Vol.2019-HCI-185 No.12
Vol.2019-UBI-64 No.12
2019/12/11

Preliminary investigation of using deep reinforcement learning to control a mobile robot for human activity recognition

Teerawat Kumrai[1] Joseph Korpela[1] Takuya Maekawa[1] Yen Yu[2] Ryota Kanai[2]

Abstract: Due to recent advances in robotics technologies, it is becoming feasible for mobile robots to use their sensors to observe daily human activities for the purpose of human activity recognition (HAR) in indoor environments. However, when doing so, the robot will have difficultly observing and recognizing human activities when it is positioned behind the human or some obstacle. Therefore, this work investigates a method for using deep reinforcement learning to control the mobile robot's movement when observing human activities. Our objective is to minimize the movement of the robot (i.e., its energy consumption) while maximizing its human activity recognition accuracy. Moreover, our method introduces a new HAR method based on skeletal and visual features extracted from the robot's captured images.

Keywords: Human activity recognition, Reinforcement learning, Robotics

1. Introduction

Recently, human activity recognition (HAR) using wearable and environmental sensors has been actively studied. For example, many methods have been proposed for using body-worn inertial sensors to recognize simple activities [1], [2], [3], [4]. Moreover, wearable cameras have been used to recognize complex activities in the computer vision research, with examples including the use of a chest-worn camera to recognize location-related events in [5] and a wrist-worn camera to recognize activities involving the use of daily objects in [6]. However, to support HAR, always wearing sensors (e.g., body-worn inertial sensors, cameras) in daily life is not practical. There are many reasons for this, such as rapid battery consumption by the sensors, the physical burden imposed, and so on.

However, mobile robots are becoming more common to use in indoor environments because of advances in robotics technologies, e.g., house-cleaning robots, more complex humanoid robots, and so on. Therefore, using robots with mounted sensors (e.g., camera, microphone) for HAR is becoming feasible [7]. Researchers can now use these robots to perform HAR through a mobile platform, without the need for wearable sensors. Moreover, the ability to recognize human activities allows those robots to provide context-aware services for their residents and also improve their human-robot interactions.

This study focuses on performing HAR using a camera mounted on a mobile robot. When doing so, the robot should be controlled to ensure that its camera captures activities from an appropriate position and orientation while the person moves throughout the indoor environment. In order to maximize activity recognition accuracy, deep reinforcement learning is used to train a neural network to automatically control the actions of a robot. Specifically, we employ a deep Q-network (DQN) [8] to control the robot in order to maximize the recognition accuracy of a HAR neural network while minimizing the distance moved by the robot (i.e., its energy consumption). The HAR neural network performs activity recognition using images captured by the robot's camera, with the confidence from the HAR network also fed to the RL network to facilitate its estimation of Q values for its possible actions.

Additionally, we propose an efficient action space in or-

[1] Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University
[2] Araya Inc.

ⓒ 2019 Information Processing Society of Japan

der to address the slow convergence of the deep Q-network that occurs when using an action space that allows the robot to move freely while performing HAR. Our action space ensures that the robot does not hamper the daily activities of the human and that it keeps its distance from the human.

The research contributions of this study are summarized as follows:

- We propose an architecture consisting of a deep Q-network for controlling robot movement and a secondary HAR network for helping estimate Q values.
- We achieve efficient deep Q-learning for HAR by designing effective action and state spaces and incorporating state-of-the-art RL techniques.
- We create virtual environments for deep Q-learning in order to provide it with the interactive environment needed during training.

In the rest of this paper, we first introduce previous studies on activity recognition. Then, we describe our proposed method for activity recognition using a mobile robot. Finally, we evaluate our method in a virtual environment.

2. Related Work

Several previous studies have proposed methods for human activity recognition in indoor environments through the use of embedded sensors (e.g., RFID, switch sensor) [9], [10]. Human activities have also been recognized using motions, postures, and sounds captured by wearable sensors (e.g., microphones, body-worn accelerometers) [11], [12].

Recently, mobile robots with embedded sensors (e.g., camera, microphones) have been used to recognize human activities. Piyathilaka et al. [13] generated 3D skeleton features from the depth camera of a robot and used those features as the basis for human activity recognition. Vieira et al. [7] applied a Dynamic Bayesian Mixture Model (DBMM) to implement a real-time HAR application for use by a robot.

