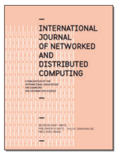
International Journal of Networked and Distributed Computing
Vol. 8(1); December (2019), *pp.* 16–24
DOI: https://doi.org/10.2991/ijndc.k.191118.001; ISSN 2211-7938; eISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc



Research Article

Toward Affordable and Practical Home Context Recognition: –Framework and Implementation with Image-based Cognitive API–

Sinan Chen[1,*], Sachio Saiki[1], Masahide Nakamura[1,2]

[1]*Graduate School of System Informatics, Kobe University, 1-1 Rokkodai-cho, Nada, Kobe, 657-0011, Japan*
[2]*RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo-ku, Tokyo, 103-0027, Japan*

ARTICLE INFO

Article History
Received 23 April 2019
Accepted 20 May 2019

Keywords
Context recognition
image
cognitive APIs
machine learning
smart home

ABSTRACT

To provide affordable context recognition for general households, this paper presents a novel technique that integrate image-based cognitive Application Program Interface (API) and light-weight machine learning. Our key idea is to regard every image as a document by exploiting "tags" derived by the API. We first present a framework that specifies a common workflow of the machine-learning-based home context recognition. We then propose a pragmatic method that implements the framework using the "image-as-a-document" approach.

© 2019 *The Authors*. Published by Atlantis Press SARL.
This is an open access article distributed under the CC BY-NC 4.0 license (http://creativecommons.org/licenses/by-nc/4.0/).

1. INTRODUCTION

With the rapid progress of Information and Communication Technology (ICT) and Internet of Things (IoT) technologies, research and development of *smart homes* have been actively conducted. The smart home enables to collect various data within the house, and use the data for value-added services. The value-added services include elderly monitoring [1–3], autonomous security [4], and personalized healthcare [5,6]. The key challenge in the smart home is how smartly the system is able to recognize various *contexts* at home, automatically. The *home context* refers to any situational information at home, including daily activities of residents, the environment in the house, and the status of the room.

Technologies for the home context recognition have been studied for many years in the field of *ubiquitous computing*. The traditional ubiquitous computing uses ambient sensors (e.g., temperature, humidity, presence) [7], wearable sensors (e.g., accelerometer, heart rate), and indoor positioning systems [8]. They are installed at rooms or worn by residents to collect essential data characterizing various contexts. Using such sensors and devices, a lot of systems have been proposed. They include an elderly watching system using human motion sensors [9], the context labeling and recognition system based on environment change [10], the recognition of daily activities with time-series environmental data [11], the activity recognition with power consumption of home appliances and user's location [12], and context sensing with smart phones [13].

In more recent years, the emerging *deep learning* [14] allows the system to recognize *multimedia* data. Since image, voice, and text data usually contain richer information than the conventional sensor data, it is promising to use such multimedia data for recognizing the home contexts.

Unfortunately, however, these existing technologies are yet far from practical use in general households, since they require expensive resources at home. It is difficult for ordinary users to operate and maintain proprietary systems at home on a daily basis. One may try to recognize home contexts via image recognition based on deep learning. However, constructing a custom recognition model dedicated for a single house requires a huge amount of labeled datasets and computing resources [14,15]. Thus, there is still a big gap between the research and real life.

The goal of research is to make the home context recognition more affordable and feasible for general households.

To achieve the goal, we first present a *framework* for the machine-learning-based home context recognition. This framework specifies a common workflow where individual users to define custom contexts, collect data, and construct their own context recognition models. The framework does not specify concrete data or algorithms, so that the users can exploit any methods (even the deep learning) based on their requirements and constraints.

As an implementation of the framework, we then propose a home context recognition method. The proposed method extensively utilizes the *image-based cognitive Application Program Interface (API)*, which is the application program interface to cloud-based computer vision services [16–18]. The API receives an image from an external application, recognizes specific information within the image, and returns the information as a set of text words called *tags*. Our key idea is to use these tags as features of the image, and

[*]*Corresponding author. Email:* chensinan@ws.cs.kobe-u.ac.jp

apply light-weight machine-learning techniques to infer the target context. Since every image can be considered as a document, the expensive deep learning is no more needed. We call this idea the *image-as-a-document* approach.

To demonstrate the practical feasibility, we have conducted an experiment that recognizes contexts within our laboratory. We installed an USB camera to take a snapshot every 5 s, and stored the images in a server for 2 weeks. We then defined seven custom contexts: "General meeting", "Cleaning", "Eating", "No people", "Personal discussion", "Gaming", and "Studying". For each context, we selected 100 representative images considered to expose the context well. The selected 700 images were sent to *Microsoft Azure Computer Vision API* [16], in order to retrieve tags from each image. Regarding the tags as a document (corpus), we exploit *Term Frequency - Inverse Document Frequency (TF-IDF)* [19] to transform each tag set into a document vector. Finally, we imported the vectors *Microsoft Azure Machine Learning Studio* [20], and constructed a classifier that recognizes the seven contexts, using *Multi-class Neural Network*.

The experimental results showed that the overall accuracy of the recognition model was 0.929, and the average accuracy was 0.980. Then, the micro-averaged precision and the macro-averaged precision were 0.929 and 0.924, respectively. According to the confusion matrix, the recognition accuracy of "General meeting" was 95.3%, "Cleaning" was 90.9%, "Eating" was 83.3%, "No people" was 100.0%, "Personal discussion" was 96.0%, "Gaming" was 82.2%, and "Studying" was 100.0%. Thus, the constructed model was able to recognize "No people" and "Studying", perfectly. On the other hand, the model was bad at recognizing "Eating" or "Gaming", where multiple people appears within an image.

