene or an unknown scene, the SR grows very fast in the first 8 time steps of AOD network. And it can finally reach a high value. For three different filling methods, AOD network1 performs better in known scenes, while our method performs better in unknown scenes.

In order to evaluate the efficiency and quality of our method, we also observed the APL and score of each method. As shown in Table II, Our method (Ours3) can complete the AOD task with fewer steps while ensuring a high score of the classifier.

Through all the above experiments, we can see that our model and method can achieve better success rate and higher efficiency on AOD tasks.

V. CONCLUSION

In this paper, we use historical information to improve the accuracy and efficiency of AOD task. We construct our AOD network based on CNN and LSTM. The AOD network can generate series of actions to achieve a best view according to historical actions, historical scenes and current scene. We also propose an effective training strategy to avoid trapping situations and repetitive actions. We evaluate AOD network on the Active Vision Dataset, the experiment results show that our method surpasses state-of-the-art methods in both accuracy and efficiency on AOD tasks.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Nan Ye proposed the idea, conducted the research and wrote the paper; Ruogu Wang analyzed the data; Ning Li reviewed the article and supervision of whole research; all authors had approved the final version.

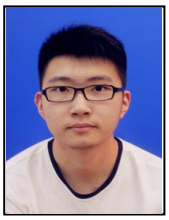
ACKNOWLEDGMENT

This work is supported by National Key R&D Program of China(2018YFB1305902).

REFERENCES

[1] G. Ross, "Fast R-CNN," in *Proc. the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. the IEEE Conference on Computer Vision and Segmentation, Pattern Recognition (CVPR)*, June 2014.
[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[4] M. R. Loghmani, B. Caputo, and M. Vincze, "Recognizing objects in-the-wild: Where do we stand?" in *Proc. 2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2170-2177.
[5] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966-1005, Aug. 1988.
[6] Z. Jia, A. Saxena, and T. Chen, "Robotic object detection: Learning to improve the classifiers using sparse graphs for path planning," in *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, 2011, vol. 22, p. 2072.
[7] J. Yang, Z. Ren, M. Xu, X. Chen, D. J. Crandall, D. Parikh, and D. Batra, "Embodied visual recognition," *Computer Vision and Pattern Recognition*, 2019.
[8] K. Vasiliy, C. Alessandro, and S. Soatto, "Controlled recognition bounds for visual learning and exploration," in *Advances in Neural*

Information Processing Systems, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2915-2923.
[9] V. Javier, H. Garrett, A. S. Huang, P. Ingmar, and R. Nicholas, "Modelling observation correlations for active exploration and robust object detection," *Journal of Artificial Intelligence Research*, vol. 44, 2004.
[10] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T. Kim, "Recovering 6D object pose and predicting next-best-view in the crowd," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[11] N. Atanasov, B. Sankaran, J. Le Ny, T. Koletschka, G. J. Pappas, and K. Daniilidis, "Hypothesis testing framework for active object detection," in *Proc. 2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4216-4222.
[12] H. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *OALib Journal of Computer Science*, 2015.
[13] Y. Sangdoo, C. Jongwon, Y. Youngjoon, Y. Kimin, and J. Young Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[14] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *Proc. 2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1378-1385.
[15] X. Han, H. Liu, F. Sun, and X. Zhang, "Active object detection with multistep action prediction using deep q-network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3723-3731, June 2019.
[16] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4580-4584.
[17] D. Abhishek, D. Samyak, G. Georgia, L. Stefan, P. Devi, and B. Dhruv, "Embodied question answering," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
[18] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A MATLAB-like environment for machine learning," in *Proc. the BigLearn NIPS Workshop*, 2011.
[19] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229-256, May1992.
[20] M. Volodymyr, H. Nicolas, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2204-2212.



Nan Ye received the B.S degree from Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 2018.

He is currently a master with the control science and engineering, in Shanghai Jiao Tong University, Shanghai, China. His research interests are artificial intelligence and computer vision.

Ruogu Wang received the B.S degree from Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 2018.

He is currently a master with the control engineering, in Shanghai Jiao Tong University, Shanghai, China. His research interests are artificial intelligence and object detection.

Ning Li received the B.S. and M.S. degrees from Qingdao University of Science and Technology, Qingdao, China, in 1996 and 1999, respectively; and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2002.

She is currently a professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Her research interests include modeling and control of complex system, artificial intelligence, and big data analysis.