Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Cloud-assisted cognition adaptation for service robots in changing home environments*

Qi WANG[1], Zhen FAN[1], Weihua SHENG[2], Senlin ZHANG[1], Meiqin LIU‡[1,3]

[1]*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*

[2]*School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA*

[3]*Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China*

E-mail: wang9562@zju.edu.cn; fanzhen@zju.edu.cn; weihua.sheng@okstate.edu;
slzhang@zju.edu.cn; liumeiqin@zju.edu.cn

Received Aug. 26, 2020; Revision accepted Jan. 10, 2021; Crosschecked Jan. 7, 2022

Abstract: Robots need more intelligence to complete cognitive tasks in home environments. In this paper, we present a new cloud-assisted cognition adaptation mechanism for home service robots, which learns new knowledge from other robots. In this mechanism, a change detection approach is implemented in the robot to detect changes in the user's home environment and trigger the adaptation procedure that adapts the robot's local customized model to the environmental changes, while the adaptation is achieved by transferring knowledge from the global cloud model to the local model through model fusion. First, three different model fusion methods are proposed to carry out the adaptation procedure, and two key factors of the fusion methods are emphasized. Second, the most suitable model fusion method and its settings for the cloud–robot knowledge transfer are determined. Third, we carry out a case study of learning in a changing home environment, and the experimental results verify the efficiency and effectiveness of our solutions. The experimental results lead us to propose an empirical guideline of model fusion for the cloud–robot knowledge transfer.

Key words: Home service robot; Cloud–robot knowledge transfer; Model fusion
https://doi.org/10.1631/FITEE.2000431 **CLC number:** TP242.6

1 Introduction

The world's elderly population is steadily increasing. It is estimated that by the year 2050, one in six persons in the world will be older than 65 years of age (United Nations, 2020). Moreover, most of the older adults prefer to remain living in their homes because of the greater privacy and autonomy offered by one's own home (Secker et al., 2003), which calls for more home service robots for elderly care. Although we have witnessed many research prototypes of home service robots developed by various groups (Graf and Staab, 2009; Fachantidis et al., 2018; Shuai and Chen, 2019), they are not yet ready for mass deployment into real homes due to the lack of robot intelligence that can deal with a changing home environment.

In Wang et al. (2021), we proposed the concept of "self-evolving home service robot," which allows continuous growth of intelligence in the robot. In that work, the provider or manufacturer of the home service robot maintains an initial deep neural network (DNN) (LeCun et al., 2015) model trained on a predefined dataset, which serves as the robot's initial local model, while each robot adapts its initial

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. U21A20485 and 62088102), the Natural Science Foundation of China-Shenzhen Basic Research Center Project (No. U1713216), and the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT20026)

ORCID: Qi WANG, https://orcid.org/0000-0003-4231-860X; Meiqin LIU, https://orcid.org/0000-0003-0693-6574

local model into a customized model that is fit for the initial environment of the user's home with the help of manual labeling. Apparently, such a customized model has been adapted only to the initial home environment and may not be able to fit a changing home environment, since, after a certain time, the user's home environment may deviate significantly from the initial home environment. One or more object classes in the user's home environment may change due to the changes in the user's preferences or habits, which will cause a degradation in the classification accuracy of the local customized model (Ditzler et al., 2015). For example, imagining a scenario in which the user changes his/her pet dog from a Dalmatian to a Chihuahua, although both Dalmatian and Chihuahua belong to dogs, Chihuahua is closer to cats in size and is more likely to be misclassified as a cat. Such a phenomenon can be interpreted from the probabilistic perspective as the probability distribution of the data shifts from one part of the sample space to the other, which is called concept drift (CD) (Žliobaitė et al., 2016).

One approach is to use the idea of the former adaptation mechanism (Wang et al., 2021) and adapt the customized model to the changes in a changing home environment with the help of manual labeling; however, this is not the best solution for two main reasons. First, the changes in a changing home environment are unlimited (Žliobaitė, 2010). Continuous requests for manual labeling may overwhelm the user, which is undesirable. Second, data generated by the probability distribution after changes may be similar to the data from the environments of other users' homes. The robots that have adapted to the initial environments of the other users' homes may have learned some knowledge about the new probability distribution. It is obviously more efficient to transfer the relevant knowledge learned by other robots to the current robot than learning environmental changes from scratch.

To facilitate adaptation to environmental changes, achieving knowledge transfer between robots is crucial. Recently, the emergence of cloud robotics has made it possible to achieve knowledge transfer between different robots. A collection of robots can easily share data and knowledge through the cloud (Gouveia et al., 2015). Meanwhile, it is better to accomplish knowledge transfer between robots by sharing knowledge through a global cloud model built by the data collected from robots rather than sharing data, since directly transmitting the raw data collected by one robot to the other robots will cause privacy concern. However, since the local customized model contains knowledge learned from the initial home environment, simply replacing the local model with the cloud model may not be the best approach for learning from other robots.

In this study, we introduce a cloud-assisted robot cognition adaptation mechanism that adapts the robot's customized model to changes in a home environment with the help of other robots. The proposed adaptation mechanism relies on an active CD detection method that detects the changes in the user's home environment. The CD detection is based on a continuous comparison between the historical and the current probability distributions. Once CD is detected, a model fusion method is developed to transfer knowledge from the global cloud model to the local customized model, which draws inspiration from the feature transferability of the convolutional neural network (CNN) (Yosinski et al., 2014).