2. PRELIMINARIES

2.1. Home Context Recognition

The *home contexts* refer to any situational information at home, including daily activities of residents, the environment in the house, the status of the room. Typical home contexts are, for example, "residents are in the dining room", "it is warm in the dining room", "the dining room is clean", "the light in the dining room is on", etc.

We use the term *custom* home context to represent home context that is specifically defined by individual residents, houses, and environment, for a special purpose of application. For example, suppose that a son is worried about his old parents living in a remote place. Then, the contexts like "parents are eating breakfast in a dining room", "father is taking medicine in a dining room", "mother is cleaning a dining room", are crucial information for the son. If these custom contexts can be recognized by an elderly monitoring system, and the information is regularly sent to the son, it would be a great value for the son.

2.2. Technical Challenges

In the home context recognition, we consider that there are two big challenges.

The first challenge lies in *individuality*. As seen in the above example, the custom contexts are defined by every user depending on a special purpose. Also, the layout, the environment, and the configuration of the target space vary from one house to another. Therefore, it is impossible to construct a universal context recognition method that satisfies all users, houses, and environments.

The second challenge is *acceptability*. Most existing technologies are developed and tested on research labs or dedicated smart homes, and few of them are actually operated on general households. In order for the technology to be accepted, it should be easy to operate and maintain, should be affordable enough, and should not be intrusive for daily life. Of course, the security and privacy issues influence the acceptability. The user of the technology must be fully convinced what data is collected for what purpose and consumed by whom.

2.3. Image-based Cognitive API

In recent years, world's cloud companies such as Microsoft, IBM, and Google, released *cognitive services*. A cognitive service provides the capability to understand multimedia data based on sophisticated machine-learning algorithms powered by big data and large-scale computing resources. Typical services include image recognition, speech recognition, and natural language processing. A cognitive service usually provides *cognitive APIs*, with which developers can easily integrate powerful recognition features in their own applications. We consider that the cognitive APIs make full use of multimedia data, therefore, they have great potential to improve smart homes since the user would no longer need to maintain an expensive system at home.

Although various kinds of cognitive APIs exist, we especially focus on *image-based* cognitive APIs in this paper. An image-based cognitive API receives an image from an external application, extracts specific information from the image, and returns the information as a set of words called *tags*.

Figure 1 represents the usage of image recognition APIs. The information of interest varies between services. For example, MS-Azure Face API [16] estimates age, sex, and emotional values from a given human face image. IBM Watson Visual Recognition [17] recognize items in the image such as home appliances, furniture, and tools. Google Cloud Vision API [18] outputs pre-defined concept labels associated to recognized objects.

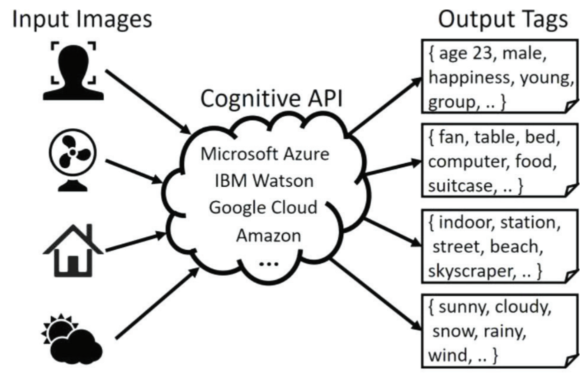

Input Images
Output Tags
Cognitive API
Microsoft Azure
IBM Watson
Google Cloud
Amazon
...
{ age 23, male, happiness, young, group, .. }
{ fan, table, bed, computer, food, suitcase, .. }
{ indoor, station, street, beach, skyscraper, .. }
{ sunny, cloudy, snow, rainy, wind, .. }

Figure 1 | Usage of image-based cognitive APIs.

2.4. Machine Learning Platform on Cloud

A reasonable approach to cope with the individuality is to use the machine learning, in order to build a custom context recognition model for a house. However, using machine learning would decrease the acceptability, since it is too expensive for general households to have sufficient computing resources for the machine learning.

One solution for this problem is to use the *machine learning platform* on the cloud. The platform provides a space and resources, with which individual users can construct and deploy custom machine-learning models, depending on requirements of users. A user uploads the data on the cloud platform, constructs and deploys their own machine-learning models. For this, most platforms usually provide the visual programming environment, in which the user just connects built-in components of pre-processing, procedures, algorithms, and evaluation, without writing any program code.

The well-known machine-learning platforms include *Microsoft Azure Machine Learning Studio* [20], *Google Cloud Machine Learning Engine* [21], and *Amazon Sage Maker* [22].

3. FRAMEWORK OF MACHINE-LEARNING-BASED HOME CONTEXT RECOGNITION

3.1. Purpose of Framework

To encourage the individual users to build custom home context recognition models by machine learning, we should define a solid and clear method. At the same time, when we consider the individuality, we should leave a certain extent of *freedom*, so that individual users can freely choose appropriate methods.

For this purpose, we propose a *framework* of the machine-learning-based home context recognition in this section. This framework specifies a common workflow where individual users to define custom contexts, collect data, and construct their own context recognition models. The framework does not specify concrete data or algorithms, so that the users can choose appropriate methods (even the deep learning) based on requirements and constraints.

Figure 2 shows the overview of the proposed framework. The framework roughly consists of four steps, each of which is further divided into sub-steps. The following subsections explain the four steps in details.

3.2. Step 1: Acquiring Data

In this step, a user of the proposed method acquires the data essential for the machine-learning-based home context recognition.