Moreover, several studies have focused on optimizing the camera's position when monitoring humans in an indoor environment. Schroeter et al. [14] tried to optimize the mobile robot's position by focusing on obstacles and light sources. Kessler et al. [15] proposed a method based on particle swarm optimization to optimize the position of the robot when observing humans. In contrast, our study focuses on the control of a mobile robot to maximize HAR accuracy by using deep reinforcement learning (DRL).

図 1: Example images captured by a robot in our virtual environment

3. Activity Recognition Method

3.1 Preliminaries

In this study, the HoME platform virtual environment [16] is used to evaluate the proposed method. Using this platform, we are able to simulate daily human activities by having a humanoid character perform a variety of activities, with the position of each activity set to a suitable location in the environment. For example, making tea and preparing a meal are performed in the kitchen, sleeping is performed on a bed in the bedroom, and so on. Moreover, walking activity was used to connect when the humanoid character transits from one activity to the next. A virtual mobile robot can also be simulated by controlling the camera position and orientation used to view the virtual environment. Since this study mainly focuses on controlling the movement of the mobile robot during HAR by using reinforcement learning, the task of indoor positioning of the human and robot is simulated using an API provided by the HoME platform.

The humanoid characters are animated using skeletal animation models that were generated using a mocap system. The virtual mobile robot is designed based on a commercially-available humanoid robot (Softbank Pepper) as follows: (i) the camera is mounted on the head of the robot at the height of 1 meter facing forward, (ii) the resolution of the camera is 512 by 512 pixels, (iii) the frame rate is 24 fps, (iv) the movement speed of the robot is 0.83 m/s, and (v) the rotation speed of the robot is 34.26 deg/s. Example images captured by the robot are shown in figure 1.

An overview of the proposed method is shown in figure 2. In RL, an agent learns based on its experiences from exploration and exploitation. At each time t, the agent uses its deep Q-network to determine the next action A_t to take based on its current state S_t. These actions correspond to movements by the robot, with the robot's cam-

ⓒ 2019 Information Processing Society of Japan

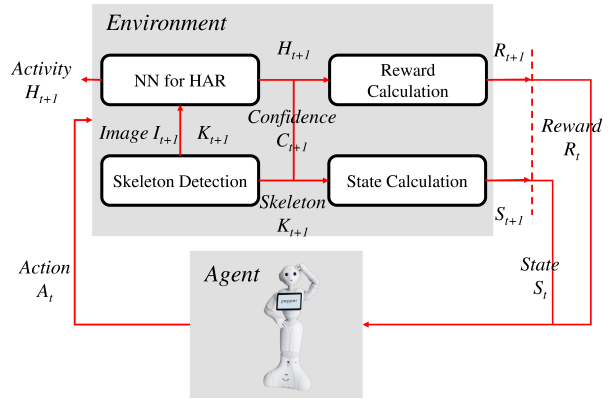


図 2: Overview of proposed method

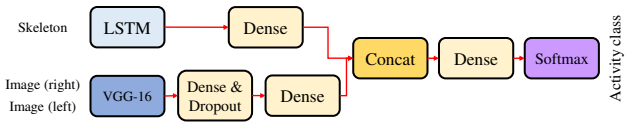
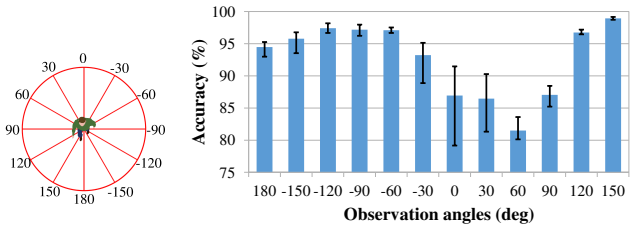


図 3: Architecture of the neural network for HAR



Observation angles　　Macro-averaged F-measures for different angles

図 4: The performance of the HAR NN

era changing position after each action. Images are then captured by the camera at this new position and skeleton information K_{t+1} is extracted from those images using the OpenPose library [17]. This information is used to generate a new state S_{t+1} and is also used by the HAR network to estimate a human activity class H_{t+1}. The portions of the images that correspond to the hands' locations are also fed into the HAR network to capture information about the objects used during the activities. Moreover, the confidence of the HAR network's estimate (C_{t+1}) is output to allow the deep Q-network to estimate a value for the agent's current state.

