Evaluating Feasibility of Image-Based Cognitive APIs for Home Context Sensing

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
Email: chensinan@ws.cs.kobe-u.ac.jp, sachio@carp.kobe-u.ac.jp, masa-n@cs.kobe-u.ac.jp

***Abstract*—Cognitive API is API of emerging AI-based cloud services, which extracts various contextual information from non-numerical multimedia data including image and audio. Our interest is to apply image-based cognitive APIs to implement smart and affordable context sensing services in a smart home. However, since the existing APIs are trained for general-purpose image recognition, they may not be of practical use in specific configuration of smart homes. In this paper, we therefore propose a method that evaluates the feasibility of cognitive APIs for the home context sensing. In the proposed method, we exploit document similarity measures to see how well tags extracted from given images characterize the original contexts. Using the proposed method, we evaluate practical APIs of Microsoft Azure, IBM Watson, and Google Cloud for recognizing 11 different contexts in our smart home.**

***Index Terms*—smart home, contexts, cognitive API, document similarity**

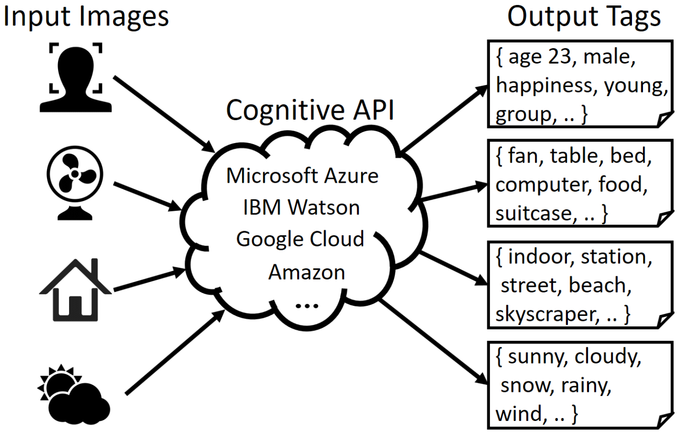
Input Images
Output Tags
Cognitive API
Microsoft Azure
IBM Watson
Google Cloud
Amazon
…
{ age 23, male, happiness, young, group, .. }
{ fan, table, bed, computer, food, suitcase, .. }
{ indoor, station, street, beach, skyscraper, .. }
{ sunny, cloudy, snow, rainy, wind, .. }

Fig. 1. Usage of image recognition APIs of cognitive services

I. Introduction

With the rapid progress of ICT and Internet of Things (IoT) technologies, research and development of *smart homes* have been actively conducted. In smart homes, it is common to use ambient and/or wearable sensors such as temperature, humidity, motion, and accelerometer in order to retrieve *contexts* of users and homes. Typical systems include an elderly watching system using human motion sensors [1], and a daily living activity sensing system for elderly people using environmental sensors [2].

Using multimedia data, such as image and audio, for home context sensing is promising for value-added smart services, since the multimedia data contains richer information than the conventional sensor data. However, recognizing multimedia data generally requires massive computation. It was thus unrealistic for general households to install and maintain such an expensive system at home.

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* (Application Program Interface), 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.

Although various kinds of cognitive APIs exist, we especially focus on *image recognition* APIs in this paper. An image recognition 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 [3] estimates age, sex, and emotional values from a given human face image. IBM Watson Visual Recognition [4] recognize items in the image such as home appliances, furniture, and tools. Google Cloud Vision API [5] outputs concept labels associated to recognized objects.

Our interest is to apply these image recognition APIs to implement smart and affordable context sensing at home. More specifically, we aim to realize a system, where a simple edge system just capturing and pre-processing images is deployed at home, and all heavy tasks of image recognition are delegated to the cognitive service in the cloud. Note, however, that the existing cognitive APIs are trained for general-purpose image recognition. Therefore, the API may not be of practical use for our specific purpose of the home context sensing.

The goal of this paper is to propose a method that evaluates the feasibility of cognitive APIs for a home context sensing. Since the detailed configuration varies from one house to another, the feasibility would be different among individual

978-1-7281-0257-3/18/$31.00 ©2018 IEEE

smart homes. However, using the proposed method, one can reproduce the experiment with different configurations. Also, one can understand the coverage and limitation of APIs towards specific home contexts.

