REAL-TIME EATING ACTION RECOGNITION SYSTEM ON A SMARTPHONE

Koichi Okamoto and Keiji Yanai

Department of Informatics, The University of Electro-Communications, Tokyo
{okamoto-k,yanai}@mm.inf.uec.ac.jp

ABSTRACT

Recently, many mobile applications to record everyday meals for dieting have been popular. Some of them can recognize names of food items in meals by only taking photos. However, such image-recognition-based food recording systems requires taking meal photos before eating, which are not applicable for the meals in which the amount of food to be eaten is not decided before eating such as large platter for sharing and barbecue-style dishes.

Then in this paper, we propose a mobile real-time eating action recognition system. It continuously recognizes user's eating action and estimates the categories of eaten food items during mealtime. With this system, we can get to know total amount of eaten food items, and can calculate total calories of eaten foods even for the meals where the amount of foods to be eaten is not decided before starting eating.

The system implemented on a smartphone continuously monitor eating actions during mealtime. It detects the moment when a user eats foods, extract food regions near the user's mouth and classify them. In the experiments, we implemented a mobile system the target of which is Japanese-style "Yakiniku" where people eat meats and vegetables while grilling. It can recognize five different kinds of ingredients for "Yakiniku" such as beef, carrot and pumpkin in the real-time way. It has achieved 74.8% classification rate, and was evaluated as being superior to the baseline system which employed no eating action recognition by user study.

Index Terms— Mobile Food Recognition, Eating Action Recognition, Food Recording System, Yakiniku

1. INTRODUCTION

In recent years, due to a rise in healthy thinking on eating, many people take care of eating and foods, and some people record daily diet regularly. To assist them, many mobile applications for recording everyday meals have been released so far. Some of them employ food image recognition, which enable users to record daily foods only by taking photos. As a commercial application, FoodLogApp [1] is a representative one, which recognizes food photos taken by users [1]. Because it employs image-search-based image annotation method for food photo recognition, users have to annotate their photos for a certain time after starting using of the application, and, however, the number of foods to be recognized is not limited. Kawano *et al.* proposed a mobile food recording system which has a 100-food recognition engine [2]. Since the recognition engine employs the state-of-the-art image recognition method, Fisher Vector [3] and liner SVM classifiers, the classification rate is relatively high. The top-5 classification rate for 100 classes is 79.2%. However, both of the systems require taking a meal photo before eating, and all the foods taken as a photo have to be served before eating. The mobile food recognition applications which assume all the foods are pre-served is not applicable for the case that the amount of food eaten by one person is not decided before eating such as sharing large dishes or barbecue-style meal (grilling meats and vegetables while eating).

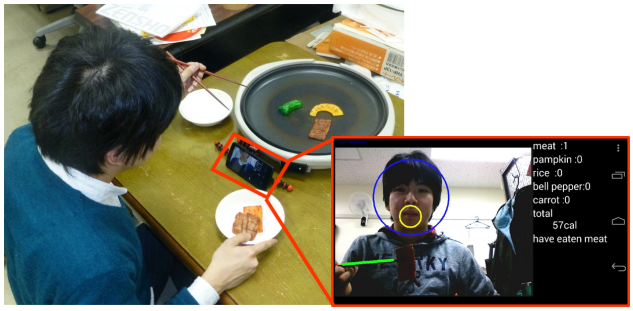


Fig. 1. A typical usage of the proposed system. A smartphone on which the proposed system is running is put toward a user's face. It continuously detects eating action and estimates food categories of eaten foods.

To handle such situations, in this paper, we propose a new-style food recording system which recognizes eating actions continuously during having a meal instead of recognizing a meal photo taken before eating. We implement a system to recognize eating actions in a video stream on a smartphone. Every time a user conveys one food item such as a piece of a grilled meat and vegetable to his/her mouth, the system recognizes and records the eating action. In addition, the system recognizes categories of eaten foods as well. Based on the information on what kinds of foods and how many times a user eats, the system can calculate an accumulative calorie the user took in the real-time way and show it on a smartphone screen. This is completely a new type of a mobile food recording system. Figure 1 shows a typical scene when we use the proposed system. In this scene, the user is having a "Yakiniku" meal which is a Japanese-style barbecue of grilling thin-sliced beef and vegetables on the hot plate. With the proposed system, users can enjoy meals as checking their current caloric

[1] http://app.foodlog.jp/

intakes.

