FROM FOOT TO HEAD: ACTIVE FACE FINDING USING DEEP Q-LEARNING

Hui Zhang, Huaping Liu, Di Guo, Fuchun Sun

Department of Computer Science and Technology, Tsinghua University
State Key Lab. of Intelligent Technology and Systems, TNLIST, China

ABSTRACT

In the existing work on active face detection and tracking, it is usually required that the face has to appear in the field-of-view. However, this may not be practical in some challenging scenarios. In this paper, we formulate the problem of active face finding as a Markov Decision Process and resort to the deep Q-learning to solve it in an end-to-end manner. Under the proposed framework, the agent is able to learn how to adjust the control parameters of a camera in order to find the face. Even if the captured image contains only some parts of the person, the PTZ camera can still adjust its pose until the face is found. Extensive experimental validations are performed to show the effectiveness of the developed system.

Index Terms— Active face finding, Deep Q-learning

1. INTRODUCTION

Face detection and recognition are necessary components in modern visual surveillance systems and play important roles in robotics. Traditional face detection systems use the camera to scan the environment according to a predetermined procedure. If some person enters the Field-Of-View(FOV), his face must be detected at first, and then be recognized or tracked.

In many cases where although the person's face does not appear in the FOV, some of other parts, such as the foot, knee, body, even shadow may appear. The goal of this paper is to improve the quality of the surveillance by utilizing such cues to guide the camera to localize the face. In Fig.1 we show a simple illustration of this study. Generally speaking, in face localization problem, the existence of a face is assumed, and the aim is to precisely localize the positions of facial features [1]. Active face detection had been studied in many literatures such as [2] [3] [4] [5] [6], all of which used active camera. [6]

An intuitive solution to this problem is the active perception [7] [8]. Ref. [9] addressed the active object recognition and pose estimation problems for household objects in a highly cluttered environment. Very recently, Ref. [10] developed an information theoretic framework that unified online feature selection and view planning. Since active object recognition is essentially a process of making sequences of decisions, some scholars have proposed to develop the reinforcement learning method to solve it [11] [12] [13] [14] [15] [16] [17].

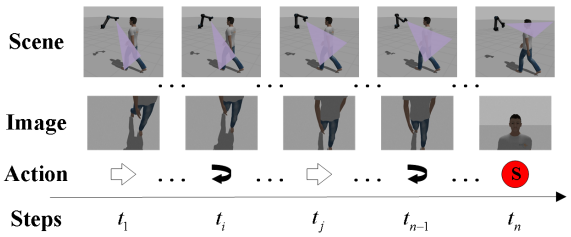


Fig. 1. An illustration of the work. A sequence of actions taken by the proposed algorithm to find a face.

On the other hand, deep learning has shown its great ability in solving sequential decision problems. Most prominent is the recent development of Deep Q-Network (DQN) in Q-learning to solve Atari games [18]. DQN has now been a very hot topic in the fields of machine learning and robotics [19] [20] [21]. Currently, most of this work only performs the experimental validation the offline-collected dataset or simulation environment. The real-world application of DQN is still very challenging.

In this work, we show how to use the reinforcement learning method to develop an end-to-end system for active face finding. Using the proposed system, once any part of the person (such as the foot, knee, or body) enters the FOV, the camera will capture images and automatically adjust its pose to capture the face and then bring the face into the center of the captured image. An illustration of this function is listed in Fig.1. The main contributions are summarized as follows:

1. We establish a new reinforcement learning framework for the active face localization and recognition problem.
2. We utilize the DQN to perform reinforcement learning for the active face finding.
3. We make extensive experimental validations on simu-

This work was supported in part by the National Natural Science Foundation of China under Grant U1613212, Grant 61673238, Grant 91420302, and Grant 61327809, in part by the National High-Tech Research and Development Plan under Grant 2015AA042306, and in part by the National Science and Technology Pillar Program during the Twelfth Five-Year Plan Period under Grant 2015BAK12B03. E-mail: hpliu@tsinghua.edu.cn.

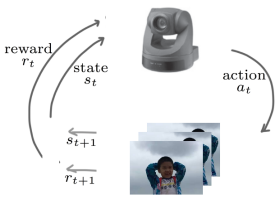


Fig. 2. An illustration about the reinforcement learning.

