International Journal of Grid and Distributed Computing
Vol. 11, No. 6 (2018), pp.69-78
http://dx.doi.org/10.14257/ijgdc.2018.11.6.07

Real-time Calorie Extraction and Cuisine Classification through Food-Image Recognition

Hye-Jun Suh[1] and Kang-Hee Lee[2*]

[1]Global School of Media, Soongsil University, Seoul, South Korea
[2]Global School of Media, Soongsil University, Seoul, South Korea
*[1]hj@soongsil.ac.kr, [2]*kanghee.lee@ssu.ac.kr*

Abstract

The self-monitoring of user-activity tracking is the most common method of both individuals and most health-care systems. A growing demand for services that can easily monitor food and calorie information through the use of food photographs has also emerged. In this paper, an existing food-recognition and calorie-extraction system is combined with a context-recognition system that recognizes the meal type. To recognize the food images, a preceding Tensorflow-based machine-learning process was performed, and an Expert.js-based semantic network was constructed for the meal-type recognition. The food-recognition accuracy rate is 55.3 %, and Korean, Chinese, and Italian food were recognized. As a result, the objective is the combining of the context awareness with the existing self-monitoring systems to enable the user to implement dietary adjustments.

Keywords: *Vision system, Cuisine classification, Calorie extraction, Image recognition, Machine learning*

1. Introduction

Health care is one of the most important social issues, and to improve personal health, people engage in exercise or dieting. The tracking of user activities through self-monitoring is the most common practice and is included in most health-care systems [1]. A system that can be easily self-monitored through the use of a vision system also exists [2, 3]. In this paper, a food-classification procedure is combined with an existing self-monitoring system using the vision system. By analyzing the food in real time through photos of the food that the user eats, the calories are extracted and the type of food (Korean, Chinese, *etc.*,) that the user has consumed is classified according to a maximum of three extracted food names. Accordingly, the service can recommend food depending on the context, thereby helping the dietary control of the user.

2. Design of the System

In this paper, the functions of the food-type extraction in the existing calorie-extraction system are combined through the food-recognition process. Claus (a classifier that uses real-time images) recognizes the photos in the vision system in real time, extracts the food names and the calorie data in real time, classifies the food types in the classification system using the extracted food-name data, and the user interface (UI) makes its structure directly visible to the user. As shown in Figure 1, the system consists of a vision system, a classification system, and the UI. The overall system is made up of a Web platform that is easy to access. To construct the whole system, HTML and

Received (January 5, 2018), Review Result (March 9, 2018), Accepted (March 12, 2018)
* Corresponding Author

ISSN: 2005-4262 IJGDC

Javascript are used, while real-time client-server data transmissions allow Ajax to materialize the asynchronous UI, as shown in Figure 2.