In the proposed method, we first capture images of different contexts. Afterward, we send the image to cognitive API to retrieve tags from the images. Finally, we evaluate the performance of the APIs by checking if the tags can sufficiently characterize (or distinguish) the context shown in the original image. Our key idea of evaluation is to regard a set of tags (obtained from an image) as a document (corpus), and to apply *document similarity measures* [6] to see how clusters of contexts are constructed. More specifically, we evaluate the document clusters, with respect to the *internal cohesion* and *external isolation*. That is, we see if images belonging to the same (or different) context(s) are associated with similar tags (or dissimilar tags, respectively).

Based on the proposed method, we have conducted an experiment. In the smart home space of our laboratory, we collected images of 11 different contexts: general meeting, reading, cleaning, eating, gaming, no people, personal meeting, studying, sleeping, touching smartphone, and watching TV. We then sent the images to three cognitive APIs to retrieve tags from the images: Microsoft Azure Computer Vision API [3], IBM Watson Visual Recognition [4], and Google Cloud Vision API [5]. Finally, we analyzed the retrieved tags using *Term Frequency - Inverse Document Frequency (TF-IDF)* [7] and *cosine similarity*.

The experimental results showed that among the three cognitive APIs, there was no significant difference in the performance of the internal cohesion. Tags produced by Google Cloud Vision API tend to be more similar with each other, compared to tags produced by Microsoft Azure Computer Vision (or IBM Watson Visual Recognition). As for the external isolation, we found that background objects irrelevant to the context would produce a steady bias component. It was shown that removing the bias by subtracting tags produced by the context "no people" improved the performance of the external isolation.

II. Proposed Method

In this section, we present a method that evaluates and compares the capability of multiple image recognition APIs, for a given set of home contexts.

Figure 2 depicts the essential part of the proposed method. In the figure, $\{c_1, c_2, \cdots, c_m\}$ represent a given set of home contexts. For each context, we collect n images at home, and then send the images to cognitive APIs. Finally, we evaluate the performance of the APIs, by analyzing the output tags. More specifically, the proposed method consists of the following six steps:

Step1: Acquiring images

A user of the proposed method deploys an image capturing device (e.g., USB camera) in the target space, and configures the device to take snapshots of the space periodically with an appropriate interval.

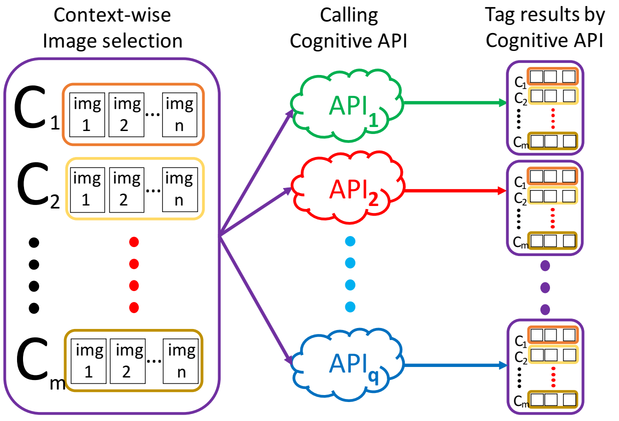

Fig. 2. The flow from context label setting to analysis of results

Step2: Defining home contexts to recognize

The user defines a set $C = \{c_1, c_2, ..., c_m\}$ of home contexts to be recognized by the cognitive API.

Step3: Selecting representative images

For each context $c_i \in C$, the user manually selects representative n images $IMG(c_i) = \{img_{i1}, img_{i2}, ..., img_{in}\}$ that well expose c_i, from all images obtained in Step 1.