This study has three major contributions. The first contribution is that a novel cloud-assisted robot cognition adaptation mechanism is proposed. The adaptation procedure does not require labeled data from the changing home environment, which is more realistic for robot applications. The second contribution is that we develop several new model fusion methods, in which only the higher layers' specific parameters need to be downloaded from the cloud, while the lower layers' general parameters can be inherited from the local customized model, thus reducing the communication cost. The third contribution is that our work offers guidance for the cloud–robot knowledge transfer process in the robot cognition adaptation mechanism and demonstrates that other robots can provide much-needed new knowledge for the current robot through the cloud.

2 Related works

2.1 Cloud robotics

The concept of "cloud robotics" (Siciliano and Khatib, 2016) was first proposed by Dr. Kuffner of Carnegie Mellon University, USA. In a cloud robotic system, robots that carry out complex tasks in the physical space are seamlessly connected to a cloud

that serves as a back-end support center in the digital space through the network (Huaimin et al., 2018). Using the abundant computing power and storage resources provided by the cloud, cloud robots can obtain capabilities that traditional robots cannot achieve. The cloud robots can hand over complex computational tasks to the cloud, receive massive data from the cloud, and share information and skills through the cloud. In this way, the cloud robot itself does not need to possess supercomputing power or store all the information, but just connects to the cloud to obtain the task-required knowledge when needed. Collaboration can also be achieved between multiple cloud robots.

The RoboEarth (Zweigle et al., 2009) project is an early exploration concept in the field of cloud robotics. The main idea of this project is to provide a World Wide Web platform for robots to share knowledge and skills. Multiple robots can share information about objects, environments, and tasks with each other on this platform. As the cloud engine of RoboEarth, the Rapyuta (Hunziker et al., 2013) project provides an open-source platform-as-a-service (PaaS) framework, which enables the cloud robot to upload computationally intensive tasks to a secure customized computing environment in the cloud for completion. The Distributed Agents with Collective intelligence (DAvinCi) (Arumugam et al., 2010) project uses the computing power of the cloud to achieve the simultaneous localization and mapping (SLAM) of multiple robots in a large indoor environment, which greatly improves the realization speed of various complex computational tasks in the robotic SLAM process.

In reviewing previous cloud robotics related research, we find that all the projects were aimed at designing a new software platform that facilitates the implementation of the cloud robotic concept on the multi-robot mapping task. Studies on using the idea of cloud robotics to achieve the robot's cognition adaptation to a changing home environment is currently lacking. Meanwhile, the much-needed changes-related knowledge is more efficient to be directly transferred from other robots to the current robot than learning from scratch, as we mentioned in the previous section. Therefore, it is very suitable to use the cloud to serve as an intermediary that allows knowledge sharing between robots.

2.2 CD solutions

In recent years, researchers have developed several solutions to adapt the CNN model to CD. Disabato and Roveri (2019) introduced an adaptive mechanism for learning CNNs that are able to operate in the presence of CD. They also followed an active approach and used an active CD detection method to trigger the subsequent adaptation procedure. In their adaptation procedure, only part of the knowledge from the previous model is transferred to the current model, while the remainder of the current model is randomly initialized and retrained using labeled data generated by the probability distribution after changes. However, both of their CD detection and adaptation procedures assume a "test-then-train" scenario (Ditzler et al., 2015), in which the correct labels of the image generated by the probability distribution after changes are available, which rarely holds considering the workload of human labeling. Moreover, totally relying on the data obtained from the probability distribution after changes to retrain the model is neither accurate nor efficient due to the limited retraining data and retraining time.

Different from the above approaches in which one client carries out its adaptation independently, Yoon et al. (2020) studied the scenario wherein each client learns on a sequence of tasks from private local data with the help of tasks learned by other clients through the cloud, and proposed a federated continual learning framework called Fed-APC. In this framework, the neural network weights are decomposed into global task-shared parameters and sparse task-adaptive parameters, which represent the global and generic knowledge across all clients and the specific knowledge for each task, respectively. By selectively updating the global task-shared parameters and selectively using the sparse task-adaptive parameters from other clients, their method can minimize interclient interference while allowing interclient knowledge transfer at the same time to accelerate the learning of the current task. Yet, each specific task of clients needs to maintain the corresponding task-adaptive parameter in the cloud, and the attention of each task-adaptive parameter and the task-shared parameters must be updated once a new task arrives, which causes scalability problems since the numbers of clients and tasks may be very large.

In reviewing previous research, we notice that both methods require new labeled training data to update the model once a new task is detected. An experimental study of the idea of directly merging the local model with the global model to adapt to the environmental changes without retraining is currently lacking. Moreover, existing research (Yosinski et al., 2014) about the feature transferability of CNN shows that it is possible to achieve knowledge transfer between the two models. Therefore, it is both desirable and feasible to develop a cloud-assisted robot cognition adaptation mechanism based on the cloud–robot knowledge transfer.

3 Methodology

In this section, we introduce the overall concept of the proposed cloud-assisted robot cognition adaptation mechanism and discuss the detailed methodology of this mechanism, including cloud model building, CD detection, and cloud–robot knowledge transfer.

3.1 Overall concept

The proposed cognition adaptation mechanism involves the use of knowledge from other robots through the cloud to adapt the local customized model to the changes in the home environment. We present the conceptual design of the cloud-assisted robot cognition adaptation mechanism, as shown in Fig. 1, which consists mainly of three parts: (1) a global cloud model, which is built using the training data collected from a collection of robots; (2) a CD detection procedure, which detects data changes in the user's home environment to trigger the cloud–robot knowledge transfer procedure; (3) a cloud–robot knowledge transfer procedure that transfers the new knowledge from the global model to the local model through model fusion.

3.2 Cloud model building

As an intermediary that allows knowledge sharing between robots, the global cloud model is trained using a cloud dataset, which is collected from the local dataset of a collection of robots to ensure that it contains all the knowledge learned by the robots. Only a small portion of the robot's local dataset is uploaded to the cloud since uploading the entire local dataset will cause a heavy communication burden on the robot. Such a data-uploading procedure can be implemented in parallel with the robot's adaptation to its initial home environment; that is, a small fraction of the manually labeled data will be sent to the cloud once the manual labeling process is completed.