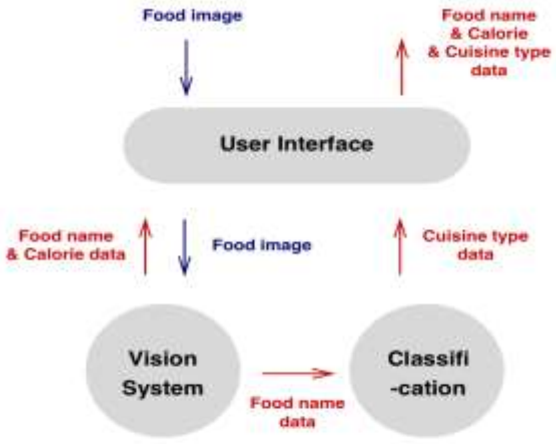


Figure 1. Structure of the System

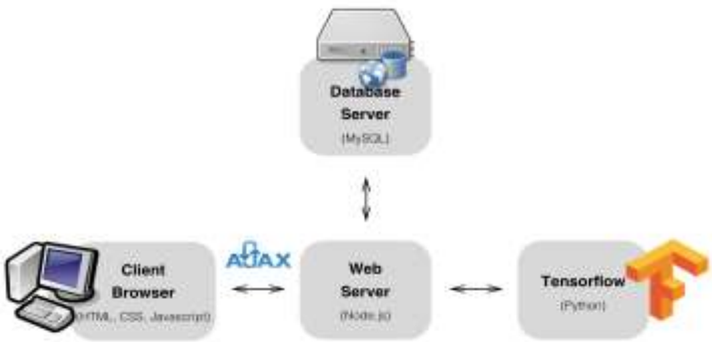


Figure 2. Data Transfer Between the Client and the Server

A MySQL database is used to store the user data. The database consists of the following two tables: a user table for the storage of the user ID and password, and an ItemTbl table for the recording of the user food name, ID, date, and calories.

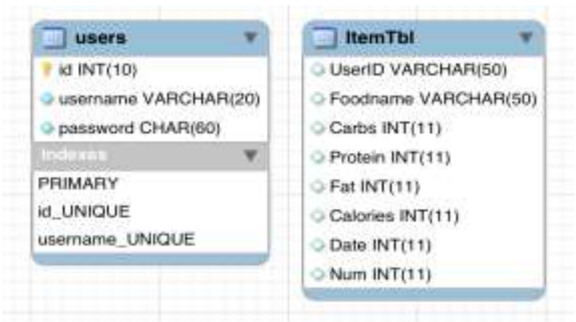
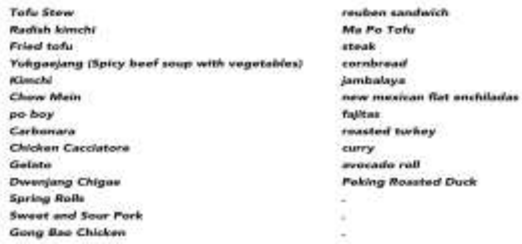


Figure 3. Components of the Tables

The vision system aims to recognize the names of foods and the corresponding calorie information through its attainment of the user food images. For this information, it is necessary to construct a classification model using machine learning. To construct the classification model, the training data are needed, and the food data corresponding to the list of Figure 4 are collected and used.



Tofu Stew	reuben sandwich
Radish kimchi	Ma Po Tofu
Fried tofu	steak
Yukgaejang (Spicy beef soup with vegetables)	cornbread
Kimchi	jambalaya
Chow Mein	new mexican flat enchiladas
po boy	fajitas
Carbonara	roasted turkey
Chicken Cacciatore	curry
Gelato	avocado roll
Dwenjang Chigae	Peking Roasted Duck
Spring Rolls	.
Sweet and Sour Pork	.
Gong Bao Chicken	.

Figure 4. Trained Food List

The goal of machine learning is the programming of computers so that example data or past experience can be used to solve a given problem [4], as shown in Figure 5.

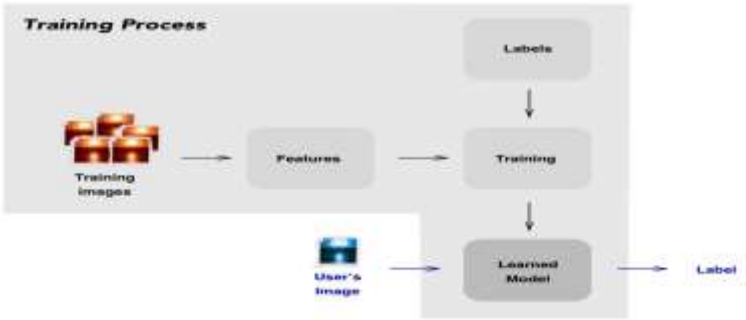


Figure 5. Process of Machine Learning

Tensorflow, a system that allows machine-learning practitioners and researchers to experiment with new techniques, was used for the machine learning [5]. By using Tensorflow, it was possible to develop a machine-learning system that can be processed in a simpler and faster way, and thus, it is possible and easy to develop the real-time interactions into a website.