Step4: Calling cognitive API

The user designates a set $API = \{api_1, api_2, \cdots, api_q\}$ of cognitive APIs to be evaluated. For every $c_i \in C$, $img_{ij} \in IMG(c_i)$, and $api_k \in API$, $api_k(img_{ij})$ is invoked, and a set $Tag(img_{ij}, api_k) = \{w_1, w_2, w_3, ...\}$ of output tags is obtained. $Tag(img_{ij}, api_k)$ represents a recognition result for cognitive API api_k for an image img_{ij} belonging to a context c_i. The size of $Tag(img_{ij}, api_k)$ varies for img_{ij} and api_k. Since there are m contexts, n images for each context, and q APIs, this step creates totally $m \times n \times q$ sets of output tags.

Step5: Analyzing output tags

Regarding every set $Tag(img_{ij}, api_k)$ of output tags as a document, the method calculates the similarity, which is denoted as '$\approx$', between any two of documents using a certain method of *document similartity measure*.

For each api_k, we evaluate the performance of api_k of context recognition, with respect to *internal cohesion* and *external isolation*. The internal cohesion represents a capability that api_k can produce similar output tags for images in the same context. That is, for $c_i \in C$, we evaluate $Tag(img_{ij}, api_k) \approx Tag(img_{ij'}, api_k)$. On the other hand, the external isolation represents a capability that api_k can produce dissimilar output tags for images in different contexts. That is, for $c_x \neq c_y$, we evaluate $Tag(img_{xj}, api_k) \not\approx Tag(img_{yj'}, api_k)$.

Regarding the calculation of *document similarity*, there exists a variety of methods in the field of natural language processing. One of most basic methods is to use TF-IDF and the cosine similarity [7]. Modern techniques include Word2Vec [8] and Doc2Vec [9]. The selection of the similarity measure is left for the user of the proposed method.

TABLE I
EXPERIMENT SETTINGS

Target space	Smart space in CS27 Nakamura Lab
Image accumulation period	7 Days
Shooting method	USB camera
Image Resolution	1280 × 1024
Number of contexts (m)	11
Number of selected images (n)	10
Number of APIs (q)	3
Document vectorization	TF-IDF
Document similarity metrics	Cosine similarity

III. Experimental Evaluation

A. Experimental Set up

Using the proposed method, we evaluate the feasibility of Microsoft Azure Computer Vision (Azure) API [3], IBM Watson Visual Recognition (Watson) [4], and Google Cloud Vision API (Google) [5], for context sensing of a smart space in our laboratory. Table I summarizes settings of the experiment.

In this experiment, we set the target space to be a smart home space, which is a part of our laboratory. For Step 1, we install an USB camera to acquire images of the daily activities of members of the laboratory. We develop a program that takes a snapshot with the USB camera every 5 seconds, and the images are cumulated in a server during one week. In Step 2, we define 11 contexts: general meeting, reading, cleaning, eating, gaming, no people, personal meeting, studying, sleeping, touching smartphone, and watching TV. In Step 3, for each context, we select 10 representative images considered to expose the context well. The selection is done by visual inspection so that the 10 images are chosen from as different date and time as possible. In Step 4, the images are sent to the three different APIs, and the total 330 sets of output tags (= 11 contexts × 10 images × 3 APIs) are obtained. In Step 5, we use TF-IDF to encode each set of output tags to a vector, and the cosine similarity to calculate the similarity.

B. Encoding Output Tags by TF-IDF

As mentioned in Step 5 in Section II, we evaluate the internal cohesion and external isolation among the sets of output tags. For the internal cohesion of API api_k for context c_i, we want to see the similarity between output tags $Tag(img_{ij}, api_k)$ $(j = 1, 2, ..., 10)$. Therefore, TF-IDF is calculated among 10 documents for each context c_i, regarding each $Tag(img_{ij}, api_k)$ $(j = 1, 2, ..., 10)$ as a unique document.

For the external isolation of API api_k, we want see how far a context c_x is from another context c_y. Therefore, TF-IDF is calculated among 11 contexts c_i $(i = 1, 2, ..., 11)$, where we join $Tag(img_{ij}, api_k)$ $(j = 1, 2, ..., 10)$ into a single document $Tag(c_i, api_k)$ that characterizes context c_i. Once a pair of documents is encoded as TF-IDF vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we define the similarity between the documents as

$$cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||}$$

which expresses the similarity by the angle formed by the vectors. When the angle is small, the cosine value is close to 1.0 (or -1.0), meaning that two documents are similar (or dissimilar, respectively).