3.2.1. Step 1-I: defining custom contexts

First of all, the user defines a set $C = \{c_1, c_2, ..., c_m\}$ of custom home contexts to be recognized. The contexts are associated to a certain space within a house. Each element in C works as a *label* of the machine learning.

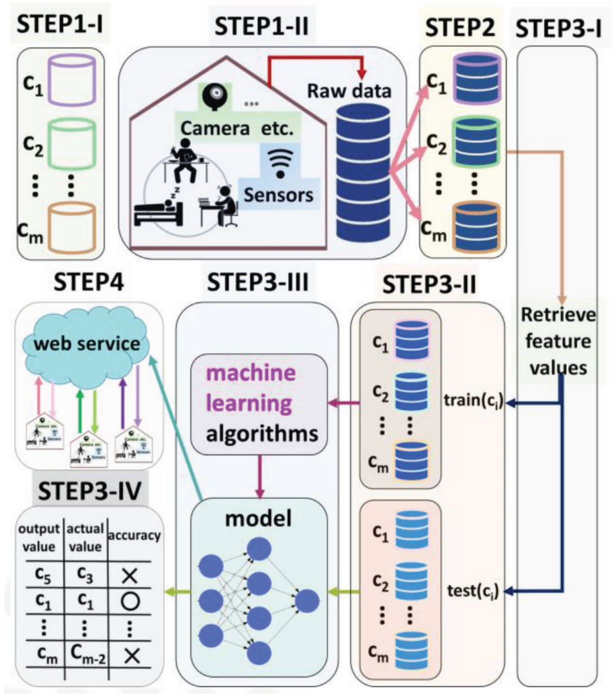


Figure 2 | The framework of home context recognition using machine learning.

3.2.2. Step 1-II: acquiring raw data

The user then deploys a device within the target space, in order to acquire raw data. The device may be a sensor, a camera, or microphone, depending on the user's choice. However, it should be carefully chosen so that the acquired data should be relevant to the target contexts. Using the device, the system acquires the data periodically, with an appropriate interval. The data should be accumulated in a server.

3.3. Step 2: Creating Dataset

In this step, the user creates dataset used for the machine learning.

For each context $c_i \in C$, the user samples representative n data $data(c_i) = \{d_{i1}, d_{i2}, ..., d_{in}\}$ that well expose c_i from raw data obtained in Step 1. In this step, the total $m \times n$ data items are obtained

3.4. Step 3: Constructing Recognition Model

In this step, the user constructs a context recognition model using machine learning. The step is further divided into the following four sub-steps.

3.4.1. Step 3-I: retrieving feature values

From the dataset obtained in Step 2, the user retrieves *feature values* relevant for inferring the target context.

For $data(c_i) = \{d_{i1}, d_{i2}, ..., d_{in}\}$ $(1 \le i \le m)$, the user extracts $F(data(c_i)) = F(d_{i1}), F(d_{i2}), ..., F(d_{in})$, where F is a function that retrieves relevant feature values from d_{ij}. Note that F is automatically calculated if the user uses the deep learning.

3.4.2. Step 3-II: splitting data

For each context c_i, the user splits the feature values $F(data(c_i))$ into training and test data. Specifically, the user divides n feature values $F\{(d_{i1}), F(d_{i2}),..., F(d_{in})\}$ into α data items and $n\alpha$ data items, which are denoted by $train(c_i)$ and $test(c_i)$, respectively.

Regarding the way of data split, there are various methods including random sampling, hold-out, and cross validation.

3.4.3. Step 3-III: training the model

The user chooses a *supervised* machine learning algorithm A. By applying A to the training data $train(c_1),train(c_2), ..., train(c_m)$ and corresponding labels $c_1, c_2, ..., c_m$, the user constructs a model M that infers c_i from given input feature values $F(d_{ij})$. In fact, M is a multi-class classifier that outputs a category c_i for input $F(d_{ij})(1 \le i \le m)$.

Regarding algorithm A, there are various algorithms including Neural Network (NN), Support Vector Machine (SVM), Decision Tree, Random Forest, and a variety of deep learning techniques.

3.4.4. Step 3-IV: evaluating the model

Using the test data $test(c_1),test(c_2), ..., test(c_m)$ the user evaluates the model M. The evaluation is done by checking if M correctly outputs c_i for given $test(c_i)$. The more M outputs the correct contexts, we say that M is more *accurate*. The accuracy is computed by the ratio of the number of test data correctly classified against the number of all test data.

If the accuracy of M is low or unable to meet the requirements, the user reconstructs the model by repeating the previous steps.

3.5. Step 4: Moving to Operation

Finally, the user deploys the model, and start operation of automatic home context recognition.

3.5.1. Step 4-I: deploying the model

The user deploys the trained model M in a platform (cloud or edge server) that can be accessed from the target space in the home, and makes it online. A cloud-based machine-learning platform (see Section 2.4) usually provides a service that deploys a trained model as a Web service. Once the model is deployed as a Web service, the external application can consume the model via HTTP/REST protocol.

3.5.2. Step 4-II: operating the model

The user creates applications using the context recognition model. The application acquires data of the target space, then extracts the feature values from the data, and finally sends the feature values to the deployed model. The model returns a custom context inferred from the feature. Then, the application is able to know the current context, and use the context for various context-aware actions.

4. IMPLEMENTING AFFORDABLE HOME CONTEXT RECOGNITION BY IMAGE-AS-A-DOCUMENT APPROACH

In this section, we propose a pragmatic method that implements the framework in Section 3. In the proposed method, we collect *images* of the target area as essential data of the machine learning. In order to achieve affordable and practical home context recognition, we extensively introduce the *image-as-a-document concept* using the image-based cognitive API.