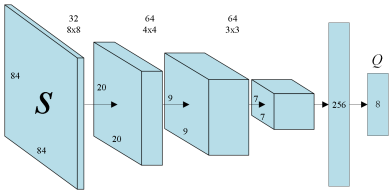


Fig. 3. The architecture for DQN.

lated environments and a PTZ active camera platform.

The rest of this paper is organized as follows: In Section 2, we introduces how to formulate the active face finding as a reinforcement learning problem and Section 3 gives the deep Q-learning algorithm. In Section 4 we give extensive experimental validations.

2. PROBLEM FORMULATION

In this work, we formulate the active face finding as a reinforcement learning problem. In a reinforcement learning setting, a learning agent interacts with an unknown environment in order to maximize the cumulative reward accumulated by adapting its behaviour within the environment.

2.1. State Representation

The agent learns from raw images to control the camera finding the human face actively. We validate our method in both simulation and real world environment. Frames obtained from the simulation and real world are RGB images of 320×240 pixel and 780×480 pixel respectively. In order to reduce the state dimension and the amount of calculation, the RGB images are converted to grayscale images and then rescaled to 84×84 pixel. The grayscale image is the input of the deep Q network and we regard it as the state $\boldsymbol{s}$.

2.2. Reward Function

The reward function $R(\boldsymbol{s}, a)$ is granted to the agent when it chooses the action a to move from state $\boldsymbol{s}$ to the next state. Our goal is to find the face actively and bring the face into the center of the captured image. To this end, we adopt the face detector in OpenCV. It is considered as a success search when a positive reward is returned. In detail, the reward function is defined as

$$R(\boldsymbol{s}, a) = \begin{cases} 1, & \frac{D}{H+W} \leq \tau \\ -1, & out\ of\ the\ range \\ -0.05, & otherwise \end{cases}$$

where H and W are the height and width of image, and D is the distance between cthe enter of the detected face and image. The parameter τ is set to 0.0625 empirically.

If the agent gets a success search or out-of-the-range occurs, we terminate the episode and start a new one. In addition, to avoid looping among some states, we limit the maximum number of searching steps to be 20.

3. DQN ALGORITHM

Under a stationary policy π, the Q-value function for selecting action a in state $\boldsymbol{s}$ is defined as the expected future discounted return:

$$Q^{\pi}(\boldsymbol{s}, a) = \sum_{t=0}^{+\infty} \{\gamma^t \mathcal{R}(\boldsymbol{s}_t, a_t) | \boldsymbol{s}_0 = s, \boldsymbol{a}_0 = a\}$$

where $\gamma \in [0, 1]$ is a discount factor for future rewards. This value function is a measure of the long-term reward obtained by taking action a at state $\boldsymbol{s}$.

The optimal Q-value function can be defined as

$$Q^*(\boldsymbol{s}, a) = \max_{\pi} Q^{\pi}(\boldsymbol{s}, a)$$

At testing stage, after receiving the state $\boldsymbol{s}$, the action can be obtained as

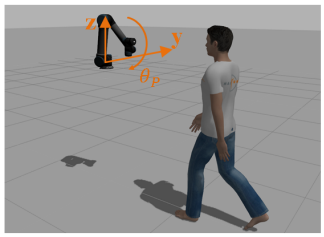
$$a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q^*(\boldsymbol{s}, a)$$

Therefore, the core is to estimate the optimal value function $Q^*(\boldsymbol{s}, a)$, which is the maximum value achievable under any policy.

In high dimensional state spaces, we utilize a neural network which is parameterized by $\boldsymbol{\beta}$ to approximate it as

$$Q^*(\boldsymbol{s}, a) \approx \hat{Q}(\boldsymbol{s}, a; \boldsymbol{\beta}).$$

The architecture of network we used here is shown as Fig.3. The network input is a 84×84 state $\boldsymbol{s}$ obtained from the camera. It is followed by three convolutional layers which are in turn followed by two fully connected layers. Except the last fully connected layer, each layer is applied by a rectifier nonlinearity. The first convolutional layer has 32 8×8 filters with stride 4; the second 64 4×4 filters with stride 2; and the third convolutional layer contains 64 3×3 filters with stride 1. The first fully connected layer has 256 units. The number of the fully connected output layer is equal to the number of actions, 8 actions in our experience.



(a). Simulation Environment

(b). PTZ Camera

Fig. 4. The experimental scenes.