C. Results

Table II shows the result of the internal cohesion. The values in the table show the average of cosine similarities of output tags within each of the 11 contexts, produced by the three APIs. According to the definition, the higher value represents better performance, meaning that the API can produce similar tags for images belonging to the same contexts. Although the difference was not so significant, Google marked a slightly better performance for the internal cohesion.

Table III shows the result of the external isolation. Each entry shows a cosine similarity between $Tag(c_x, api_k)$ and $Tag(c_y, api_k)$. According to the definition, the lower value represents better performance, meaning that the API can produce dissimilar tags for images belonging to different contexts. In the experiment, we found that background objects irrelevant to the context produced a steady bias component. To remove the bias, we applied a pre-processing which subtracts $Tag(``No\, people", api_k)$ from every $Tag(c_i, api_k)$. As the result of the pre-processing, some output tags produced by Google became empty, which means that the API could not distinguish the context from “No people”. Azure and Watson marked similar performance for the external isolation.

From the experiment, we obtained the following findings:

- Individual APIs have their own strong (or week) contexts.
- For home sensing with arbitrary contexts, we cannot expect too much performance for the general-purpose APIs without training.
- Depending on the target context, we should consider appropriate pre-processing of the image to improve the recognition performance.

IV. Conclusion

In this paper, we proposed a method that evaluates the feasibility of image-based cognitive APIs towards the home context sensing of smart home. Applying the document similarity measure to the output tags produced from the image, the proposed method evaluates the performance of image-based context recognition with respect to the internal cohesion and the external isolation. In the experiment, we evaluated the feasibility of three different APIs towards context sensing within our laboratory. The experimental evaluation shows that we cannot expect too much performance for those general-purpose APIs without training.

As future work, we investigate natures of cognitive APIs to identify strong or week contexts, ideal home environment, the way of capturing images, and so on. It is also important to consider appropriate pre-processing of images to improve the performance of context recognition. Integrating output tags from different APIs to further improve context classification ability is also an interesting topic.

TABLE II
RESULT REPRESENTING THE INTERNAL COHESION OF OUTPUT TAGS

API	G-meeting	Reading	Cleaning	Eating	Gaming	No people	P-meeting	Studying	Sleeping	Smartphone	TV
Azure	0.650	0.641	0.581	0.561	0.607	0.807	0.649	0.696	0.819	0.709	0.599
Watson	0.643	0.611	0.474	0.512	0.650	0.759	0.555	0.680	0.785	0.818	0.607
Google	0.679	0.780	0.727	0.564	0.821	0.738	0.708	0.670	0.784	0.844	0.690

TABLE III
RESULT REPRESENTING THE EXTERNAL ISOLATION OF OUTPUT TAGS (PRE-PROCESSING APPLIED)