3.2 Neural Network Used for HAR

Figure 3 shows the structure of the HAR network used in our study. The HAR network is based on a long-short term memory (LSTM) network for time-series analysis [18] and a pre-trained convolutional neural network (CNN) for object recognition (VGG-16 [19]).

The skeleton information from 2-second windows of images and the cropped images from detected hand positions are used as input for the HAR network. The skeleton information is fed into the LSTM layer while the cropped images fed into the CNN layer. Then, the output of the LSTM and CNN layers are concatenated and processed in densely connected layers. Finally, an output layer with H nodes outputs a predicted activity class, where H is the number of activity classes. The rectified linear units (ReLU) function is used as the activation function for the nodes in the LSTM and densely connected layers. We employ the softmax function as the activation function for nodes in the output layer. Additionally, the HAR network outputs the confidence of its estimates, which is used to estimate values for subsequent actions, $C_t = \max_i P(H_i|K_t, K_{t-1}, K_{t-2}, ..., I_t)$, where H_i is the i-th activity class.

The HAR network is trained to minimize the cross-entropy between the distribution of the ground truth and the distribution estimated by the softmax output layer. To enable us to adjust the learning rate, we employ back-propagation using Adam [20] with the pre-trained VGG-16 layers frozen during training.

The performance of the HAR network is shown in figure 4. The results are shown for varying observation angles for the agent (robot). These results show that the macro-averaged F-measure decreases when the agent observes a human from behind.

3.3 Reinforcement Learning for HAR

In general, the agent in RL is mainly seeking a policy that maximizes its expected future rewards. In this study, the agent is trained to find a policy that maximizes its HAR accuracy while minimizing the robot's movement distance. To facilitate this, the confidence of the HAR network's output is computed and incorporated into the agent's current state S_t. The deep Q-network [8] then learns a Q function for a policy π that takes an agent's state and action as input and maps its input to probable future rewards as follows: $Q_\pi(s; a) = E[R_{t+1}|S_t = s; A_t = a]$, where R_{t+1} shows the reward at time $t + 1$. Then, the function is used to select an action that maximizes the expected discounted sum of future rewards [*1] by the agent. The network is updated when training the deep Q-network to correct the difference between its expected reward and the observed reward to adjust its weights as

[*1] A future reward is discounted by using a discount factor $[0, 1]$.

ⓒ 2019 Information Processing Society of Japan

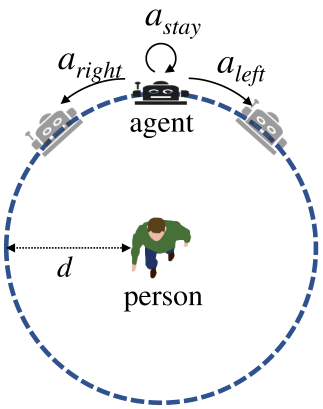
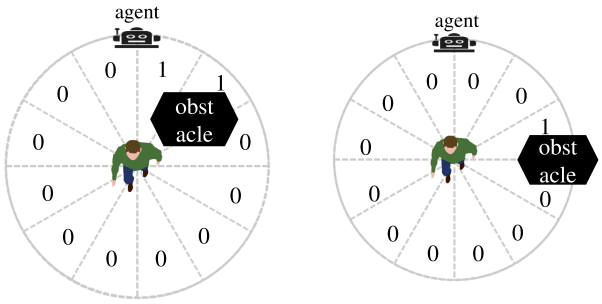


図 5: Action space in the proposed method



Encoding obstacles within the circle

Encoding obstacles on the circumference of the circle

図 6: Examples of obstacle encodings

follows.

$$Q(S_t, A_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(S_t, A_t)}_{old\ value} + \alpha \cdot \underbrace{(R_{t+1} + \gamma \cdot \max_a Q(S_{t+1}, a))}_{learned\ value}, \quad (1)$$

where α shows the learning rate and $\gamma \in [0, 1]$ is a discount factor. Therefore, stochastic gradient descent is used for training the network to minimize the following loss: $(R_{t+1}+\gamma_{t+1} \max_{a'} Q_{\overline{\theta}}(S_{t+1}, a') - Q_\theta(S_t, A_t))^2$, where γ_{t+1} is the discount at $t+1$, θ are the parameters of an online Q-network used for selecting an action, and $\overline{\theta}$ are the parameters of a target network, which is a periodic copy of the online network which is not directly optimized.