Table 1. Actions in simulation and real world

No.	Simulated Actions	PTZ Camera Actions
1	+0.1 m along Y-axis	$-4\pi/75$ rad of Tilting
2	-0.1 m along Y-axis	$-\pi/75$ rad of Tilting
3	+0.05 m along Z-axis	$\pi/75$ rad of Tilting
4	-0.05 m along Z-axis	$4\pi/75$ rad of Tilting
5	$+\pi/32$ rad around Y-axis	$-4\pi/75$ rad of Panning
6	$+\pi/8$ rad around Y-axis	$4\pi/75$ rad of Panning
7	$-\pi/32$ rad around Y-axis	$\pi/75$ rad of Panning
8	$-\pi/8$ rad around Y-axis	$4\pi/75$ rad of Panning

In [18], the loss function at each iteration used for learning a Deep Q Network is as below:

$$\mathcal{L}(\boldsymbol{\beta}) = \mathop{\mathbb{E}}_{\{\boldsymbol{s},a,r,\boldsymbol{s}'\}\sim\mathcal{M}}[(r+\gamma \mathop{max}_{a'} Q(\boldsymbol{s}', a'; \hat{\boldsymbol{\beta}}) - Q(\boldsymbol{s}, a; \boldsymbol{\beta}))^2]$$

where $\mathcal{M}$ is a uniform probability distribution over a replay memory, which is a set of m previous $(\boldsymbol{s}, a, r, \boldsymbol{s}')$ transition tuples, where m is the size of the memory. Here, $\hat{\boldsymbol{\beta}}$ represents the parameters of a separate target network, while $\boldsymbol{\beta}$ represents the parameters of the online network. The usage of a target network is to improve the stability of the learning updates.

4. EXPERIMENTAL VALIDATIONS

In this section, we test the effectiveness of the developed active face finding method in both simulation and real-world environment. The two experimental scenes are shown in Fig.4.

4.1. Simulation Examples

In the Gazebo simulation environment, we place a 6-DOF robotic arm and a person model. A camera is fixed at the end of the robotic arm. The robotic arm is limited to translate in the direction of Y-axis and Z-axis and rotate around Y-axis. We equip the manipulator with 8 actions, which are listed in Table 1.

To collect the training image samples, we divide uniformly along the y-axis in the interval of $[-0.5, 0.5]$ at a step of 0.05m, along the z-axis in the interval of $[0.1, 0.3]$ at a step of 0.05m and the θ_p is uniformly divided into 65 parts at a step of $\pi/64$. Therefore, there are altogether $21 * 5 * 65 = 6825$ images for training. After training, we test the agent in the Gazebo simulation environment.

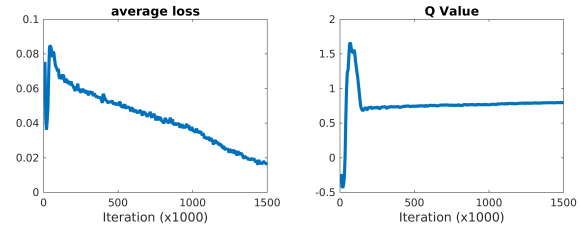


Fig. 5. The loss and Q-value curves during training period.

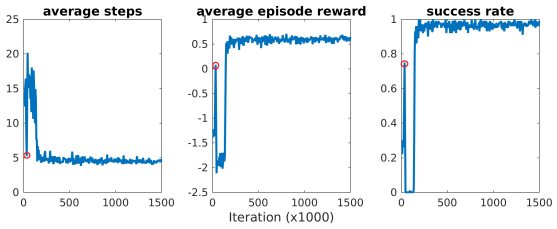


Fig. 6. Testing curves of average steps, average reward and success rate.

The curves of loss function and the Q-value function are illustrated in Fig.5, which shows that a stable Q value can be achieved after sufficient numbers of iterations.

To test the effectiveness of the trained model, we test the models per 5000 iterations. In testing, 100 episodes are performed and the start position is randomly initialized and actions are chosen by the learned Q network. The test are evaluated using the following metric: average steps, average reward, success rate. The results are shown in Fig.6. The average steps reach to a stable interval of [4,6], the steady reward reaches about 0.5 and the success rate reaches about 0.9.