Different from our previous work (Wang et al., 2021), where we used an incremental learning approach to achieve the robot's cognition adaptation to unknown classes, in this study, we focus on a fixed-class scenario, wherein the target classes of the classification task acrossing all the robots are the same, since adding new classes to the classification task will result in differences between the robots' model

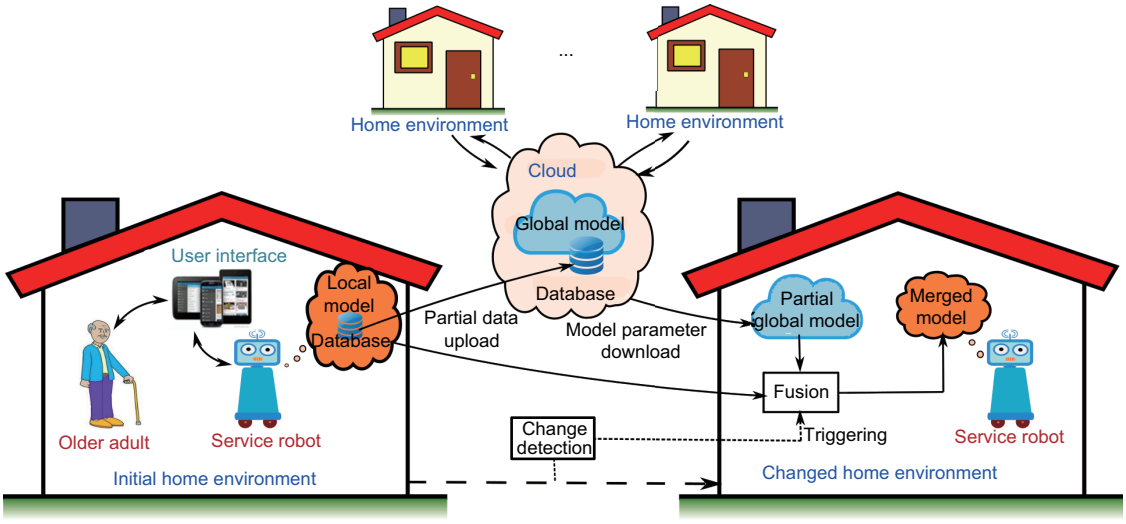

Fig. 1 Concept of the cloud-assisted robot cognition adaptation mechanism

structures and cause obstacles for the subsequent cloud–robot knowledge transfer procedure. The training of the global cloud model and each robot's local customized model is achieved by the fine-tuning approach starting from the same pretrained model with the cloud dataset and the robot's local dataset, respectively.

Note that although the federated learning method, e.g., the FedAvg algorithm (McMahan et al., 2017), can also allow many robots to collectively build such a global cloud model, we do not use the federated learning method in our cognition adaptation mechanism because this method will cause local models of all the robots to become the same as the global model, which means that the robot's local model will lose its customization characteristics.

3.3 CD detection

As one of the key components of the whole adaptation mechanism, CD detection procedure acts as an indicator of the changes in the user's home environment to trigger the subsequent cloud–robot knowledge transfer procedure. A false-positive detection will lead to an unnecessary adaptation and waste precious computation resources of the robot, while a false-negative detection will induce degradation in the classification accuracy since the local model is not adapted to the environmental changes.

Generally, CD detection methods are classified into three categories, i.e., sequential analysis based, data distribution based, and learner output based detection methods (Yang et al., 2020). Sequential analysis based detection methods (Page, 1954) analyze the newly acquired data one by one until the probability of observing the data sequence under a new distribution is significantly larger than that under the original distribution. Data distribution based detection methods (Kuncheva, 2013) typically consider two distributions of data features from a fixed window containing historical information and a sliding window containing the most recent data. Meanwhile, hypothesis tests are used to compare data distributions in the two windows and detect changes when they are significantly different. Learner output based detection methods (Gama et al., 2004; Baena-García et al., 2006) track the learner's error rate fluctuations and detect changes when the error rate exceeds a threshold. In many home service robot applications, it is hard to obtain the data labels due to the workload of labeling. Therefore, it is difficult to obtain the accuracy/error rate. Moreover, sequential analysis based methods can easily cause a detection delay since the detection results can be given only when sufficient new data is collected.

Therefore, we use a data distribution based detection method called the semiparametric log likelihood (SPLL) (Kuncheva, 2013) method to detect CD of the user's home environment without newly labeled data. SPLL models the historical and current data distributions as two Gaussian mixtures and uses the log-likelihood ratio of these two distributions as the detection metric, which is defined as follows:

$$\mathrm{SPLL}=\frac{1}{|W_{\mathrm{s}}|}\sum_{\boldsymbol{x}\in W_{\mathrm{s}}}(\boldsymbol{x}-\boldsymbol{\mu}_{i^*})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_{i^*}),\quad(1)$$

where W_{s} represents the sliding window of the current data and $|W_{\mathrm{s}}|$ represents the cardinality of W_{s}, i^* is the component of the historical distribution whose mean $\boldsymbol{\mu}_{i^*}$ has the smallest Mahalanobis distance to the current data $\boldsymbol{x}$, and $\boldsymbol{\Sigma}$ is the feature covariance matrix. In other words, the detection metric of SPLL is the mean squared Mahalanobis distance of the current data to the nearest historical distribution components. Such a metric has a chi-square distribution whose number of degrees of freedom is equal to the dimensionality of the data feature n. CD is declared if $\mathrm{SPLL}>n+\sqrt{2n}$.