3.3.1 Design of the Action Space

Figure 5 shows the movement space available to the deep Q-network. The movement is restricted to the circumference of a circle centered on the person with a radius of d. There are three actions that the agent can take: stay (a_{stay}), go right (a_{right}), and go left (a_{left}). The agent moves along the circumference of the circle for 10 degrees (about 0.35 meters) if the agent selected the a_{right} or a_{left} actions. The deep Q-network will then determine the next action to take after the agent has arrived at its new position.

3.3.2 Design of the State Space

At each time t, the skeleton information K_t, the obstacle information O_t, and the confidence of the HAR network C_t are concatenated as the agent's state S_t. For the skeleton information, K_t, the x- and y-coordinates of each body part is normalized to the range [-1,1]. The obstacle information O_t and the confidence of the HAR network C_t help the deep Q-network how the local environment will affect its ability to perform HAR and to estimate a value for its current state. In this study, we assume that the agent has a floor map, including obstacle information of the environment. Figure 6 shows two examples with the information encoded. The circle is divided into N regions, and the existence of an object in each area is represented using a binary-encoding. The agent is then able to use this information when learning a movement policy that takes into account the possibility of camera occlusion by obstacles.

3.3.3 Design of the Reward

In this study, the reward R_t is computed from the HAR result (H_t); the movement distance of the previous action (d_m), which represents the energy consumed for the previous action; and the travel distance between the positions of the robot and person (d_{rp}) as follows: $R_t = A(H_t) - (e_p \cdot d_m + (d_{rp}/d))$, where d_m is the movement distance of the previous action, e_p is a hyperparameter related to the energy consumption of movement, and $A(H_t)$ indicates whether the HAR output is correct and is computed from the HAR result H_t at time t. $A(H_t)$ equals 1 if the agent can predict the activity correctly and is 0 otherwise. Furthermore, we compute d_{rp} using Dijkstra's algorithm in order to punish situations where the robot is situated in a different room from the person.

3.3.4 Deep Q-network

In this study, three densely connected layers with eight nodes and an output layer with three nodes were used in the deep Q-network. We employ the ReLU function as the activation function of the nodes in the densely connected layers. The number of nodes in the output layer corresponds to the number of actions, with each node in the output layer outputting the class probability of its corresponding action. The optimizer used for training is RMSProp [21]. We introduce the following state-of-the-art RL techniques according to [22] to efficiently train the network. The state-of-the-art RL techniques consist of Categorical DQN [23], Multi-step RL [24], Double DQN [24], Prioritized Experience Replay [25], Dueling Networks

ⓒ 2019 Information Processing Society of Japan

[26], and Noisy Nets [27].

3.3.5 Obstacle Avoidance

Here we introduce our method for obstacle avoidance. Assume that the deep Q-network outputs a_left, but there is an obstacle on the circumference of the circle on the left side of the robot, preventing the robot from moving to the left. In this case, the robot will pause control by the RL process and then detour around the obstacle to a new position on the circumference of the circle. Finally, the robot will restart control by the RL process. If there is no unobstructed position beyond the obstacle on the circumference of the circle, the robot will simply ignore the action.