It is hard to analyze Q-values of each action, as the input is of high dimension. We choose 21 states which are taken by translating the robotic arm by 0.05 meters per step along the Y-axis in range of

$$\begin{cases} -0.5 \leq y \leq 0.5 \\ z = 0.2 \\ \theta_p = 0 \end{cases}$$

and calculate the Q-value of each state. 11 representative images among them are shown in the top panel of Fig.7. By this way, we can map the image to its position and analyze the Q-value distribution of two actions along the Y-axis position.

In Fig.7, we can see the Q-values are almost randomly distributed at the initial stage, while the curves becomes more smooth and meaningful when increasing the iteration numbers. The Q values approximate a constant value gradually.

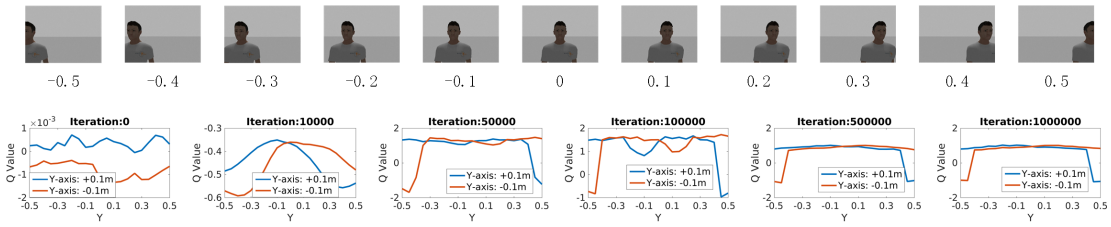


Fig. 7. Q Value of the first and second actions on the 21 states which are obtained 0.05 meters per step alone the Y-axis.

Fig. 8. Example sequences observed by the agent and the actions selected to find the faces.

The right-most panel in Fig.7 shows that the distribution is rather stable and we can find it will choose positive translation when $y \leq 0$ and negative translation otherwise. This illustrates that the deep Q-network indeed learns the correct actions. However, we also notice an interesting phenomenon which occurs at the 10000-th iteration, which was marked as red in Fig.6. Those results also provide very strong discrimination between the two actions. This partially interprets the local peak of the success rate at the early stage in Fig.6 and the local peaks in the curves of loss functions and Q value functions in Fig.5.

4.2. Real-World Applications

For the real-world application, we use a SONY EVI D70P camera for the active finding of the face. An illustrative scene is shown in Fig.4. The 8 actions are listed in Table 1.

For the data collection, we invite 10 volunteers to stand in the laboratory. Two different images are taken of everyone, one for training and the other for testing. The PTZ camera traverses the entire workspace, namely $[-\pi/6, \pi/6]$ in the pan direction at the step of $\pi/75$ rad and $[-\pi/6, 7\pi/16]$ in the tilt direction at the step of $13\pi/8748$ rad. The range of PTZ camera is limited in the range of

$$\begin{cases} -\pi/6 \leq pan \leq \pi/6 \\ -\pi/6 \leq tilt \leq 7\pi/16 \end{cases}$$

So we have $10 \times 26 \times 244 = 63440$ images separately in training and testing dataset.

In Fig.8 we list some representative examples which show the sequential decision process. Each row represents one episode. Among these episodes, the first images always only contain some parts of the person, such as foot(the second line), body(the third line), or leg(the fourth line), traditional active detection will fail to find the face, while our method can successfully find the face and bring it to the center of the images. This shows promising applications of the proposed active learning systems.

5. CONCLUSIONS

In this paper, we formulate the active face finding as a Markov Decision Process and resort to the DQN to solve it in an end-to-end manner. Experimental validations are performed to show the effectiveness of the developed system. This paper has shown the feasibility and applicability of the proposed approach by carrying out two challenging experiments.

Though the results are promising, there still exists many limitations. In the future, we will try to increase number of degree-of-freedom and transfer the developed learning systems to mobile robots for active finding of general objects. We will also extended this method to mullti-modal fusion scenarios [22] [23].

6. REFERENCES

[1] Xuan Zou, Josef Kittler, and Kieron Messer, "Accurate face localisation for faces under active near-ir illumination," *FGR*, pp. 1–6, 2006.

[2] Masakazu Matsugu, Kan Torii, Yoshinori Ito, Tadashi Hayashi, and Tsutomu Osaka, "Face tracking active vision system with saccadic and smooth pursuit," *ROBIO*, pp. 1322–1328, 2006.