Furthermore, to calculate the detection metric of SPLL in the context of using CNN to perform classification tasks, we use the output of the penultimate layer of the CNN as the feature representation of the input data and project the data feature to a lower-dimensional space by adopting principal component analysis (PCA) (Pedregosa et al., 2011), which ensures that the feature is an accurate representation of the raw data and that the feature covariance matrix is invertible. The number of principal components in PCA can be determined empirically.

3.4 Cloud–robot knowledge transfer

The next problem of our cloud-assisted robot cognition adaptation mechanism is how to integrate the global cloud model and the local customized model to adapt to environmental changes. Intuitively, the local customized model trained by labeled data obtained from the initial home environment is focused on only one part of the sample space and underperforms in the other part of the sample space,

since the robot has not learned the remainder of the sample space. The global cloud model built by the data collected from a collection of robots contains the knowledge of the whole sample space, but is inferior to the local customized model in the region corresponding to the initial home environment. This is mainly because the global cloud model spreads its knowledge capacity to the whole sample space instead of focusing on a small part of the sample space. Thus, knowledge transfer from the global model to the local model should be achieved through model fusion, which we believe may benefit from the feature transferability of CNN.

The transferability of features in CNN can be categorized into two types, either specific to a particular task or general across many tasks. Existing research (Yosinski et al., 2014) has revealed that the generality and specificity of features are closely related to the depth of a particular layer; i.e., the lower-layer features of CNNs are more general, while the higher-layer features of CNNs are more specific. Based on this conclusion, we can assume that the reason of the accuracy degradation of the local customized model is that higher-layer parameters become obsolete to the environmental changes rather than all parameters of the model. Therefore, only the higher-layer parameters of the local customized model need to be modified with the help of the corresponding parameters of the global cloud model, while the lower-layer parameters can remain unchanged. Hence, we propose two model fusion methods to merge the higher-layer parameters of the local customized model and the global cloud model, which are described as follows:

1. Merge–combine fusion method. The higher-layer parameters of the local customized model are replaced with the corresponding parameters of the global cloud model, while the lower-layer parameters remain unchanged.

2. Merge–added fusion method. The higher-layer parameters of the local customized model are replaced by the weighted sum of the corresponding parameters of the local customized model and the global cloud model, while the lower-layer parameters remain unchanged.

In addition, we define the third model fusion method for comparison, which can be described as follows:

3. Merge–random fusion method. The higher-layer parameters of the local customized model are randomly initialized, while the lower-layer parameters remain unchanged.

The three model fusion methods can be expressed as follows:

$$\text{Merge–random fusion: } M_{\mathrm{A}}(\mathrm{L})|R \to \mathrm{A}+\mathrm{B}, \quad (2)$$

$$\text{Merge–combine fusion: } M_{\mathrm{A}}(\mathrm{L})|M_{\Omega}(\mathrm{H}) \to \mathrm{A}+\mathrm{B}, \quad (3)$$

$$\text{Merge–added fusion: } M_{\mathrm{A}}(\mathrm{L})|w_{\mathrm{A}}M_{\mathrm{A}}(\mathrm{H})+w_{\Omega}M_{\Omega}(\mathrm{H}) \to \mathrm{A}+\mathrm{B}, \quad (4)$$

where A and A + B represent the sample space of the user's home environment before and after changes respectively, and B represents the environmental changes. $M_{\mathrm{A}}(\mathrm{L})$ and $M_{\mathrm{A}}(\mathrm{H})$ are the lower-layer parameters and the higher-layer parameters of the local customized model M_{A}, respectively. $M_{\Omega}(\mathrm{H})$ denotes the corresponding higher-layer parameters of the global cloud model M_{Ω}. The left side of the arrow represents the model fusion pattern, where the part before the vertical line represents the source of the lower-layer parameters and the part after it represents the source of the higher-layer parameters. Moreover, w_{A} and w_{Ω} are the weights of the local customized model and the global cloud model, respectively. Here, we define w_{A}, $w_{\Omega} \in [0,1]$ and $w_{\mathrm{A}}+w_{\Omega}=1$. R represents the randomly initialized parameters. The arrow means that the merged model should adapt to the sample space of the user's home environment after changes.

As one of the key factors that affect the effectiveness of the model fusion methods, the boundary, or the dividing line, between the lower layers and the higher layers determines how many lower layers will remain unchanged in the merged model. For example, if the high–low boundary is determined to be the sixth layer of a 16-layer CNN, it means that the first- to sixth-layer parameters remain unchanged, while the seventh- to sixteenth-layer parameters are modified. Considering the two extreme choices of the high–low boundary, when the boundary is determined as the "0^{th} layer" of the CNN, the merge–combine fusion method, merge–added fusion method, and merge–random fusion method are equivalent to directly replacing the local customized model with the global cloud model, the weighted sum of the local customized model and the global cloud model, and a randomly initialized model, respectively. When the boundary is determined as

the last layer of the CNN, it means that all of the three model fusion methods are equivalent to keeping the whole local customized model unchanged. Intuitively, common choices between the "0^{th} layer" and the last layer of the CNN should achieve a compromised performance between these two extremes.

The weights of the merge–added fusion method comprise another key factor that affects the model effectiveness. Similarly, considering the two extreme values of w_{A}, $w_{\text{A}} = 0$ means that the merge–added fusion method is equivalent to the merge–combine fusion method, and $w_{\text{A}} = 1$ means keeping the whole local customized model unchanged. Intuitively, common values of w_{A} between zero and one should also achieve a compromised performance between these two extremes. Experimental evaluations of the performance of the three model fusion methods under different boundary and weight settings are conducted in the next section.