4. Evaluation

4.1 Data Set and Environments

We used the virtual home environment, as described in Section 3.1, with humanoid characters to evaluate our method. The MakeHuman[*2] toolkit was used to create humanoid characters based on the three participants in our experiment: participant A (Height: 168 cm., Age: 30s, Sex: M), participant B (Height: 173 cm., Age: 20s, Sex: M), and participant C (Height: 162 cm., Age: 30s, Sex: F). The movements (human activities) of the humanoid characters were generated based on data collected by a Perceptron Neuron mocap system[*3], with mocap data collected for 25 body parts from each participant (e.g., head, left/right shoulder, left/right hand, etc.) while they performed the 13 activities used in our evaluation. The 13 activities consist of '**Preparing Meal**' (with objects: pan, spatula, knife, and vegetable), '**Making Tea**' (with objects: can of tea, teacup, and kettle), '**Making Juice**' (with object: blender), '**Washing Dishes**' (with objects: dish, and sponge), '**Reading Book**' (with object: book), '**Using Smartphone**' (with object: smartphone), '**Talking on Smartphone**' (with object: smartphone), '**Eating Meal**' (with objects: knife, and fork), '**Watching TV**' (with object: remote control), '**Brushing Teeth**' (with object: toothbrush), '**Washing Face**', '**Sleeping**', and '**Walking**'. Each participant conducted 10 sessions of these activities in either an actual home environment or in our laboratory. During each session, they performed the 13 activities in an arbitrary order, with each session containing instances of each activity.

Furthermore, we animated the three humanoid characters in three separate virtual home environments, shown in figure 7. Each humanoid character performed activities in their own corresponding virtual environment and each activity was conducted at a suitable location in the environment; for example, making tea, preparing meals, and making juice were performed in the kitchen. The average duration of each activity was about 45 seconds. The length of one virtual session was about 15 minutes. Note that the walking activity was used to connect different locations of each activity throughout the environment.

Environment A

Environment B

Environment C

図 7: Virtual home environments used in this study

4.2 Evaluation Methodology

Each participant's 10 sessions of data were divided into

*2 http://www.makehumancommunity.org/
*3 https://neuronmocap.com/

ⓒ 2019 Information Processing Society of Japan

a set of training data and a set of testing data, with 5 sessions assigned to each set. We generated the training data used for the HAR neural network by animating the humanoid characters using the corresponding participant's 5 sessions of training data and then recording their activities using the virtual mobile robot's camera in a virtual environment from 12 different angles. The deep Q-network was trained separately for each environment using the corresponding participant's 5 sessions of training data to animate the humanoid character in that environment. Note that, we randomized the order of activities for each training iteration in the deep Q-network.

Furthermore, we prepared four methods to evaluate the effectiveness of the proposed method:

- Proposed: The proposed method.
- Naive: This method does not employ RL. In this method, the virtual robot simply follows the person and captures images of the person with the person centered in the images. However, the robot does still maintain a distance d from the person.
- NaiveAct: This method employs RL with a more complex action space than that used by the proposed method. In this method, the robot is allowed to move freely (forward, backward, left, and right in increments of 0.35 meters; stay; and rotate left/right in increments of 10 degrees) while maintaining a minimum distance d.
- DQN: This method employs RL, but does not use the six state-of-the-art RL techniques mentioned in Section 3.3.4.

4.3 Results

4.3.1 Recognition Accuracy and Reward

Comparisons of reward curves among three of the methods (Proposed, NaiveAct, and DQN) are shown in figure 8. These results show that in many cases the reward of Proposed increases earlier than for the other methods.

Comparisons of the evolution of training F-measures for HAR for each method are shown in figure 9. These results show that the transitions of the F-measures are unstable. This may be due to the agents performing exploration and exploitation.

Comparisons of the evolution of the robot's movement distances when performing HAR for each method are shown in figure 10. These results show that the movement distances of the Proposed decrease earlier than those of the other methods.

The average movement distance for each activity and the macro-averaged F-measures for HAR for each of the methods in each of the environments during testing are shown in figure 11 and figure 12, respectively. The results in figure 12 show that Proposed achieved the highest overall HAR accuracy. Additionally, since Naive does not use reinforcement learning to control the robot in order to improve its view of the activities, the HAR accuracies for Naive are poor.

The macro-averaged F-measures for HAR for each activity for Proposed, Naive, and NaiveAct are shown in figure 13. Here we can see that the activities "making tea" and "washing dishes" both have substantially lower F-measures than the other activities. We found that these two activities were often confused with each other, which may be in part due to the robot having difficulty capturing images of the associated objects due to the sink and kitchen counter.

4.3.2 Effectiveness of Action Space

NaiveAct, which allows the robot to move freely, had lower F-measures for HAR than Proposed, as shown in figure 12. There are seven choices of actions for the RL network in NaiveAct. Therefore, due to the increase in choices, the RL network likely required more training episodes than Proposed. Moreover, the average movement distance for NaiveAct in all the environments was longer than that of Proposed, as shown in figure 14.