4.1. Image as a Document

We consider to implement the framework using *images* by installing a camera in the target space. This is because camera devices (e.g., USB camera) are already so common that the general users can easily introduce the cameras at home.

The most straight-forward way to construct a high-quality image recognition model is to use *deep learning*. In this approach, the user constructs M using a deep-learning technique A in Step 3 of the framework (See Section 3), where the function F is automatically computed by the deep learning. However, since this approach requires a huge amount of training data and computing resources, it is not realistic to implement it in general households.

Instead of using the deep learning, the proposed method extensively utilize the image-based cognitive API (see Section 2.3). In the proposed method, we acquire images of the target space, and create the dataset by sampling representative images for the custom contexts. Then, every sampled image is sent to the image-based cognitive API. As a result, the API returns a set of text words, i.e., *tags*, which represent concepts that the API recognized within the image. Our key idea is to use these tags as the feature of the original image. Thus, every image (as a collection of pixels) is converted as a document of words. This is the *image-as-a-document* concept.

Once every image is converted into a document of tags, it is possible to apply *document-based* machine-learning techniques, which are much less expensive than the deep learning. There are various methods that extract features from documents, including TF-IDF [19], Word2Vec [23], GloVe [24], and fastText [25].

Using one of these methods, every image as a document is converted into a *document vector*. By applying an ordinary (light-weight) machine learning to the document vectors, the proposed method constructs a classifier of custom home contexts.

Figure 3 shows a comparison between the proposed image-as-a-document approach and the conventional deep learning approach. This figure describes Step 3 and later of the framework. The upper part of the figure shows the proposed method, where (1) sampled images for each custom context are sent to the image-based cognitive API, (2) the resultant documents are encoded into document vectors, and (3) an ordinary machine-learning algorithm is applied

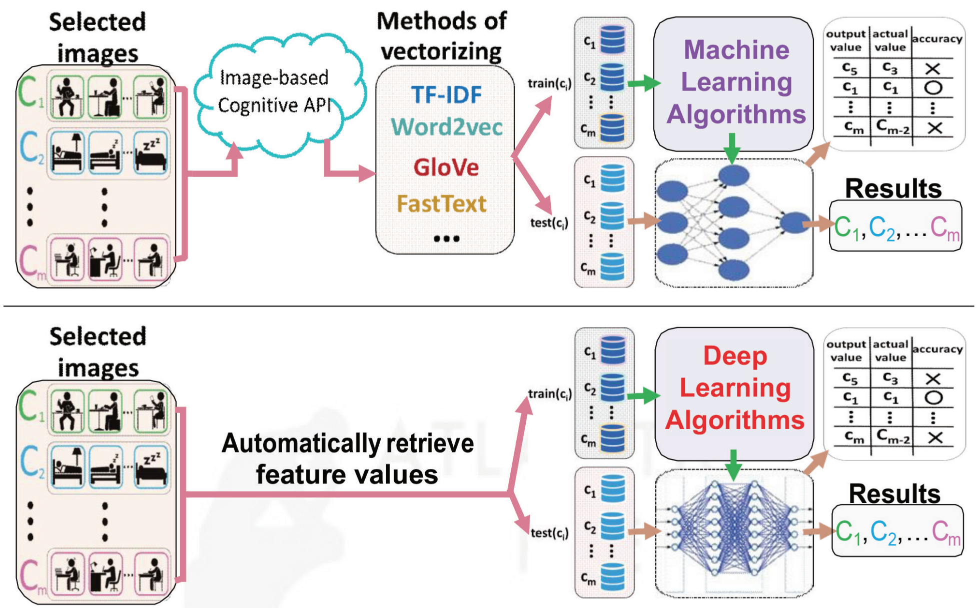


Figure 3 | The comparison of the proposed method and the method with deep learning.

to construct and evaluate a context classifier. On the other hand, the lower part of the figure shows the method with deep learning approach. The sampled images are directly fed by the deep learning algorithm, where the features are extracted automatically.

4.2. Proposed Method

Now, we implement the framework in Section 3 with the image-as-a-document approach.

In Step 1, the user deploys a camera in a target space. Using the camera, the system periodically shoots the image of the target space, and accumulates the images in a server.

In Step 2, for each custom context c_i $(1 \le i \le m)$, the user chooses n representative images $data(c_i) = \{\text{img}_{i1}, \text{img}_{i2}, \ldots, \text{img}_{in}\}$.

In Step 3, the user retrieves features from every image img_{ij}. More specifically, Step 3-I of the framework is implemented as follows:

4.2.1. *Step 3-I-I: retrieving tags from each image*

The system sends every image img_{ij} to an image-based cognitive API, and obtains a set of tags $\text{Tag}(\text{img}_{ij}) = w_1, w_2, w_3$, $\text{Tag}(\text{img}_{ij})$ represents an image as a document of the original image img_{ij}.

4.2.2. *Step 3-I-II: Converting tags into a document vector*

Regarding all tag sets $\bigcup_{ij} \text{Tag}(\text{img}_{ij})$ as a document corpus, the system converts each $\text{Tag}(\text{img}_{ij})$ into a document vector $v_{ij} = [v_1, v_2, \ldots]$, using a document vectorizing technique (e.g., TF-IDF). The document vector v_{ij} is count as the feature of img_{ij}, i.e., $F(\text{img}_{ij}) = v_{ij}$.