4 Experiments

In this section, we present the implementation details of the proposed cloud-assisted robot cognition adaptation mechanism and the evaluation results of the three model fusion methods, which include the CNN we used, the dataset used to build the local customized model and the global cloud model, the detection accuracy of the SPLL CD detection method, the comparison of the three model fusion methods under different settings, and the guideline of the most suitable fusion method for the cloud–robot knowledge transfer.

4.1 Experimental settings

To evaluate the proposed cloud-assisted robot cognition adaptation mechanism, the object classification task was chosen as a case study, which is one of the most typical cognitive tasks of home service robots. The whole dataset for this task was collected by manually selecting six classes of common objects in the home environment from the ImageNet dataset (http://www.image-net.org/), which are bottle, cat, chair, dog, knife, and table. Meanwhile, a total of 22 child classes were selected from these six parent classes which show different patterns within the parent classes. Figs. 2 and 3 show the training examples of different child classes of dogs and tables, respectively. The goal of the object classification task was to classify samples of the child classes as its parent class. The number of samples for each child class was larger than 1000, where 70% of the sample images were chosen randomly to be used for training, 15% for validation, and 15% for testing. The selected parent–child classes can be described as follows:

Bottle: {beer bottle, ink bottle};
Cat: {Egyptian cat, Persian cat, Siamese cat, tabby cat, tiger cat};
Chair: {armchair, folding chair, swivel chair, wheelchair};
Dog: {Basenji, corgi, Dalmatian, Griffon, Newfoundland};
Knife: {cleaver, pocketknife};
Table: {desk, dining table, pool table, tabletennis table}.

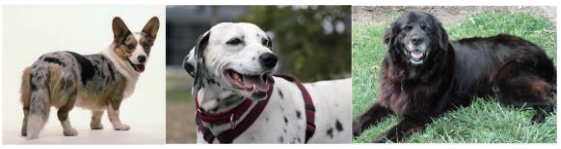

Fig. 2 An example of the corgi (left), Dalmatian (middle), and Newfoundland (right) training sample

Fig. 3 An example of the desk (left), dining table (middle), and pool table (right) training sample

To simulate the real robotic applications, we distributed samples of one child class under each parent class for each robot as the training set of the initial environment of the user's home and ensured that the parent–child class pattern between the robots was different; thus, a total of 1600 different initial home environments could be obtained. The changes in the user's home environment were defined as the addition of several other child classes to some of the parent classes. Here, we focused on one robot r as a representative of the robot community to conduct experimental evaluations of the proposed cognition adaptation mechanism; the parent–child class pattern of the initial home environment for this robot is described as follows:

Home environment of robot r before changes:
Bottle: {beer bottle}; Cat: {tiger cat};
Chair: {armchair}; Dog: {Dalmatian};
Knife: {cleaver}; Table: {dining table}.

The parent–child class pattern of the home environment of robot r after changes can be described as follows:

Home environment of robot r after changes:
Bottle: {beer bottle, ink bottle}; Cat: {tiger cat};
Chair: {armchair, wheelchair}; Dog: {Dalmatian};
Knife: {cleaver, pocketknife};
Table: {dining table, desk}.

Herein the child classes of ink bottle, wheelchair, pocketknife, and desk were added to the initial home environment as the environmental changes. Samples of these newly added child classes were unlabeled.

To build the local customized model and the global cloud model, we implemented the VGG16 neural network (Simonyan and Zisserman, 2014) as the learner of the object classification task and followed the fine-tuning approach to train the network with the training set of each robot's initial environment and the cloud dataset; the cloud dataset consisted of a fraction of the training data for all the child classes, and the starting point of the fine-tuning process was a pretrained VGG16 available in the TensorFlow-Slim model library (TensorFlow, 2020). After 60 epochs of training, the accuracy of the local customized model of robot r and the global cloud model when applied on the test set of the home environment of robot r before and after changes in the VGG16 neural network (Table 1) confirmed our expectation about the performance difference between the local customized model and the global cloud model, as mentioned in Section 3.4. Moreover, the same experiments were carried out on the VGG19 neural network to further illustrate the generality of the proposed model fusion methods. Table 2 shows the accuracy of the local customized model of robot r and the global cloud model when implemented on the test set of the home environment of robot r before and after changes in the VGG19 neural network.

Furthermore, to verify the effectiveness of the CD detection procedure, we tested the SPLL detection method with the local customized model of robot r on the test set of the initial home environment and the added child classes which served as the environmental changes, whose ground truths are not drift (false) and drift (true) compared to the initial home environment. The means and the covariance matrix of the historical distribution were calculated using the features of the whole training set of the initial home environment on the local customized model. The cardinality of the sliding window was set to 32. Finally, the SPLL detection method showed a true-positive rate of 0.7273 and a true-negative rate of 0.8333 on the test set of the added child classes and the initial home environment of robot r, respectively, which means that this method can provide a high detection accuracy for the environmental changes.

Table 1 Test accuracy of the local customized model of robot r and the global cloud model in the VGG16 neural network

Home environment	Test accuracy	
	Local customized	Global cloud
Before changes	0.9551	0.9394
After changes	0.8550	0.9400

Table 2 Test accuracy of the local customized model of robot r and the global cloud model in the VGG19 neural network

Home environment	Test accuracy	
	Local customized	Global cloud
Before changes	0.9540	0.9349
After changes	0.8521	0.9450

4.2 Results and analysis

Fig. 4 shows the test results of the three model fusion methods under the home environment of robot r before and after changes with different high–low boundaries. The red dotted line and blue dotted line represent the accuracies of the local customized model and global cloud model, respectively. The magenta, cyan, and green curves represent the results of the merge–random fusion method, merge–combine fusion method, and merge–added fusion method with $w_\Omega = 0.5$, respectively. Moreover, not only the results of common choices of the high–low boundary between the "0^{th} layer" (boundary layer 0) and the last layer (boundary layer 16), but also the results of these two extremes are shown in Fig. 4 for comparison.