5. Conclusion

This study proposed a new method for using images captured by a mobile robot's camera to conduct human activity recognition in the home environment, with our method employing DRL to control the robot's movement. The objective of our method is to maximize activity recognition accuracy while minimizing the movement distances (i.e., energy consumption). We evaluated the proposed method using virtual home environments, with the results confirming the effectiveness of our method.

6. Acknowledgments

This work is partially supported by JST CREST JPMJCR15E2.

参考文献

[1] Bao, L. and Intille, S. S.: Activity recognition from user-annotated acceleration data, *Pervasive 2004*, pp. 1–17 (2004).
[2] Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., Millán, J. d. R. and Roggen, D.: The Opportunity challenge: A benchmark database for on-

ⓒ 2019 Information Processing Society of Japan

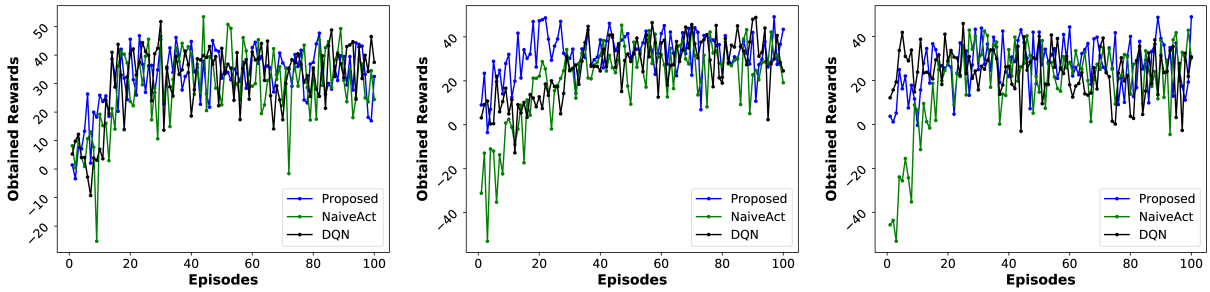


図 8: Comparisons between proposed and the other methods in terms of reward curves plotted for 100 episodes of training

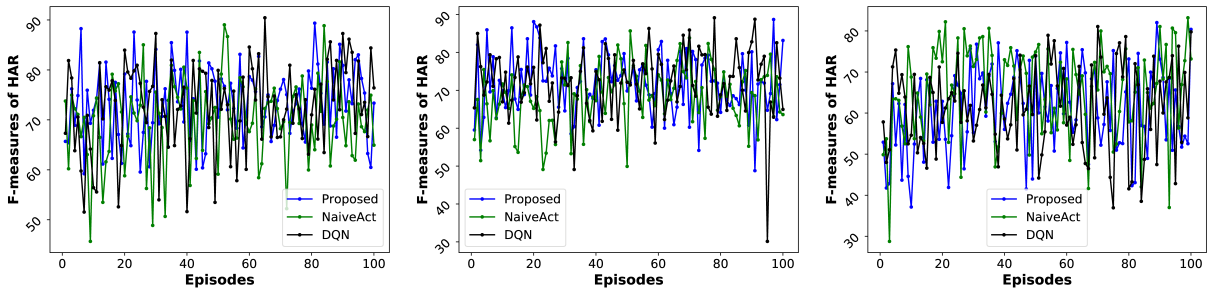


図 9: Comparisons between proposed and the other methods in terms of the evolution of training F-measures for HAR [%] over 100 episodes of training

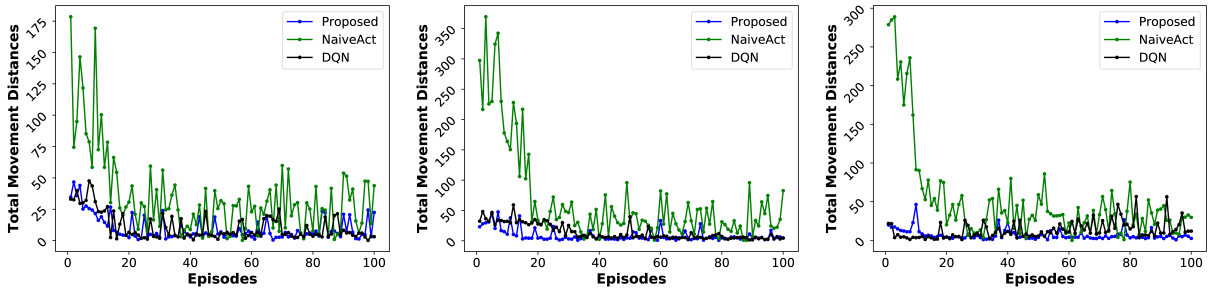


図 10: Comparisons between proposed and the other methods in terms of the evolution of total movement distances [m] over 100 episodes of training