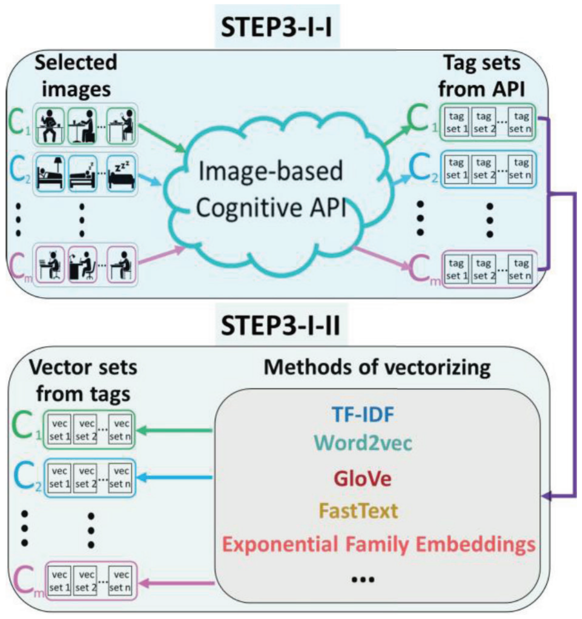


Figure 4 | Step 3-I implemented by the proposed method.

Figure 4 shows the above steps schematically.

In Step 3-II, these document vectors are split into training data and test data. In Step 3-III, the user applies an ordinary supervised machine-learning algorithm A to the training data to constructs a model M, and evaluates M with the test data in Step 3-IV.

In Step 4, when the application operates the model M, the application first acquires an image img_x of the target space, sends img_x to the same image-based cognitive API to obtain $\text{Tag}(\text{img}_x)$, converts $\text{Tag}(\text{img}_x)$ into a document vector $v_x = [v_1, v_2, \ldots]$, and sends v_x to M. Then, the model M returns a context c_y inferred from img_x, by which the application knows the current context of the target space.

5. EXPERIMENTAL EVALUATION

To demonstrate the practical feasibility of the proposed method, we have conducted an experiment. In our laboratory, there is a shared space that is used by members of our laboratory for various activities. The purpose of the experiment is to recognize contexts of the shared space using the proposed image-as-a-document approach.

5.1. Creating Image Dataset

According to Step 1-I of the framework, we have defined the seven custom contexts to be recognized: "General meeting", "Cleaning", "Eating", "No people", "Personal discussion", "Gaming", and "Studying". These are common activities which our members often do within the shared space. In Step 1-II, we installed a USB camera to take images of the shared space from a *fixed point*. We then developed a program taking a snapshot with the USB camera every five seconds. The images were accumulated in a server for 2 weeks.

As for Step 2, for each of the seven contexts, we selected 100 representative images considered to expose the context well. The image sampling was done manually by visual inspection. At this time, we obtained a total of 700 images (100 images, seven contexts). Figure 5 shows the representative images for the seven contexts, and the picture of the USB camera.

5.2. Constructing Context Recognition Model

We performed Step 3 of the framework according to the sub-steps presented in Section 4.2. For the image-based cognitive API, we used the *Microsoft Azure Computer Vision API* [16]. According to Step 3-I-I, we first sent all the sampled images to API, and obtained tag sets for each image.

Table 1 shows examples of extracted tags. Each row represent a context, and a set of tags that the API extracted from a single image belonging to the context. We can see in the table that the API recognized slightly different concepts for different contexts. However, the extracted tags were not directly related to the target context as they were. This is because the API just performs general-purpose image recognition, but is not trained for the special purpose of our custom contexts.

Table 1 | Examples of tags extracted from images

Context label	Tag results
General meeting	Indoor, person, room, living, people, table, sitting, man, group, computer, woman, playing, laptop, video, large, television, standing, game
Cleaning	Indoor, living, room, table, television, fire, fireplace, man, standing, filled, video, playing, woman, furniture, large, people, wii, dog, game
Eating	Indoor, person, room, table, living, man, sitting, food, filled, luggage, people, standing, suitcase, television, young, large, fire, kitchen
No people	Indoor, living, table, room, television, kitchen, sitting, furniture, food, fireplace, fire, place, filled, wooden, large, white, counter, refrigerator, stove, people, oven
Personal discussion	Indoor, living, room, table, sitting, computer, television, laptop, fire, area, cluttered, man, filled, woman, furniture, fireplace, desk, people, young, playing, video, large, standing, wii
Gaming	Indoor, person, room, table, sitting, living, people, man, food, standing, large, group, woman, playing, computer, kitchen, game
Studying	Indoor, living, room, table, television, sitting, fire, fireplace, furniture, area, computer, laptop, filled, screen, desk, place, large, people, woman, video, playing, standing, man, white