The rest of this paper is organized as follows. Section 2 mentions related works to our work, and Section 3 overviews the proposed system. Section 4 describes the detail of the method to recognize eating actions and eaten food items. Section 5 explains an implementation of the proposed system on an Android smartphone, and Section 6 describes experimental results including evaluation of recognition accuracy and user study for the whole proposed system. Finally, Section 7 concludes this paper.

2. RELATED WORK

2.1. Food Recognition

Food recognition is a challenging task, because so many categories of foods exist, and diversity in their appearances is very large even in the same category. This is a kind of a fine-grained image recognition problem.

Joutou *et al.* [4] showed that it was important to select effective features and integrate them for food recognition. From their results, color features are effective for food recognition. According to their results, we will carefully select and combine features to recognize food items.

Matsuda *et al.* [5] performed food region segmentation and recognized the region of each food item one by one in order to perform accurate food recognition. The region estimation was accurately carried out in this paper. However, Because food region segmentation is too time-consuming to do in the real-time way, we cannot use this method. The system we propose in this paper detects a piece of food being carried toward a user's mouth region based on background subtraction.

2.2. Food Recording System

The previous two papers mentioned only food recognition engines, which is assumed to run on a PC. Recently, some food record systems employing a food recognition engine have been proposed. The representative ones are "Foodlog" [1, 6] and "FoodCam" [2, 7]. However, these systems assume that all the food for one person is served on the table before eating, which cannot be applied to the case of sharing a large platter or eating while grilling.

"FoodLog" is a server-based food logging service. In the every mealtime, a user takes meal photos and send them to the FoodLog server. On the server-side, uploaded meal photos are analyzed to estimate the ratio of basic food components such as meat, vegetable and grain in the given photo. Basically, image analysis in "FoodLog" is intended to estimate not the names of food items but nutrition balance. They estimated basic food components from low-level features directly. Therefore, detailed food categories are not recorded in the personal food history automatically. If needed, a user have to enter the name of food items manually.

On the other hand, "FoodCam" is a client-based mobile food recognition and recording system, which can classify 100 food categories in the real-time way. Because it employs image classification of 100 food categories, it can recognize the names of food items in the given photo without external recognition servers. The latest-version of "FoodCam" [2] adopted Fisher Vector [3] as feature representation, while the old-version of "FoodCam" [7] adopted Bag-of-features and explicit feature maps [8]. For simplicity of implementation, we use the latter method for recognizing categories of food item in our system.

3. SYSTEM OVERVIEW

As shown in Figure 1, we implement the proposed system on a smartphone. When we use it, we stand it with slight tilt in front of a user who is eating so that the built-it camera of the smartphone faces to the user's face. It is important to be able to capture the situation clearly by the camera that a user's hand is carrying food items to his mouth with chopsticks or a fork. It does not need any calibration and any conditions except for how to place the smartphone in which the system is running.

We cut out a frame to detect food items from the video stream being recorded by the smartphone camera, and apply face, mouth, and chopsticks detectors and food classifiers. We show the screen-shot of the proposed system in Figure 6.

The proposed system performs eating action recognition according the following processing flow .

1. Detect a user's face and mouth from the video stream recorded by a smartphone camera continuously
2. Detect and track chopsticks after a face is detected.
3. Cut out a region around the tip of the detected chopsticks at the moment when the tip of the chopsticks is approaching the detected mouth.
4. Recognize a food item category for the cut region.
5. Calculate and accumulate food calories and display it on the screen,

We show a typical usage of the system in Figure 1, and processing flow in Figure 2.