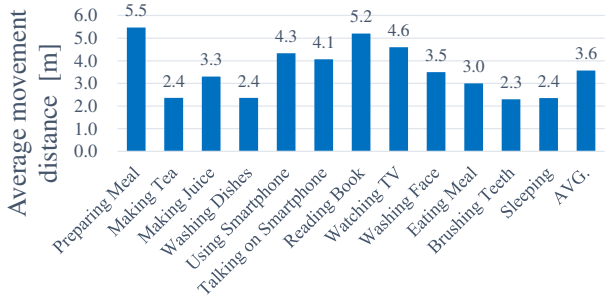


図 11: Average movement distance by Proposed for each activity class during testing

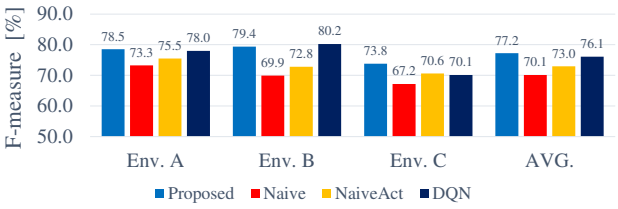


図 12: Macro-averaged F-measures for HAR during testing

body sensor-based activity recognition, *Pattern Recognition Letters*, Vol. 34, No. 15, pp. 2033–2042 (2013).

[3] Korpela, J., Takase, K., Hirashima, T., Maekawa, T., Eberle, J., Chakraborty, D. and Aberer, K.: An energy-aware method for the joint recognition of activities and gestures using wearable sensors, *International Symposium on Wearable Computers (ISWC 2015)*, pp. 101–108 (2015).

[4] Maekawa, T. and Watanabe, S.: Unsupervised activity recognition with user's physical characteristics data, *International Symposium on Wearable Computers (ISWC*

ⓒ 2019 Information Processing Society of Japan

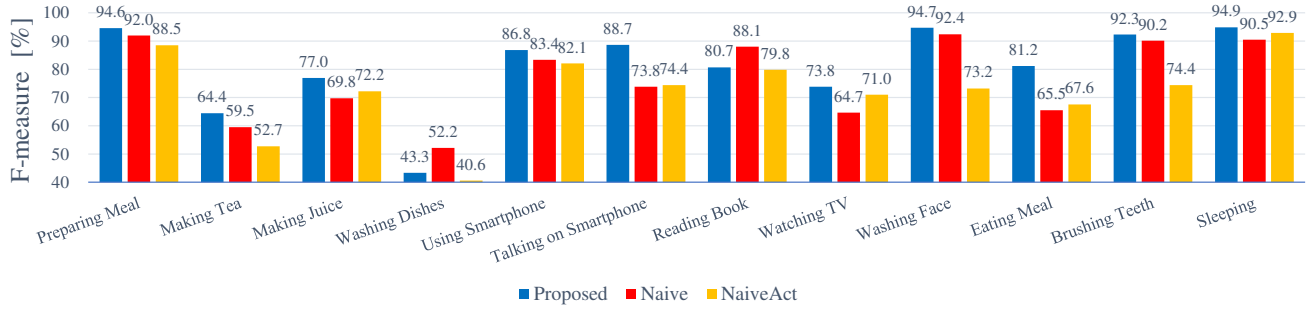


図 13: Macro-averaged F-measure for each activity class during HAR

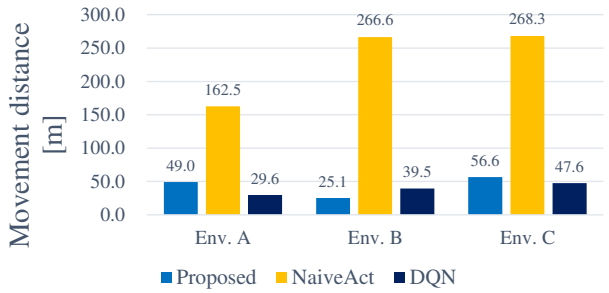


図 14: Average movement distances for each environment

2011), pp. 89–96 (2011).