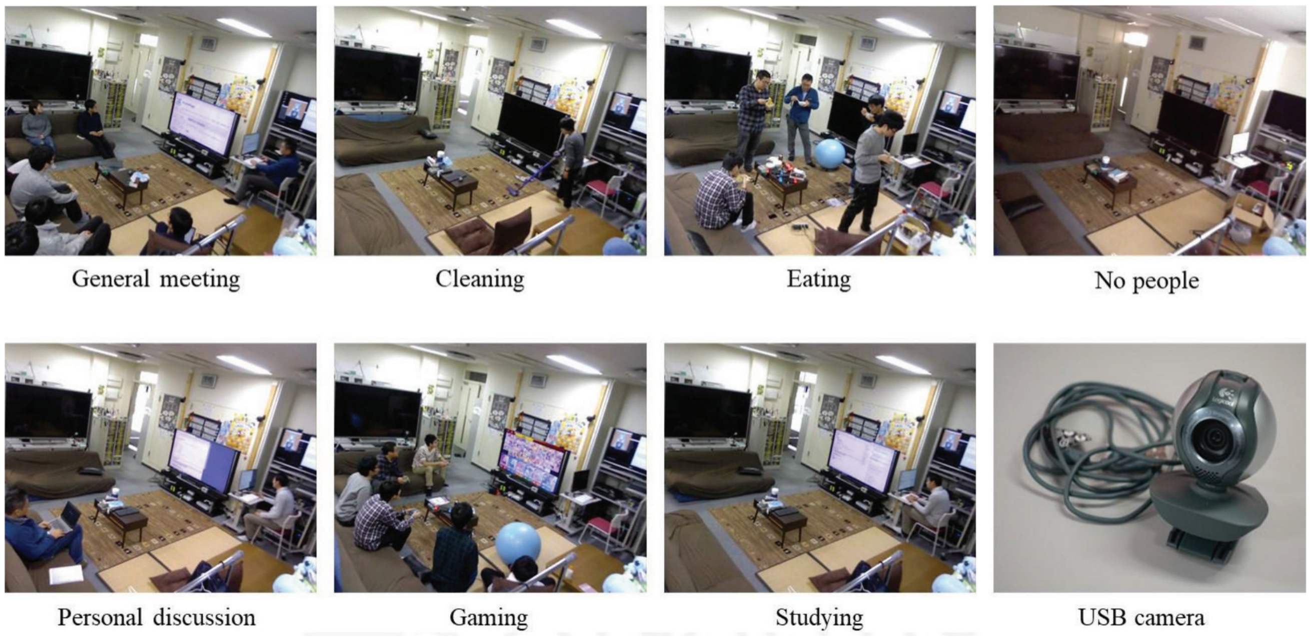


Figure 5 | The representative images for each context and USB camera.

In Step 3-I-II, considering each tag set as a document, we converted the tag sets into document vectors. For the vectorization, we used the TF-IDF method [19]. TF-IDF is a method of numerical statistic that reflects the *importance* of a word in a document, as well as in the whole collection of documents (corpus).

In Step 3-II, we randomized all created data sets, and we split them into half as training data and test data, where $\alpha = 0.5$.

In Step 3-III, we finally constructed a classifier for the seven custom contexts using the training data, using the Multi-class Neural Network algorithm. The classifier was built and deployed on a cloud platform using Microsoft Azure Machine Learning Studio (see Section 2.4).

5.3. Evaluating the Recognition Model

According to Step 3-IV, we evaluated how accurately the trained model M was able to classify every image as a document into the one of the seven contexts. For a given class c_i of a custom context, let all_i be the number of all images tested for c_i. Let tp_i be the number of images that the model M correctly classified as c_i (true-positive), Let fp_i be the number of images that M classified as c_i, but actually not (false-positive). Let fn_i be the number of images that M mistakenly classified as c_j ($i = j$) (false-negative). Then, the model was evaluated by the following metrics:

- Overall accuracy: = $\sum_{i=1}^{7}(tp_i) / \sum_{i=1}^{7}(\text{all}_i)$
- Average accuracy: = $\sum_{i=1}^{7}(tp_i / \text{all}_i) / 7$
- Micro-averaged precision: = $\sum_{i=1}^{7}\{tp_i / (tp_i + fp_i)\} / 7$
- Macro-averaged precision: = $\sum_{i=1}^{7}(tp_i) / \sum_{i=1}^{7}(tp_i + fp_i)$
- Micro-averaged precision: = $\sum_{i=1}^{7} t\{p_i / (tp_i + fn_i)\} / 7$
- Macro-averaged recall: = $\sum_{i=1}^{7}(tp_i) / \sum_{i=1}^{7}(tp_i + fn_i)$

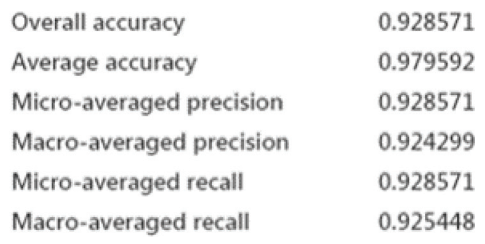
To see the performance of individual contexts, we also visualize the result with a *confusion matrix*.

5.4. Results

Figure 6 shows the evaluation metrics of the constructed model. It can be seen that the constructed model achieved quite high-quality recognition, where all the metrics marked over 0.92. Although the number of training image was just 50 for each of the seven context, the proposed method implemented a high-quality recognition model for the seven contexts in our laboratory.

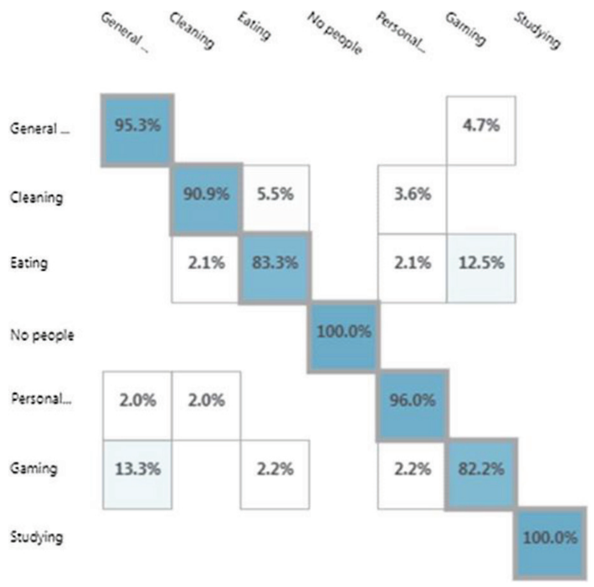
Figure 7 shows the confusion matrix, representing how individual contexts were recognized by the recognition model. From the confusion matrix, we investigated the proportion of correct and/or incorrect recognition for each context.