In the current implementation, we do not estimate the volume of each food item. Instead we use the pre-defined standard calorie values on each food item for estimating the total calorie of eaten food items.

4. DETAILS OF THE PROPOSED SYSTEM

4.1. Mouth Detection

First, we detect a mouth of a user. We use a cascaded classifier implemented in the OpenCV library. This method is basically the same as Viola-Jones face detector [9]. This method provides about hundred simple classifier trained by AdaBoost, and integrated them in the cascade way. Also each classifier perform authenticity determination of whether the mouth. By using the integral image at the time of classification, calculation can be performed at high speed.

However, just applying the detector to the entire image simply causes false positive detection. Therefore, we detect

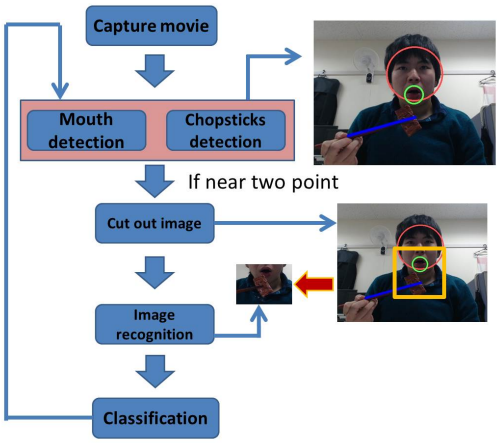


Fig. 2. The processing flow of the propose system.

Fig. 3. Face detection before(left), after(right) at time of mouth detection.

face first by applying face detector for an entire image. We use a cascade classifier in the same manner as described above in this face detection, and we apply a mouth detector for only the face region.

Nevertheless, the system sometimes erroneously detects an eye region as mouth. So it is with increased precision by detecting the mouth is limited to the lower half of the face even in the face area.

We show comparison results obtained by using the face detection in Figure 3.

4.2. Chopsticks Detection

Next we do the detection of chopsticks. Chopsticks are basically a straight line. We use the Hough transform of OpenCV in the straight line detection. It can be detected as a line segment from the image by performing probabilistic Hough transform. However, would detect a straight line from the background which is also given a function to the entire image, resulting in the erroneous detection. So that increased

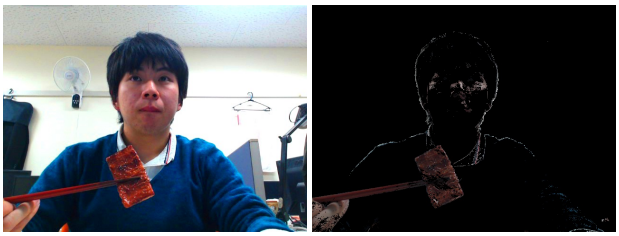


Fig. 4. Ssing the detection of the front part before(left), after(right).

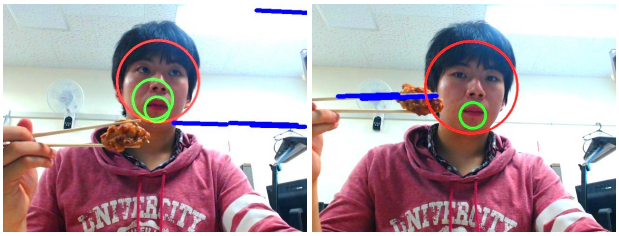


Fig. 5. Chopsticks detection, using the detection of the front part before(left), after(right).

precision by detecting the straight line that portion only extracts the front portion.

Using OpenCV functions to "BackgroundSubtractorMOG2" is also in the detection of the front part. It acquires a background image, thereby enabling extraction of the front portion by the difference between the current image and the background image. There is no problem to run background even if changed by any chance, while running because it always update the background image. However, the background becomes black by making a front portion extracted . So we would detect a straight line erroneous due to noise is pronounced. Deemed chopsticks straight line length or less certain and position certain to prevent this. So execution speed and accuracy can be improved. We show comparison results obtained by using the detection of the front part in Figure 5.