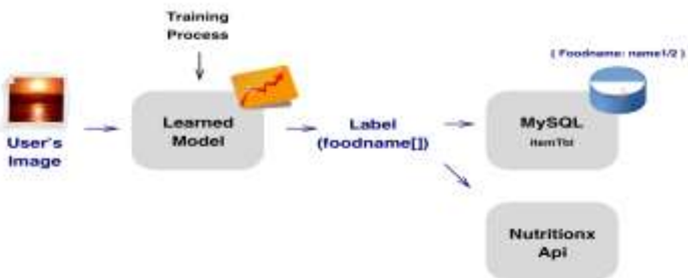


Figure 6. Structure of the Vision System

As shown in Figure 6, the food is recognized through photographs that are sent to the Nutritionx API with the recognized names in an array. The Nutritionx API returns the nutrient data that consist of the total calories and the essential nutrients according to the name of the food. The nutrient data is sent directly to the UI, and the name of the food is also sent to the UI and stored in the user-specific database that is mainly used in the classification system. An example of the data that are stored in the database is shown in Figure 7. The food names are stored in one of the variables that are separated by slashes (/), which are again separated later by the JavaScript split function. The unit of all of the nutrients except for the calories is grams (g).

UserId	"id"
Foodname	"Food1/Food2/Food3"
Carbs	30
Protein	5
Fat	10
Calories	150
Date	20170201
Num	3

Figure 7. Illustrative Data of the Database

The classification process uses the names of the foods that are recognized by the vision system to derive the food types, and it stores the combinations and the types in the database. To do this, a miniature semantic-network framework called Expert.js is used [6]. Expert.js is a framework that provides digital-subscriber-line (DSL)-like constructs for the construction of semantic networks [6]. By using Javascript as the scripting language, it is easy to develop this framework. In Expert.js, the following two concepts are used to specify the rules: isa and example, where the isa is the parent label of the label, and the example is the inverse of the isa. For example, the isa of "cat" and "dog" is "mammal," and the examples of "mammal" are "cat" and "dog." By using the label name that is learned in the vision system, the upper labels that are included in the label are designated using the isa relationship and the example, as shown in Figure 8. When the user uses the vision system, it invokes a top label (isa) with the food name that is received in real time as a parameter. The process is shown in Figure 9.

```
korean
    .example(TofuStew)
    .example(Radishkimchi)
    .example(Friedtofu)
    .example(Yukgaejang)
    .example(Kimchi)
    .example(Dryseaweed)
    .example(Dakgangjeong)
    .example(Bulgogi)
    .example(BudaeChigae)
    .example(DwenjangChigae)
    .example(SpicyChicken);
```

Figure 8. Definition of Classify.js using the isa and example Relationships

Figure 9. Structure of the Classification System using Expert.js

For a maximum of three food labels per photograph, each food type is derived through the isa, and the label to which a large number of foods belong is estimated as a meal type.

This experiment classifies the following three types of meals: Korean food, Chinese food, and Italian food.

The UI allows the user to view the data on the nutritional facts and the meal classification without the need to refresh. Figure 10 is a list of the information that is displayed on the UI.

Foodname	List of foodnames
Carbs	(g)
Protein	(g)
Fat	(g)
Calories	(Kcal)
Cuisine Type	National cuisine type

Figure 10. Data that is Displayed on the User Interface (UI)

3. Implementation

HTML, CSS, Javascript, and Node.js were used to implement the website, MySQL was used to store the database, and Ajax was used for the asynchronous transmission that facilitates the communication between the client and the server. The file structure that constitutes the whole system is shown in Figure 11.