[5] Clarkson, B., Pentland, A. and Mase, K.: Recognizing user context via wearable sensors, *Int'l Symp. on Wearable Computers (ISWC 2000)*, pp. 69–75 (2000).

[6] Maekawa, T., Yanagisawa, Y., Kishino, Y., Ishiguro, K., Kamei, K., Sakurai, Y. and Okadome, T.: Object-based activity recognition with heterogeneous sensors on wrist, *Pervasive 2010*, pp. 246–264 (2010).

[7] Vieira, M., Faria, D. R. and Nunes, U.: Real-time application for monitoring human daily activity and risk situations in robot-assisted living, *Robot 2015: Second Iberian Robotics Conference*, Springer, pp. 449–461 (2016).

[8] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, p. 529 (2015).

[9] Tapia, E. M., Intille, S. S. and Larson, K.: Portable wireless sensors for object usage sensing in the home: Challenges and practicalities, *European Conference on Ambient Intelligence*, Springer, pp. 19–37 (2007).

[10] Van Kasteren, T., Noulas, A., Englebienne, G. and Kröse, B.: Accurate activity recognition in a home setting, *the 10th International Conference on Ubiquitous Computing (Ubicomp 2008)*, pp. 1–9 (2008).

[11] Blum, M., Pentland, A. S. and Tröster, G.: Insense: Interest-based life logging, *IEEE Multimedia*, Vol. 13, No. 4, pp. 40–48 (2006).

[12] Lester, J., Choudhury, T. and Borriello, G.: A practical approach to recognizing physical activities, *Pervasive 2006*, pp. 1–16 (2006).

[13] Piyathilaka, L. and Kodagoda, S.: Human activity recognition for domestic robots, *Field and Service Robotics*, Springer, pp. 395–408 (2015).

[14] Schroeter, C., Hoechemer, M., Mueller, S. and Gross, H.-M.: Autonomous robot cameraman-observation pose optimization for a mobile service robot in indoor living space, *2009 IEEE International Conference on Robotics and Automation*, IEEE, pp. 424–429 (2009).

[15] Kessler, J., Schmidt, M., Helsper, S. and Gross, H.-M.: I'm still watching you: Update on observing a person in a home environment, *2013 European Conference on Mobile Robots*, IEEE, pp. 300–306 (2013).

[16] Brodeur, S., Perez, E., Anand, A., Golemo, F., Celotti, L., Strub, F., Rouat, J., Larochelle, H. and Courville, A.: HoME: A household multimodal environment, *arXiv preprint arXiv:1711.11017* (2017).

[17] Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E. and Sheikh, Y.: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, *arXiv preprint arXiv:1812.08008* (2018).

[18] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).

[19] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* (2014).

[20] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).

[21] Tieleman, T. and Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural networks for machine learning*, Vol. 4, No. 2, pp. 26–31 (2012).

[22] Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. and Silver, D.: Rainbow: Combining improvements in deep reinforcement learning, *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 3215–3222 (2018).

[23] Bellemare, M. G., Dabney, W. and Munos, R.: A distributional perspective on reinforcement learning, *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, pp. 449–458 (2017).

[24] Van Hasselt, H., Guez, A. and Silver, D.: Deep Reinforcement Learning with Double Q-Learning, *AAAI*, Vol. 2, pp. 2094–2100 (2016).

[25] Schaul, T., Quan, J., Antonoglou, I. and Silver, D.: Prioritized experience replay, *arXiv preprint arXiv:1511.05952* (2015).

[26] Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M. and De Freitas, N.: Dueling network architectures for deep reinforcement learning, *arXiv preprint arXiv:1511.06581* (2015).

[27] Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O. et al.: Noisy networks for exploration, *arXiv preprint arXiv:1706.10295* (2017).

ⓒ 2019 Information Processing Society of Japan