4.3. Food Classification

Because, CPUs for mobile devices is not so high-performance as CPU for PCs, the image recognition method employed in the proposed system should be light. Then one from the local features is a mainstream current image recognition, and one from the color histogram.

SIFT and SURF feature is mainly in local feature. It was also implemented using SURF feature in preliminary experiments.It was not sufficient processing speed,so we used ORB feature quantity proposed Rublee et al development system.

Less accurate because it is written in binary code than the feature quantity of the above, but it is a very fast calculation and has the advantage of strong change scaling and rotation. In this system, the value of the image it is represented as a histogram with a Bag-of-Features the ORB feature amount obtained.

Color image feature is also important for food image recognition. Therefore, we use HSV color histogram as well.

As a classifier, we use a linear kernel SVM, and we adopt the one-vs-rest strategy for multi-class classification. Although a liner SVM is fast and memory effective compared to a non-linear SVM, its classification accuracy is not as good as a non-linear kernel SVM. To compensate weakness of a liner SVM, in this paper, we adopt explicit kernel feature maps of a χ^2 kernel proposed by Vedaldi *et al.* [8]. This can be computed so that the dimension of mapped feature vectors are 3 times as many as the dimension of original feature vectors according to the following equations:

$$\phi(x) = \sqrt{x} \begin{bmatrix} 0.8 \\ 0.6\cos(0.6\log x) \\ 0.6\sin(0.6\log x) \end{bmatrix} \quad (1)$$

By introducing this feature maps, the perform ace of a linear SVM will be boosted to almost the same as the performance by a non-linear kernel SVM, while the total computational cost of a linear SVM and feature maps is still much cheaper than the cost of a non-linear SVM.

5. IMPLEMENTATION OF THE SMARTPHONE

As an implementation target, we selected an Android smartphone. In the experiments, we used Google Nexus 5 (2.3GHz Quad Core, Android 4.4).

Although it is common for image recognition application for a smartphone to use a backside camera, we use an inner camera to record user's action by the camera and show information of eaten food items to the user at the same time as shown in Figure 1.

The screenshot image of the UI of the proposed application is shown in Figure 6. A user can always check what the system is recognizing at the time regarding face, mouth and chopsticks on the left side of the screen, while on the right side of the screen total calorie and the number of eaten items are displayed. A user can check how much calorie he have taken while eating. We believe that this helps prevent people from eating too much.

Note that to estimate food calorie precisely, we have to estimate the volume of the detected food items. For simplicity, in this paper, we do not estimation food volumes, and instead we use the pre-defined fixed calories on each category of food items.

6. EXPERIMENTS

In the experiments, we evaluate classification accuracy and the system usability.

6.1. Target meal scene

We chose "Yakiniku" meal as a target domain for the experiment in the paper. "Yakiniku" is a Japanese-style barbecue meal to eat baked sliced meats and vegetables. In a typical-style Yakiniku, meats and vegetables are bakes on the hotplate or grilled on the barbecue stove.

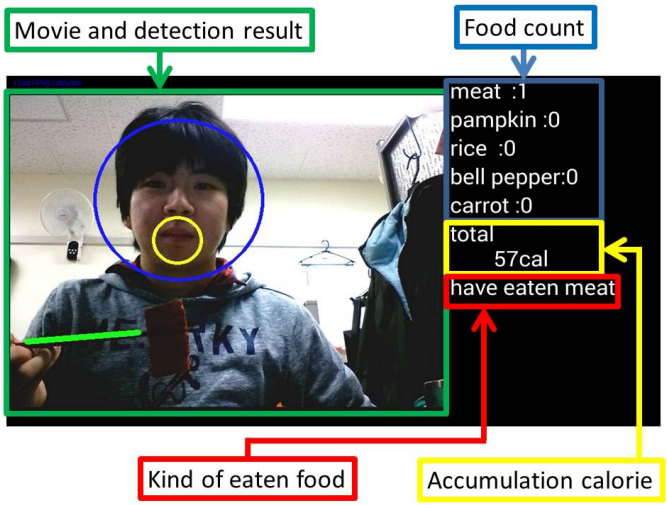


Fig. 6. The screen image of the system.