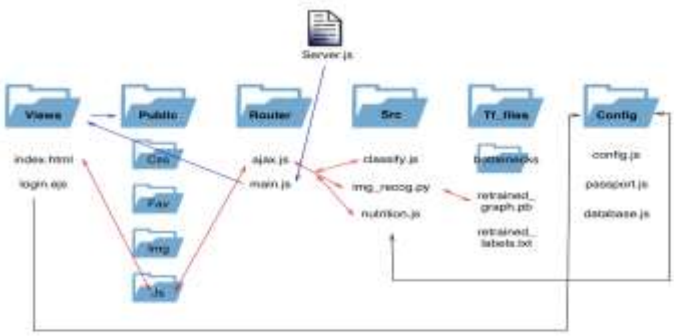


Figure 11. File Structure of the System

For the implementation of the system, a database construction must be implemented beforehand. The two tables in the MySQL database were created using the components that are shown in Figure 3 of Chapter 2, while Figure 12 shows the generated tables.

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	NO	PRI	NULL	auto_increment
username	varchar(20)	NO	UNI	NULL	
password	char(60)	NO		NULL	

3 rows in set (0.00 sec)

Field	Type	Null	Key	Default	Extra
UserID	varchar(50)	YES		NULL	
Foodname	varchar(50)	YES		NULL	
Carbs	int(11)	YES		NULL	
Protein	int(11)	YES		NULL	
Fat	int(11)	YES		NULL	
Calories	int(11)	YES		NULL	
Date	int(11)	YES		NULL	
Num	int(11)	YES		NULL	

8 rows in set (0.00 sec)

Figure 12. Generated Tables using MySQL

Tensorflow was used to train the vision system. Tensorflow machine learning consists of the following three steps, which are shown in Figure 13: training-image exploration, bottleneck generation, and step-by-step training. As a result of the machine learning, an accuracy of 55.3 % was generated, as shown in Figure 14.