Interesting phenomena can be seen in Fig. 4a. For the home environment after changes, the merge–combine fusion method showed an accuracy very close to that of the global cloud model at boundary layers 0–14, and even exceeded that of the global

cloud model at some of the boundaries, which is beyond our expectations. The merge–added fusion method with $w_\Omega = 0.5$ showed an accuracy that was close to but did not exceed that of the global cloud model. The merge–random fusion method showed a complete failure at all boundaries except in the extreme case (at boundary layer 16), since its higher-layer parameters were randomly initialized and had no relationship with the lower-layer parameters. The performances of both the merge–combine fusion method at boundary layers 0–14 and the merge–added fusion method at boundary layers 0–15 were better than the performance of the local customized model, which means that these two model fusion methods can adapt the local customized model to the environmental changes. Besides, a sudden decline can be seen in the merge–combine fusion method's test results at boundary layer 15, which implied that the feature extraction layers from the local customized model did not match the classification layer from the global cloud model.

Meanwhile, the degrees of forgetting the learned knowledge of the initial home environment by the three model fusion methods are shown in Fig. 4b, which shows the accuracy of the three model fusion methods on the test set of the initial home environment. In this subfigure, the merge–combine fusion method showed an accuracy that was close to but did not exceed that of the local customized model, while the merge–added fusion method with $w_\Omega = 0.5$ showed an accuracy that was very close to that of the local customized model and even exceeded that of the local customized model at some of the boundaries, which is also beyond our expectations. Similarly, the merge–random fusion method showed a complete failure except at boundary layer 16.

Furthermore, we tested the merge–added fusion method on the test set of the home environment of robot r before and after changes with different weight settings as shown in Fig. 5, in which the brown, orange, pink, purple, green, yellow, and black curves represent the results of the merge–added fusion method with w_Ω equaling 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, and 0.3, respectively. A distinct difference can be seen in the comparison between the merge–added fusion method and the merge–combine fusion method with different weight settings. The merge–added fusion method with $w_\Omega \in [0.6, 0.8]$ at boundary layer 13 showed a performance that was very close to those of the merge–combine fusion method and the global cloud model under the home environment after changes and almost no forgetting the learned knowledge. However, the merge–added fusion method with other weight and boundary settings either showed a poorer performance than the merge–combine fusion method under the home environment after changes or forgot more learned knowledge of the initial home environment. Meanwhile, parameters of the first 13 layers, which accounted for 10.95% of the entire set of CNN parameters, did not need to be downloaded from the cloud, which means that 10.95% of the communication cost can be reduced compared to directly replacing the local customized model with the global cloud model.

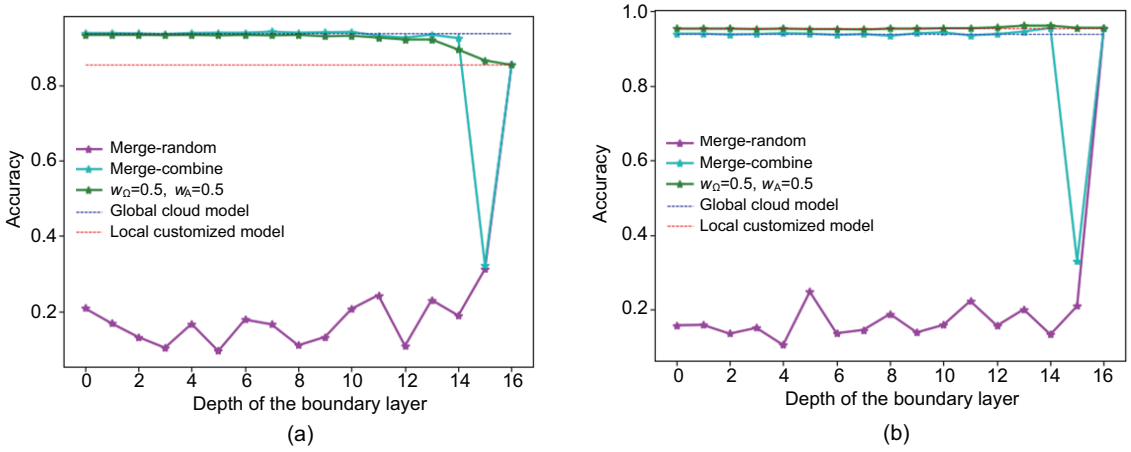

Fig. 4 Test results of the three model fusion methods with different high–low boundaries under the home environment after (a) and before (b) changes in the VGG16 neural network (References to color refer to the online version of this figure)

Similar phenomena can be seen in the experimental results of the VGG19 neural network, as shown in Fig. 6, in which the corresponding relationship between the results of the merge–added fusion method with different w_Ω settings and different curves is the same as in Fig. 5. The merge–added fusion method with $w_\Omega \in [0.6, 0.7]$ at boundary layer 16 showed a performance that was very close to those of the merge–combine fusion method and the global cloud model under the home environment after changes and almost no forgetting the learned knowledge, and the merge–added fusion method with other weight and boundary settings either showed a poorer performance than the merge–combine fusion method under the home environment after changes or forgot more about the learned knowledge of the initial home environment. As parameters of the first 16 layers accounting for 14.34% of the entire set of CNN parameters did not need to be downloaded from the cloud, 14.34% of the communication cost can be reduced compared to directly replacing the local customized model with the global cloud model.