Context	API	G-meeting	Reading	Cleaning	Eating	Gaming	No people	P-meeting	Studying	Sleeping	Smartphone	TV
General meeting	Azure	1.000	0.528	0.645	0.576	0.412	0.000	0.815	0.188	0.230	0.402	0.336
	Watson	1.000	0.143	0.222	0.616	0.271	0.000	0.222	0.137	0.037	0.281	0.184
	Google	1.000	0.000	0.000	0.369	0.234	0.000	0.368	0.094	0.000	0.232	0.232
Reading	Azure	0.528	1.000	0.683	0.405	0.687	0.000	0.691	0.725	0.228	0.360	0.703
	Watson	0.143	1.000	0.802	0.272	0.787	0.000	0.821	0.753	0.225	0.000	0.930
	Google	0.000	1.000	0.000	0.000	0.482	0.000	0.000	0.913	0.000	0.000	0.000
Cleaning	Azure	0.645	0.683	1.000	0.447	0.579	0.000	0.794	0.642	0.204	0.308	0.593
	Watson	0.222	0.802	1.000	0.471	0.733	0.000	0.747	0.625	0.344	0.154	0.812
	Google	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Eating	Azure	0.576	0.405	0.447	1.000	0.315	0.000	0.536	0.185	0.178	0.394	0.278
	Watson	0.616	0.272	0.471	1.000	0.371	0.000	0.374	0.395	0.145	0.191	0.322
	Google	0.369	0.000	0.000	1.000	0.567	0.000	0.548	0.268	0.000	0.659	0.659
Gaming	Azure	0.412	0.687	0.579	0.315	1.000	0.000	0.686	0.670	0.305	0.383	0.634
	Watson	0.271	0.787	0.733	0.371	1.000	0.000	0.759	0.636	0.235	0.000	0.849
	Google	0.234	0.482	0.000	0.567	1.000	0.000	0.763	0.791	0.000	0.861	0.861
No people	Azure	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Watson	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Google	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Personal meeting	Azure	0.815	0.691	0.794	0.536	0.686	0.000	1.000	0.536	0.262	0.461	0.534
	Watson	0.222	0.821	0.747	0.374	0.759	0.000	1.000	0.920	0.195	0.117	0.886
	Google	0.368	0.000	0.000	0.548	0.763	0.000	1.000	0.317	0.000	0.778	0.778
Studying	Azure	0.188	0.725	0.642	0.185	0.670	0.000	0.536	1.000	0.170	0.131	0.731
	Watson	0.137	0.753	0.625	0.395	0.636	0.000	0.920	1.000	0.207	0.000	0.798
	Google	0.094	0.913	0.000	0.268	0.791	0.000	0.317	1.000	0.000	0.408	0.408
sleeping	Azure	0.230	0.228	0.204	0.178	0.305	0.000	0.262	0.170	1.000	0.394	0.187
	Watson	0.037	0.225	0.344	0.145	0.235	0.000	0.195	0.207	1.000	0.000	0.253
	Google	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
touching smartphone	Azure	0.402	0.360	0.308	0.394	0.383	0.000	0.461	0.131	0.394	1.000	0.258
	Watson	0.281	0.000	0.154	0.191	0.000	0.000	0.117	0.000	0.000	1.000	0.158
	Google	0.232	0.000	0.000	0.659	0.861	0.000	0.778	0.408	0.000	1.000	1.000
Watching TV	Azure	0.336	0.703	0.593	0.278	0.634	0.000	0.534	0.731	0.187	0.258	1.000
	Watson	0.184	0.930	0.812	0.322	0.849	0.000	0.886	0.798	0.253	0.158	1.000
	Google	0.232	0.000	0.000	0.659	0.861	0.000	0.778	0.408	0.000	1.000	1.000

Acknowledgments: This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (B) (16H02908, 18H03242, 18H03342), Grant-in-Aid for Scientific Research (A) (17H00731)], and Tateishi Science and Technology Foundation (C) (No.2177004).

REFERENCES

[1] M. Tsuda, M. Tamai, and K. Yasumoto, "A monitoring support system for elderly person living alone through activity sensing in living space," vol. 2013, no. 16, pp. 1–5, may 2013.

[2] K. Tamamizu, S. Sakakibara, S. Saiki, M. Nakamura, and K. Yasuda, "Capturing activities of daily living for elderly at home based on environment change and speech dialog," *IEICE Technical Report*, vol. 116, no. 404, pp. 7–12, jan 2017.

[3] "Microsoft azure computer vision," https://azure.microsoft.com/ja-jp/services/cognitive-services/computer-vision/, visited on 2018-07-23.

[4] "Ibm watson visual recognition," https://www.ibm.com/watson/jp-ja/developercloud/visual-recognition.html, visited on 2018-07-23.

[5] "Google cloud vision api," https://cloud.google.com/vision/, visited on 2018-07-23.

[6] T. Kamishima, "Clustering," http://www.kamishima.net/archive/clustering.pdf, visited on 2018-07-23.

[7] T. Roelleke and J. Wang, "TF-IDF uncovered: A study of theories and probabilities," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '08. New York, NY, USA: ACM, 2008, pp. 435–442. [Online]. Available: http://doi.acm.org/10.1145/1390334.1390409

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781

[9] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, pp. II–1188–II–1196. [Online]. Available: http://dl.acm.org/citation.cfm?id=3044805.3045025