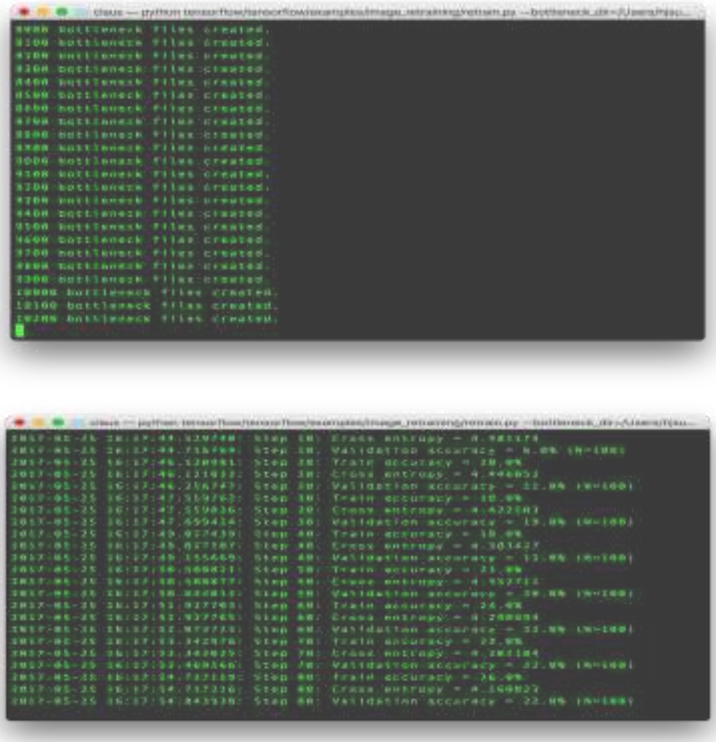


Figure 13. Process of the Tensorflow Supervised Learning

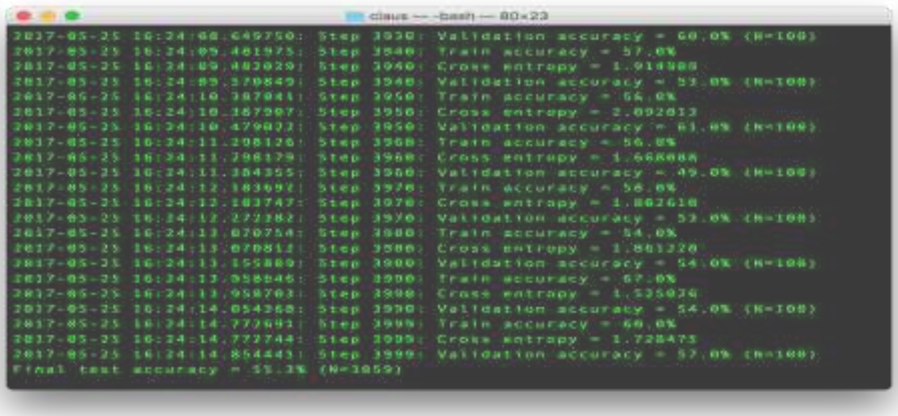


Figure 14. Accuracy of the Tensorflow Supervised Learning

The vision system is the first part of the user website access. The photos that are taken by the webcam are transmitted to the server in real time through Ajax, as shown in Figure 15. The server classifies the transmitted images using the learned classifier and stores the results in the database. The Nutritionx Api is contacted simultaneously, and the data that are stored in the database via the Nutritionx Api are shown in Figure 16. The vision system recognizes the foods, as shown in Figure 17.

Figure 15. Execution Screen of the Vision System


```
- {
      UserID: "hi",
      Foodname: "jerky/chocolate bar/fortune cookies",
      Carbs: 35,
      Protein: 10,
      Fat: 18,
      Calories: 348,
      Date: 20170523,
      Num: 2
  },
```