For the implementation, we selected the following five typical food items in Yakiniku:

1. meat
2. rice
3. pumpkin
4. bell pepper
5. carrot

We stored typical calorie values on the above food items on the system to calculate total calorie of eaten food items.

6.2. Accuracy evaluation

As training data images, we prepared 150 photos of actual food items and 300 photos of plastic food samples for each of the five food items. As a results, the training dataset consisted of 2250 images. For testing, we prepared 50 test images for each category as well.

Image feature used in the experiments are the following five ones.

[A] ORB feature with linear kernel

[B] ORB feature with explicit feature maps

[C] HSV color histogram with linear kernel

[D] Combination of ORB feature and HSV color histogram

[E] Combination of ORB feature value + HSV color histogram with explicit feature maps

Table 1. Accuracy evaluation and processing time.

Features	Accuracy(%)	time[ms]
A	58.8	30.9
B	60.0	94.6
C	33.6	0.01
D	64.8	29.6
E	74.8	94.4

The result is shown in Table 1. Since we also measured the time required for the recognition of each feature at the same time as the accuracy of the feature, the processing times are shown in the table. As results, E achieved the best result. Then, we show the confusion table of the result E in Table 2. From this table, we can see "pumpkin" tends to be confused with other items.

Although the accuracy was the lowest in case of single color histogram, it helps increase the accuracy of ORB about 5% by combination. Regarding the result of the explicit feature maps, it boosted the accuracy in case of single features, while it boosted about 10% in case of the combination of ORB and HSV color histogram.

By introducing feature maps, the execution time increased about three times. However, because 94.4ms is still fast, increase of the execution time is not a problem.

This time replaced by a food sample but recognition accuracy is increased and there is an image of the actual meal scene. So it is a major challenge of building data set and collection of more realistic data in the future.

6.3. User study

The evaluation was obtained asked to use the system in five users. The evaluation asks them to pretend to eat the five food samples when we ask them to evaluate the usability. The assessment of their also there asked to use at the same time the system where you want the button if we eat as the base system, and counts the number of eating.

We do experiments with food sample. We show the flow of the system and the operation of the user in Figure 7.

1. Sit in the front as reflected in the smartphone (a)
2. The food closer to the mouth using chopsticks (b)
3. Recognition is performed to cut out an image when it is determined to have sufficiently close (c) (d)
4. To display the recognition result, and added to the total calories (e)

In the experiments, we asked the subjects to repeat this action for each of the five food items.

We ask to evaluate it takes easy of record system and visibility. Question is as follows:

Q1 Is this system easy to take record?

Q2 Is this system easy to see record?

Q3 Which systems is easier to use?

Table 3. The mean score and standard deviation.

	Q1	Q2
base system	2.0 ± 0.6	3.0 ± 0.0
proposed system	4.8 ± 0.4	3.4 ± 0.5

We adopted five-step evaluation for Q1 and Q2 and choice from two systems for Q3. If a user think bad is one, think normal is three, think good is 5 to get an evaluation. Q1,Q2's result show the mean score and standard deviation in Table 3.

We get a reputation as better than the base system to everyone in the ease of take the record. That they can take a recording without having to put the chopsticks at meal scene also led to a high rating. It is not possible to give a clear difference between the base system in terms of visibility. We think like that can change the way of display.

For Q3, all the subjectives answered that the proposed system is easier to use. However, we received the opinion poor detection accuracy of chopsticks and view of history of what you eat as improvements, so an exercise for the future.

6.4. Discussions

In this experiment, the system records eating actions during mealtime automatically based on the top ranked food item by food recognition. However, food recognition is not perfect, but only 74.8% classificaton rate. From the practical point of view, mistakenly recognized foods should be correctable by a user. However, if the system asks a user to correct the recognition result by touching screen, it is almost the same as the baseline system. Correction of the recognized results should be possible by the other way than touching screen such as simple hand action recognition or voice recognition.