Finally, we defined the merge–added fusion method whose weights are within a certain range and the boundary is a high layer except the penultimate layer of the CNN as the best model fusion method to merge the global cloud model with the local customized model, since it has better performance under the user's home environment before and after changes. In other words, the higher-layer parameters of the robot's local customized model should be replaced by the weighted sum of the corresponding

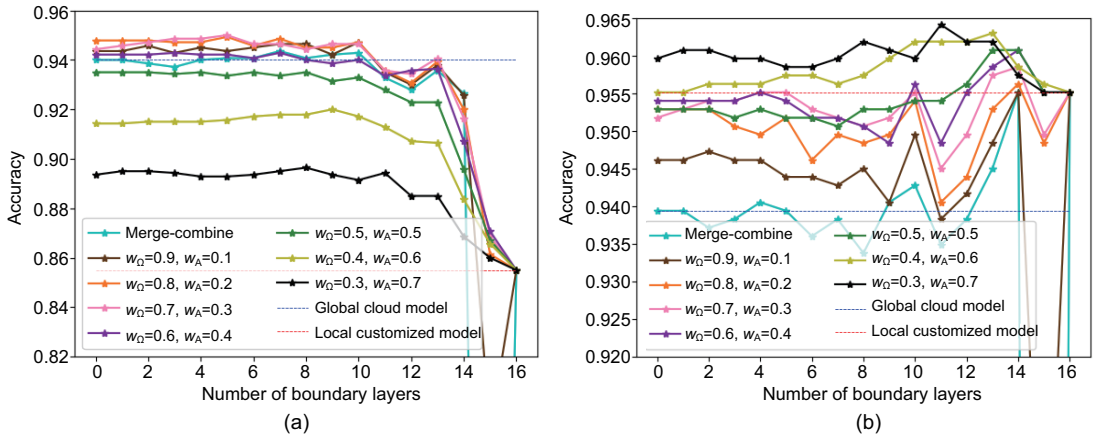

Fig. 5 Test results of the merge–added fusion method with different weight settings under the home environment after (a) and before (b) changes in the VGG16 neural network (References to color refer to the online version of this figure)

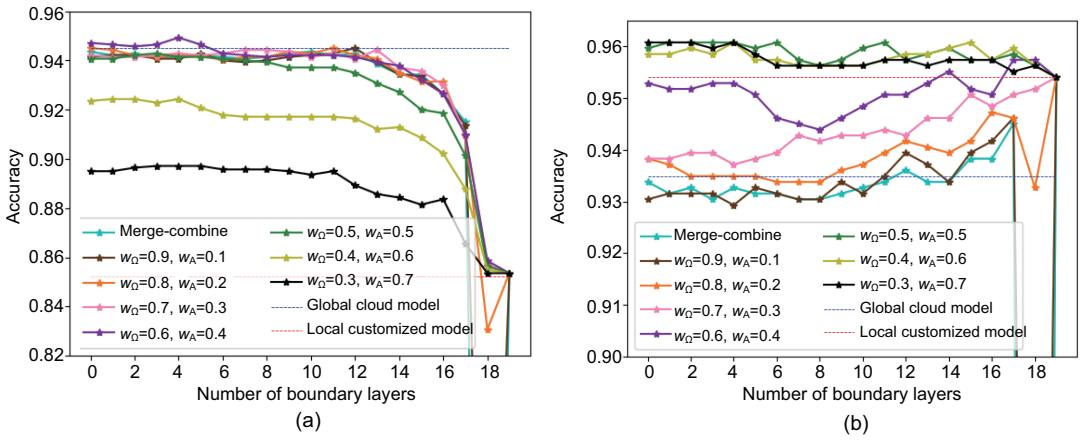

Fig. 6 Test results of the merge–added fusion method with different weight settings under the home environment after (a) and before (b) changes in the VGG19 neural network (References to color refer to the online version of this figure)

parameters of the local customized model and the global cloud model, while the lower-layer parameters can remain unchanged. Note that although our experiments were conducted with a case study using the VGG16 neural network to perform the object classification task, this guideline of knowledge transfer between the global cloud model and the local customized model is general and can be applied to other CNNs and other image classification tasks.

5 Conclusions and future work

In this paper, we proposed a new cloud-assisted robot cognition adaptation mechanism in which knowledge from other robots through the cloud and the appropriate approach was used to find the best model fusion method for the robot to carry out this adaptation process. A model trained by the manually labeled data from the initial environment of the user's home served as the robot's local customized model, and a CD detection method was implemented in the robot to trigger the adaptation procedure, which adapts the local customized model to the environmental changes with the help of a global cloud model built using the data collected from a collection of robots. Three model fusion methods were proposed to achieve knowledge transfer from the global cloud model to the robot's local model. We experimentally analyzed the difference between the three model fusion methods with a case study. The experimental results verified the efficiency and effectiveness of our mechanism and offered an empirical guideline for the cloud–robot knowledge transfer.

In our future work, we will extend the current robot cognition adaptation mechanism to more scenarios of environmental changes, such as adding unknown parent classes to the home environment. Furthermore, the layer-wise model fusion methods can be improved by a kernel-wise fusion method to achieve a finer knowledge transfer. Besides, we will investigate how to adaptively adjust the weights of the merge–added fusion method to achieve the best fusion performance. Moreover, the experimental evaluations of the current cloud-assisted robot cognition adaptation mechanism can be extended using another case study, in which the speech classification task will be chosen as the robot's classification task to prove the effectiveness of this mechanism on tasks other than the image classification task.

Contributors

Qi WANG, Weihua SHENG, and Meiqin LIU designed the research. Qi WANG processed the data and drafted the paper. Zhen FAN, Weihua SHENG, and Meiqin LIU helped organize the paper. All the authors revised and finalized the paper.