Figure 16. Food and Calorie Data that are Stored in the Database

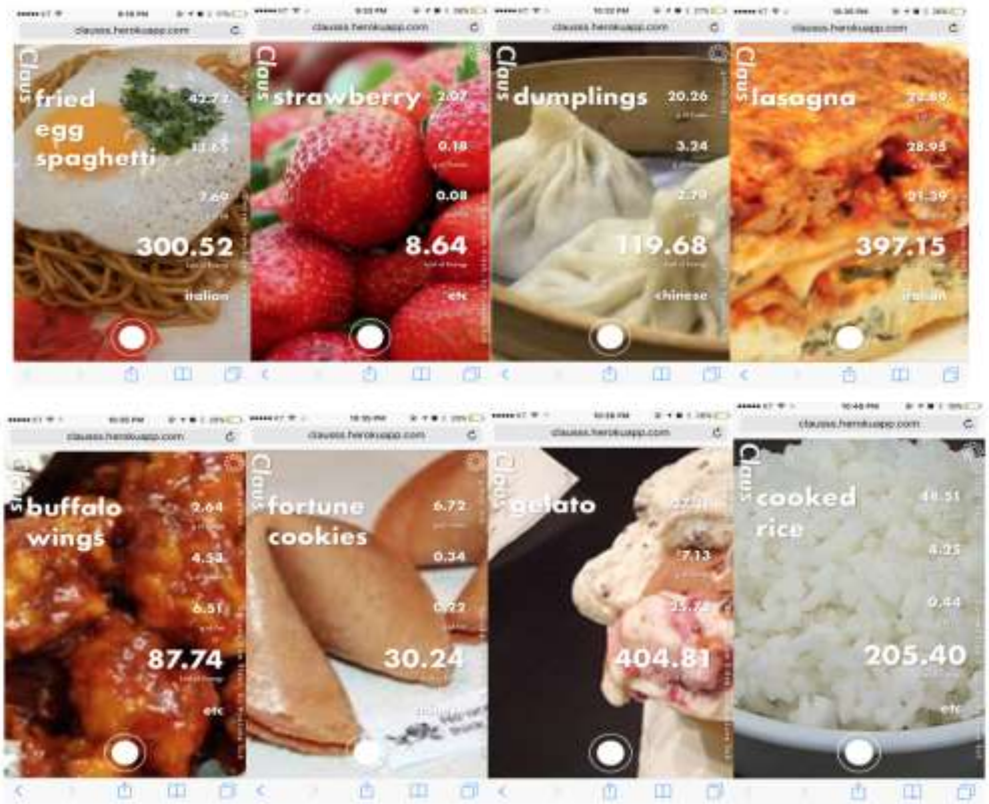


Figure 17. Recognition Results of Various Foods

In the classification process, the food names that are recognized by the vision system are classified and sent to the UI. The classification rules are declared via Expert.js. In the ItemTbl table of the database, the Foodname element is stored into a JavaScript array as “FoodNameArray,” and it is divided into “/” using the split function, as shown in Figure 18. Subsequently, the isa element of each food name with the data of each room in the FoodNameArray is stored in Classify.js as the parameter, as shown in Figure 19. Figure 20 shows the food-classification results.

```
var sqlQuery = "INSERT INTO itemtbl SET ?";
var post = {userid : req.user.username, Foodname : foodname[0] +"/"+foodname[1]+"/"+foodname[2]};
var query = connection.query(sqlQuery, post, function (err, result) {
  if (err) throw err;
  console.log("1 record inserted");
});
```

```
function classify(req) {
  connection.query('SELECT Foodname from itemtbl where userid=?',[req.user.username], function(err, rows) {
    if(err) throw err;
    num = rows.length-3;
    console.log("rows.length-3: "+num);

      FoodNameArray[0] = rows[(rows.length-1)];

    for (var i = 0; i < 3; i++) {
      isaArray[i] = FoodNameArray[0].foodname.split('/')[i].replace(/\s/gi, "");
      console.log(isaArray[i]);
    }
```

Figure 18. Codes for Storing the Foodnames

```
for (var i = 0; i < 3; i++) {
  isaArray[i] = FoodNameArray[0].foodname.split('/')[i].replace(/\s/gi, "");
  console.log(isaArray[i]);
}

for (var i = 0; i < 3; i++) {
  res[i] = isa( eval(isaArray[i]));
  res[i] = _.map( res[i], function(c){ return c.id; });
  res[i] = JSON.stringify(res[i]).replace(/\[/g,'').replace(/\]/g,'').replace(/\"/g,'');
```

Figure 19. Codes to Find the isa Element of the FoodNameArrays

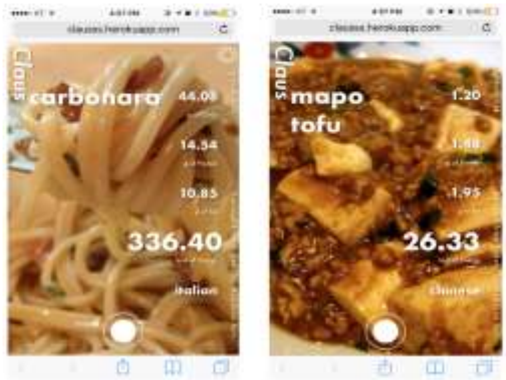


Figure 20. Results of the Cuisine Classification

The UI displays the name, calorie, and type of food in real time without the need to refresh. To receive this data, the user is encouraged not to refresh through the Spinner. Figure 21 shows the UI including the Spinner. The user can take food pictures using the website webcam screen. The food images that are taken by the user are recognized, classified, and calorie-extracted, and the food names, nutritional facts, and meal type are displayed on the UI.