The recognition accuracy of "General meeting" was 95.3%, and the remaining 4.7% was incorrectly recognized as "Gaming". The accuracy



Overall accuracy	0.928571
Average accuracy	0.979592
Micro-averaged precision	0.928571
Macro-averaged precision	0.924299
Micro-averaged recall	0.928571
Macro-averaged recall	0.925448

Figure 6 Evaluation metrics of experimental results.



	General ...	Cleaning	Eating	No people	Personal...	Gaming	Studying
General ...	95.3%					4.7%	
Cleaning		90.9%	5.5%		3.6%		
Eating		2.1%	83.3%		2.1%	12.5%	
No people				100.0%			
Personal...	2.0%	2.0%			96.0%		
Gaming	13.3%		2.2%		2.2%	82.2%	
Studying							100.0%

Figure 7 Confusion matrix of experimental results.

of "Cleaning" was 90.9%, the remaining 5.5% was incorrectly recognized as "Eating", and the 3.6% was incorrectly recognized as "Personal discussion". The accuracy "Eating" was 83.3%, the remaining 4.2% was incorrectly recognized as "Cleaning, and the 12.5% was incorrectly recognized as "Gaming. The accuracy of "No people" was 100.0%. The accuracy of "Personal discussion" was 96.0%, the remaining 2.0% was incorrectly recognized as "General meeting", and the 2.0% was incorrectly recognized as "Cleaning". The accuracy of "Gaming" was 82.2%, the remaining 13.3% was incorrectly recognized as "General meeting", the 2.2% was incorrectly recognized as "Eating", and the 2.2% was incorrectly recognized as "Personal discussion". The accuracy of "Studying" was 100.0%.

5.5. Discussion

It was seen in the experimental results that the recognition of "No people" and "Studying" was perfect. The reason is that within these contexts there were few people appearing in the shared space. When the members of our laboratory were studying, they were basically on their desks, but not in the shared space. Thus, these two contexts were "easy" for the model to recognize.

On the other hand, the constructed model was relatively bad at recognizing "Eating" and "Gaming". It was interesting to observe that "Eating" was often confused as "Gaming", and that "Gaming" was confused as "General meeting". The common characteristic among these contexts were that there were multiple people in the shared space. Furthermore, these people were interacting with each other, and were often moving around dynamically. We guessed the reason why these contexts were "difficult" as follows. Taking a snapshot image in such dynamic contexts could not always capture essential features for the model to distinguish them. Even with our human eyes, it was difficult to distinguish these contexts.

To improve the accuracy for these difficult contexts, we should deploy more cameras to observe the space from various viewpoints. Introducing multiple cognitive APIs may be also a good idea. These issues will be addressed in our future work, and they are beyond the scope of this paper.

Note that our experiment was done without proprietary sensor systems nor expensive computing resources. What we needed were an ordinary PC, and an USB camera, and the Internet connection. They are already well accepted in general households. This justifies that the proposed method is affordable and practical.

6. RELATED WORK

The research of recognizing contexts and activities within smart homes has been conducted for many years in the field of ubiquitous computing. As introduced in Section 1, the technologies include an elderly watching system using human motion sensors [9], the context labeling and recognition system based on environment change [10], the recognition of daily activities with time-series environmental data [11], the activity recognition with power consumption of home appliances and user's location [12], and context sensing with smart phones [13]. Although these systems are technically interesting, they require home users to maintain dedicated sensor systems or expensive resources. Thus, we consider that it requires still some time for these systems to be widely accepted by general households.

The activity recognition with deep learning becomes a hot topic recently (e.g., Wang et al. [14] and Brenon et al. [15]). Although the deep learning is a powerful approach to recognize image data, a huge amount of data is required to build a high-quality model. Therefore, it is unrealistic for individual households to prepare a huge amount of labeled datasets for custom fine-grained contexts.

Menicatti and Sgorbissa [26] proposed a framework that recognizes indoor scenes and daily activities using cloud-based computer vision. Their concept and aim are similar to our method. However, the way of encoding tags is based on a Naive Bayes model where each word is present or not. Also, the method is supposed to be executed on a mobile robot, where the image is dynamically changed. Thus, the method and the premise are different from ours.

Research in Menicatti et al. [27] investigates the influence of person's cultural information toward vision-based activity recognition at home. The accuracy of the fine-grained context recognition would be improved by taking such personal information into machine learning. We would like investigate this perspective in our future work.

7. CONCLUSION

To achieve the home context recognition affordable for general households, this paper proposed a method that integrates the image-based cognitive API and light-weight machine learning. The proposed method considers every image as a document by using "tags" derived by the API. We first presented a framework specifying a common workflow of home context recognition. Then, we proposed a method implementing the framework with the image-as-a-document approach.

We have also conducted an experimental evaluation that attempts to recognize seven custom contexts within our laboratory. As a result, we obtained a high-quality context recognition model with the overall accuracy of 0.92. It was shown that the model was able to recognize contexts "No people" and "Studying", perfectly. On the other hand, the model was confused by contexts "Eating", "Gaming", and "General meeting", where multiple people are dynamically interacting and moving.

In our future work, we plan to investigate techniques that improve the accuracy and the performance of the proposed method. It is also important to deploy and operate the proposed method in actual households, in order to validate if the proposed method is really beneficial for individual needs. For efficient deployment and operation, we are also interested in *transfer learning*, so as to reuse already trained models in other houses and environment. Of course, when operating the system in general households, the security and privacy issues must be addressed. Developing methods for protecting data and model, as well as for presenting comprehensive informed consent is also important future work.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

ACKNOWLEDGMENTS

This research was partially supported by JSPS KAKENHI Grant Numbers: JP19H01138, JP17H00731, JP18H03242, JP18H03342, JP19H04154, JP19K02973.