[3] Mohammad A. Haque, Kamal Nasrollahi, and Thomas B. Moeslund, "Real-time acquisition of high quality face sequences from an active pan-tilt-zoom camera," *AVSS*, pp. 443–448, 2013.

[4] Ben Benfold Eric Sommerlade and Ian Reid, "Gaze directed camera control for face image acquisition," *ICRA*, pp. 4227–4233, 2011.

[5] Morten Lidegaard, Rasmus F. Larsen, Dirk Kraft, Jeppe B. Jessen, Richard Beck, Thiusius R. Savarimuthu, Claus Gramkow, Ole K. Neckelmann, Jonas Haustad, and Norbert Kruger, "Enhanced 3d face processing using an active vision system," *Computer Vision Theory and Application*, pp. 1–8, 2014.

[6] Stefano Ghidoni, Alberto Pretto, and Emanuele Menegatti, "Cooperative tracking of moving objects and face detection with a dual camera sensor," *ICRA*, pp. 2568–2573, 2010.

[7] Shengyong Chen, Youfu Li, and Ngai Ming Kwok, "Active vision in robotic systems: A survey of recent developments," *IJRR*, pp. 1343–1377, 2008.

[8] Alexander Andreopoulos and John K. Tsotsos, "A computational learning theory of active object recognition under uncertainty," *IJCV*, pp. 95–142, 2013.

[9] Kanzhi Wu, Ravindra Ranasinghe, and Gamini Dissanayake, "Active recognition and pose estimation of household objects in clutter," *ICRA*, pp. 4230–4237, 2015.

[10] Christian Potthast, Andreas Breitenmoser, Fei Sha, and Gaurav S. Sukhatme, "Active multi-view object recognition: A unifying view on online feature selection and view planning," *RAS*, pp. 31–47, 2016.

[11] Kollar T and Roy N, "Trajectory optimization using reinforcement, learning for map exploration," *IJRR*, pp. 1979–1984, 2008.

[12] F Deinzer, C Derichs, H Niemann, and J Denzler, "A framework for actively selecting viewpoints in object recognition," *IJPRAI*, pp. 765–799, 2009.

[13] Lucas Paletta and Axel Pinz, "Active object recognition by view integration and reinforcement learning," *RAS*, pp. 71–86, 2000.

[14] Ahmad Afif Mohd Faudzi and Katsunari Shibata3, "Acquisition of context-based active word recognition by q-learning using a recurrent neural network," *Robot Intelligence Technology and Applications*, pp. 191–200, 2014.

[15] Reinaldo A.C. Bianchi, Arnau Ramisa, and Ramon Lopez de Mantaras, "Automatic selection of object recognition methods using reinforcement learning," *Advances in Machine Learning*, pp. 421–439, 2010.

[16] A. D. Bagdanov, A. Del Bimbo, and F. Pernici, "Acquisition of highresolution images through on-line saccade sequence planning," *VSSN*, pp. 121–129, 2005.

[17] Christian Derichs and Heinrich Niemann, "Handling camera movement constraints in reinforcement learning based active object recognition," *VSSN*, pp. 1–8, 2006.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare1, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, "Human-level control through deep reinforcement learning.," *Nature*, pp. 529–533, 2015.

[19] Mohsen Malmir, Karan Sikka, Deborah Forster, Javier Movellan, and Garrison Cottrell, "Deep q-learning for active recognition of germs: Baseline performance on a standardized dataset for active learning.," *BMVC*, pp. 161–171, 2015.

[20] Juan C. Caicedo and Svetlana Lazebnik, "Active object localization with deep reinforcement learning," *ICCV*, pp. 1–8, 2015.

[21] Fangyi Zhang, Jurgen Leitner, Michael Milford, Ben Upcroft, and Peter Corke, "Towards vision-based deep reinforcement learning for robotic motion control," *arXiv*, pp. 1–8, 2015.

[22] Huaping Liu, Yuanlong Yu, Fuchun Sun, and Jason Gu, "Visual-tactile fusion for object recognition," *IEEE Transactions on Automation Science and Engineering*, 2016.

[23] Huaping Liu, Yupei Wu, Fuchun Sun, Bin Fang, and Di Guo, "Weakly paired multimodal fusion for object recognition," *IEEE Transactions on Automation Science and Engineering*, 2017.