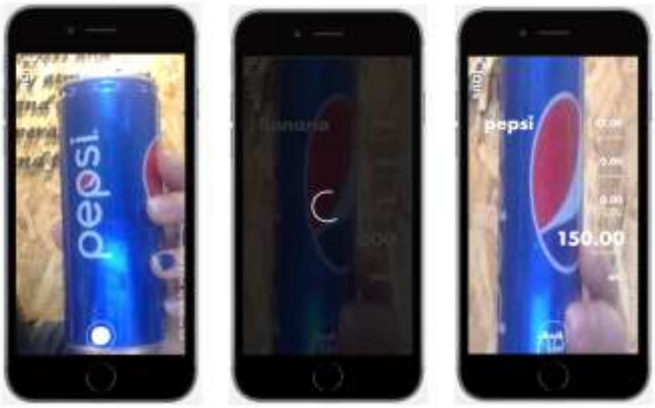


Figure 21. User Interface

4. Conclusion

In this paper, the function of a food-type classification is combined with a calorie-extraction monitoring system using food photographs. The system was trained to recognize food images using machine learning, and the food type was classified with a semantic-network framework using the "isa-example rule." This framework allows a simple photograph-based recognition of the context of the user food type. In the future, an improvement of the user-diet control for which the context recognition is combined with the existing self-monitoring systems will be explored so that, for example, the food context can be recognized and the context-appropriate foods can be recommended.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A1B05036028). This paper is a revised and expanded version of a paper entitled "A Study on Open Source Image Recognition using Deep Learning" presented at 13th International Interdisciplinary Workshop, Aug. 20-22, International Center, Jeju National University, Jeju Island, Korea, 2017.

References

[1] Klasnja, P., Consolvo, S., Pratt, W., "How to evaluate technologies for health behavior change in HCI research," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3063-3072; ACM, May **(2011).**

[2] Kawano, Yoshiyuki, Keiji Yanai, "Real-time mobile food recognition system," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, **(2013).**

[3] Shroff, Geeta, Asim Smailagic, and Daniel P. Siewiorek, "Wearable context-aware food recognition for calorie monitoring," Proceedings of the 12th IEEE International Symposium on. IEEE, Wearable Computers (ISWC), **(2008)**

[4] Alpaydin, Ethem, "Introduction to machine learning," MIT press, **(2014).**

[5] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Kudlur, M., "TensorFlow: A system for large-scale machine learning," Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, Georgia, USA, Nov. **(2016).**

[6] https://github.com/ithailevi/expert/, Jan. 25, **(2013).**

[7] Tilkov, Stefan, and Steve Vinoski, "Node. js: Using JavaScript to build high-performance network programs.," IEEE Internet Computing 14.6, **(2010),** pp. 80-83.

[8] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A, "Inception-v4, inception-resnet and the impact of residual connections on learning," arXiv preprint arXiv:1602.07261, **(2016).**

[9] T. Maruyama, Y. Kawano, and K. Yanai, "Real-time mobile recipe recommendation system using food ingredient recognition," Proceedings of ACM Multimedia Workshop on Interactive Multimedia on Mobile and Portable Devices, **(2012),** pp. 27–34.

[10] Kawano, Yoshiyuki, and Keiji Yanai, "Foodcam: A real-time food recognition system on a smartphone," Multimedia Tools and Applications, Vol. 74 , No. 14, **(2015)**, pp. 5263-5287.
[11] Mitchell, Tom Michael. "The discipline of machine learning," Carnegie Mellon University, School of Computer Science, Machine Learning Department, (**2006**).
[12] Javidi, Bahram. "Image recognition and classification: algorithms, systems, and applications," CRC Press, **(2002).**

Authors

Hye-Jun Suh, she is s an undergraduate student in Global School of Media, Soongsil University, Seoul, Korea. Her current research interests include the areas of artificial intelligence, human-computer interaction, and deep learning. Contact her at hj@soongsil.ac.kr.

Kang-Hee Lee, he received the B.S., M.S., and Ph.D degrees in electrical engineering and computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1999, 2001, and 2006, respectively. Since 2006, he has been a Senior Engineer in Digital Media & Communication Research Center, Samsung Electronics Company, Ltd., Korea. He has been a dispatched researcher in the Robotics Institute, Carnegie Mellon University in 2008. Since moving to Soongsil University in 2009, he is with Global School of Media, Soongsil University, Seoul, Korea. His current research interests include the areas of ubiquitous robotics, evolutionary robotics, emotional robotics, media robotics, cognitive task planning system, and knowledge-based reasoning system. Contact him at kanghee.lee@ssu.ac.kr.