REFERENCES

[1] L. Vuegen, B. Van Den Broeck, P. Karsmakers, H. Van Hamme, B. Vanrumste, Automatic monitoring of activities of daily living based on real-life acoustic sensor data: a preliminary study, Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies, Association for Computational Linguistics, Grenoble, France, 2013, pp. 113–118. https://www.aclweb.org/anthology/W13-3918

[2] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, A. Bauer, Monitoring activities of daily living in smart homes: understanding human behavior, IEEE Signal Process. Mag. 33 (2016), 81–94.

[3] M. Alirezaie, J. Renoux, U. Köckemann, A. Kristoffersson, L. Karlsson, E. Blomqvist, et al., An ontology-based context-aware system for smart homes: E-care@home, Sensors (Basel) 17 (2017), pii: E1586.

[4] Y. Ashibani, D. Kauling, Q.H. Mahmoud, A context-aware authentication framework for smart homes, 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, Windsor, ON, Canada, 2017, pp. 1–5.

[5] K. Deeba, R.A.K. Saravanaguru, Context-aware healthcare system based on IoT - smart home caregivers system (SHCS), 2018 IEEE Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, Madurai, India, 2018, pp. 1204–1208.

[6] S.C. Joo, C.W. Jeong, S.J. Park, Context based dynamic security service for healthcare adaptive application in home environments, 2009 Software Technologies for Future Dependable Distributed Systems, IEEE, Tokyo, Japan, 2009, pp. 220–224.

[7] S. Sakakibara, S. Saiki, M. Nakamura, S. Matsumoto, Indoor environment sensing service in smart city using autonomous sensor box, 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), IEEE, Okayama, Japan, 2016, pp. 1–6.

[8] L. Niu, S. Saiki, S. Matsumoto, M. Nakamura, WIF4InL: web-based integration framework for indoor location, Int. J. Pervasive Comput. Commun. 12 (2016), 49–65.

[9] M. Tsuda, M. Tamai, K. Yasumoto, A monitoring support system for elderly person living alone through activity sensing in living space and its evaluation, IPSJ SIG Technical Report, 2014-CSEC-64 (2014), 1–7.

[10] K. Tamamizu, S. Sakakibara, S. Saiki, M. Nakamura, K. Yasuda, Capturing activities of daily living for elderly at home based on environment change and speech dialog, in: V. Duffy (Eds.), Digital Human Modeling. Applications in Health, Safety, Ergonomics, and Risk Management: Health and Safety, Lecture Notes in Computer Science, vol. 10287, Springer International Publishing, Vancouver, Canada, 2017, pp. 183–94.

[11] L. Niu, S. Saiki, M. Nakamura, Using non-intrusive environmental sensing for ADLS recognition in one-person household, Int. J. Softw. Innov. 6 (2018), 16–29.

[12] K. Ueda, M. Tamai, Y. Arakawa, H. Suwa, K. Yasumoto, A living activity recognition system based on power consumption of appliances and inhabitant's location information, Inform. Process. Soc. Japan 57 (2016), 416–425.

[13] K. Ouchi, M. Doi, Living activity recognition technology using sensors in smartphone, Toshiba Rev. 68 (2013), 40–43.

[14] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: a survey, Pattern Recognit. Lett. 119 (2019), 3–11.

[15] A. Brenon, F. Portet, M. Vacher, Context feature learning through deep learning for adaptive context-aware decision making in the home, 2018 14th International Conference on Intelligent Environments (IE), IEEE, Rome, Italy, 2018, pp. 32–39.

[16] Microsoft azure, Computer vision. Available from: https://azure.microsoft.com/ja-jp/services/cognitive-services/computer-vision/ [visited on July 23, 2018].

[17] IBM watson, Visual recognition. Available from: https://www.ibm.com/watson/jp-ja/developercloud/visual-recognition.html [visited on July 23, 2018].

[18] Google ecloud vision AI. Available from: https://cloud.google.com/vision/ [visited on July 23, 2018].

[19] C. Maklin, Vectorize documents with TF-IDF, 2016. Available from: http://ailaby.com/tfidf/ [visited on February 01, 2019].

[20] Azure machine learning studio. Available from: https://azure.microsoft.com/ja-jp/services/machine-learning-studio/ [visited on February 01, 2019].

[21] Google, Cloud machine learning engine. Available from: https://cloud.google.com/ml-engine/?hl=ja [visited on February 01, 2019].

[22] Amazon, Machine learning on AWS. Available from: https://aws.amazon.com/jp/machine-learning/ [visited on February 01, 2019].

[23] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, Proceedings of Workshop at International Conference on Learning Representations (ICLR) 2013.

[24] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation. Available from: https://nlp.stanford.edu/projects/glove/ [visited on September 20, 2019].

[25] Facebookresearch, FastText: a library for efficient learning of word representations and sentence classification. Available from: https://github.com/facebookresearch/fastText [visited on September 20, 2019].

[26] R. Menicatti, A. Sgorbissa, A cloud-based scene recognition framework for in-home assistive robots, 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), IEEE, Lisbon, Portugal, 2017, pp. 1297–1304.

[27] R. Menicatti, B. Bruno, A. Sgorbissa, Modelling the influence of cultural information on vision-based human home activity recognition, 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), IEEE, Jeju, South Korea, 2017, pp. 32–38.