Compliance with ethics guidelines

Qi WANG, Zhen FAN, Weihua SHENG, Senlin ZHANG, and Meiqin LIU declare that they have no conflict of interest.

References

Arumugam R, Enti VR, Liu BB, et al., 2010. DAvinCi: a cloud computing framework for service robots. IEEE Int Conf on Robotics and Automation, p.3084-3089. https://doi.org/10.1109/ROBOT.2010.5509469

Baena-García M, del Campo-Ávila J, Fidalgo R, et al., 2006. Early drift detection method. Proc 4th Int Workshop on Knowledge Discovery from Data Streams, p.77-86.

Disabato S, Roveri M, 2019. Learning convolutional neural networks in presence of concept drift. Int Joint Conf on Neural Networks, p.1-8. https://doi.org/10.1109/IJCNN.2019.8851731

Ditzler G, Roveri M, Alippi C, et al., 2015. Learning in nonstationary environments: a survey. *IEEE Comput Intell Mag*, 10(4):12-25. https://doi.org/10.1109/MCI.2015.2471196

Fachantidis N, Dimitriou AG, Pliasa S, et al., 2018. Android OS mobile technologies meets robotics for expandable, exchangeable, reconfigurable, educational, stem-enhancing, socializing robot. Interactive Mobile Communication Technologies and Learning, p.487-497. https://doi.org/10.1007/978-3-319-75175-7_48

Gama J, Medas P, Castillo G, et al., 2004. Learning with drift detection. Proc 17th Brazilian Symp on Artificial Intelligence, p.286-295. https://doi.org/10.1007/978-3-540-28645-5_29

Gouveia BD, Portugal D, Silva DC, et al., 2015. Computation sharing in distributed robotic systems: a case study on SLAM. *IEEE Trans Autom Sci Eng*, 12(2):410-422. https://doi.org/10.1109/TASE.2014.2357216

Graf B, Staab H, 2009. Service robots and automation for the disabled/limited. In: Nof SY (Ed.), Springer Handbook of Automation. Springer Berlin Heidelberg, p.1485-1502. https://doi.org/10.1007/978-3-540-78831-7_84

Huaimin W, Bo D, Xu J, 2018. Cloud robotics: a distributed computing view. In: Jones C, Wang J, Zhan NJ (Eds.), Symp on Real-Time and Hybrid Systems. Springer, Cham, p.231-245. https://doi.org/10.1007/978-3-030-01461-2_12

Hunziker D, Gajamohan M, Waibel M, et al., 2013. Rapyuta: the RoboEarth cloud engine. IEEE Int Conf on Robotics and Automation, p.438-444. https://doi.org/10.1109/ICRA.2013.6630612

Kuncheva LI, 2013. Change detection in streaming multivariate data using likelihood detectors. *IEEE Trans Knowl Data Eng*, 25(5):1175-1180. https://doi.org/10.1109/TKDE.2011.226

LeCun Y, Bengio Y, Hinton G, 2015. Deep learning. *Nature*, 521(7553):436-444. https://doi.org/10.1038/nature14539

McMahan HB, Moore E, Ramage D, et al., 2017. Communication-efficient learning of deep networks from decentralized data. https://arxiv.org/abs/1602.05629

Page ES, 1954. Continuous inspection schemes. *Biometrika*, 41(1-2):100-115. https://doi.org/10.1093/biomet/41.1-2.100

Pedregosa F, Varoquaux G, Gramfort A, et al., 2011. Scikit-learn: machine learning in Python. *J Mach Learn Res*, 12:2825-2830.

Secker J, Hill R, Villeneau L, et al., 2003. Promoting independence: but promoting what and how? *Ageing Soc*, 23(3):375-391. https://doi.org/10.1017/S0144686X03001193

Shuai W, Chen XP, 2019. KeJia: towards an autonomous service robot with tolerance of unexpected environmental changes. *Front Inform Technol Electron Eng*, 20(3):307-317. https://doi.org/10.1631/FITEE.1900096

Siciliano B, Khatib O, 2016. Springer Handbook of Robotics. Springer Berlin Heidelberg, Germany.

Simonyan K, Zisserman A, 2014. Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556

TensorFlow, 2020. TensorFlow-Slim Image Classification Model Library. https://github.com/tensorflow/models/tree/master/research/slim

United Nations, 2020. Ageing. https://www.un.org/en/sections/issues-depth/ageing/

Wang Q, Zhang S, Sheng W, et al., 2021. Multi-style learning for adaptation of perception intelligence in home service robots. *Patt Recogn Lett*, 151:243-251.

Yang Z, Al-Dahidi S, Baraldi P, et al., 2020. A novel concept drift detection method for incremental learning in nonstationary environments. *IEEE Trans Neur Netw Learn Syst*, 31(1):309-320. https://doi.org/10.1109/TNNLS.2019.2900956

Yoon J, Jeong W, Lee G, et al., 2020. Federated continual learning with adaptive parameter communication. https://arxiv.org/abs/2003.03196v1

Yosinski J, Clune J, Bengio Y, et al., 2014. How transferable are features in deep neural networks? https://arxiv.org/abs/1411.1792v1

Žliobaitė I, 2010. Learning under concept drift: an overview. https://arxiv.org/abs/1010.4784

Žliobaitė I, Pechenizkiy M, Gama J, 2016. An overview of concept drift applications. In: Japkowicz N, Stefanowski J (Eds.), Big Data Analysis: New Algorithms for a New Society. Springer, Cham, p.91-114. https://doi.org/10.1007/978-3-319-26989-4_4

Zweigle O, van de Molengraft R, d'Andrea R, et al., 2009. RoboEarth: connecting robots worldwide. Proc 2nd Int Conf on Interaction Sciences: Information Technology, p.184-191. https://doi.org/10.1145/1655925.1655958