In the experiment, we use mixture of real food item photos and photos of food samples which are made of plastic as a training image set. This might have negative effect on the recognition performance. It is easy to collect photos of foods served on the table. In fact, some food photo datasets such as Pittsburgh fast-food image dataset [10] and "UEC Food 100" [11] are available on the Net. On the other hand, it is not easy to collect photos showing that food items is being picked by chopsticks or a fork. It is one of our future works to solve this problem.

7. CONCLUSIONS

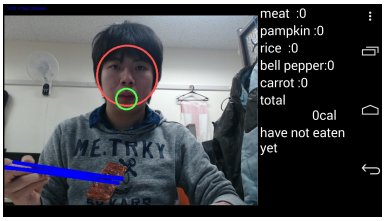
In this paper, we proposed a new-style food recording system which employs real-time eating action recognition and food categorization for meal scene. The system continuously monitors user's eating action and estimates the caloric intake in a real-time way. This enables us to estimate total calorie of the food the amount of which is undecided before eating. This special feature is totally different from existing image-recognition-based food calorie estimation systems which requires taking meal photos before starting to eat.

As an experimental system, we implemented a real-time eating action recording system for "Yakiniku" scene which is a Japanese-style barbecue. The system can classify five categories of representative Yakiniku food items including

Table 2. Accuracy evaluation with feature [E].

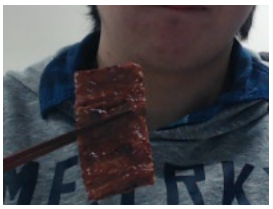
food	meat	rice	pumpkin	bell pepper	carrot	recall(%)
meat	38	1	5	6	0	76.0
rice	0	43	7	0	0	86.0
pumpkin	7	1	28	8	6	56.0
bell pepper	1	0	4	44	1	88.0
carrot	5	3	5	3	34	68.0
precision(%)	74.5	89.6	57.1	72.1	82.9	74.8

(a) The system ready for use



(b) To lift the food using chopsticks

(c) It is sufficiently close to the mouth



(d) The cropped image

(e) To display the recognition result, and added to the total calories

Fig. 7. the flow of the system and the operation of the user

meat, rice, pumpkin, bell pepper and carrot. The system has achieved 74.8% classification rate, and was evaluated as being superior to the baseline system which employed no eating action recognition by user study.

As feature work, first of all, improving of the recognition accuracy is essential, We plan to introduce Fisher Vector [3]. Furthermore, improving of the training data set is also important. We believe that to increase the accuracy it is needed to collect training food item photos taken under various conditions. In addition, we plan to extend our system to other meal scenes than "Yakiniku" as sharing food platter and conveyor-belt-style sushi. To do that, we need to detect fork, knife and hands as well as chopsticks.

8. REFERENCES

[1] K. Aizawa, Y. Maruyama, H. Li, and C. Morikawa, "Food balance estimation by using personal dietary tendencies in a multimedia food log," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2176–2185, 2013. 1, 2

[2] Y. Kawano and K. Yanai, "Rapid mobile object recognition using fisher vector," in *Proc. of Asian Conference on Pattern Recognition*, 2013. 1, 2

[3] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. of European Conference on Computer Vision*, 2010. 1, 2, 6

[4] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," in *Proc. of International Conference on Image Processing (ICIP)*, 2009. 2

[5] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, 2012. 2

[6] "Foodlog," http://www.foodlog.jp/. 2

[7] Y. Kawano and K. Yanai, "Real-time mobile food recognition system," in *Proc. of CVPR International Workshop on Mobile Vision (IWMV)*, 2013. 2

[8] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 480–492, 2012. 2, 4

[9] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004. 2

[10] "Pittsburgh Fast-food Image Dataset," http://pfid.intel-research.net/. 5

[11] "UEC-Food100," http://foodcam.mobi/dataset/. 5