Real-time Mobile Food Recognition System

Yoshiyuki Kawano and Keiji Yanai
The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan
{kawano-y,yanai}@mm.inf.uec.ac.jp

Abstract

We propose a mobile food recognition system the purposes of which are estimating calorie and nutritious of foods and recording a user's eating habits. Since all the processes on image recognition performed on a smartphone, the system does not need to send images to a server and runs on an ordinary smartphone in a real-time way.

To recognize food items, a user draws bounding boxes by touching the screen first, and then the system starts food item recognition within the indicated bounding boxes. To recognize them more accurately, we segment each food item region by GrubCut, extract a color histogram and SURF-based bag-of-features, and finally classify it into one of the fifty food categories with linear SVM and fast χ^2 kernel. In addition, the system estimates the direction of food regions where the higher SVM output score is expected to be obtained, show it as an arrow on the screen in order to ask a user to move a smartphone camera. This recognition process is performed repeatedly about once a second. We implemented this system as an Android smartphone application so as to use multiple CPU cores effectively for real-time recognition.

In the experiments, we have achieved the 81.55% classification rate for the top 5 category candidates when the ground-truth bounding boxes are given. In addition, we obtained positive evaluation by user study compared to the food recording system without object recognition.

1. Introduction

In recent years, food habit recording services for smartphones such as iPhone and Android phones have become popular. They can awake users' food habit problems such as bad food balance and unhealthy food trend, which is useful for disease prevention and diet. However, most of such services require selecting eaten food items from hierarchical menus by hand, which is too time-consuming and troublesome for most of the people to continue using such services for a long period.

Due to recent rapid progress of smartphones such as iPhone and Android phones, they have obtained enough computational power for real-time image recognition. Currently, a quad-core CPU is common as a smartphone's CPU, which is almost equivalent to a PC's CPU released several years ago in terms of performance. Old-style mobile systems with image recognition need to send images to high-performance servers, which must makes communication delay, requires communication costs, and the availability of which depends on network conditions. In addition, in proportion to increase number of users, more computational resources of servers is also required, which makes it difficult to recognize objects in a real-time way.

On the other hand, image recognition smartphone is much more promising in terms of availability, communication cost, delay, and server costs. Then, by taking advantage of rich computational power of recent smartphones as well as recent advanced object recognition techniques, in this paper, we propose a real-time food recognition system which runs on a common smartphone. To do that, we adopt a linear SVM and a fast χ^2 kernel based on kernel feature maps [13] and implement a system as a multi-threaded system for using multiple CPU cores effectively.

Figure 1 shows the screenshot of the proposed system which runs as an Android smartphone application. First a user point a smartphone camera to foods, draws a bounding box (represented in the yellow rectangular in the figure) by dragging on the screen, and then food image recognition is activated for the given bounding box. The top five candidates for the yellow bounding box are shown on the left side of the screen. If a user touches one of the candidate items, the food category name and the photo are recorded as a daily food record in the system. In addition, the proposed system has functions on automatic adjustment of bounding boxes based on the segmentation result by GrubCut [12], and estimation of the direction of the expected food regions based on Efficient Sub-window Search (ESS) [5]. Since this recognition process is performed repeatedly about once a second, a user can search for good position of a smartphone camera to recognize foods accurately by moving it continuously without pushing a camera shutter button.

In the experiments, we have achieved the 81.55% classification rate for the top 5 category candidates when the ground-truth bounding boxes are given. In addition, we obtained positive evaluation by user study compared to the food recording system without object recognition.

To summarize our contribution in this paper, it consists of four folds: (1) implementing an interactive and real-time

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.5

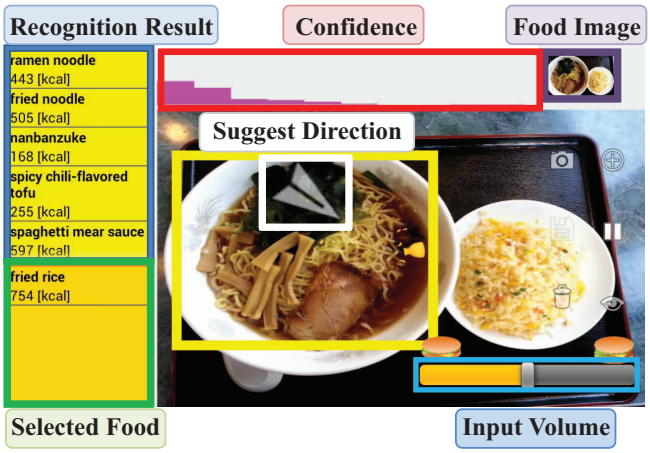


Figure 1. The screenshot of the main screen of the proposed system.

food recognition and recording system running on a consumer smartphone, (2) using a liner SVM with a fast χ^2 kernel for fast and accurate food recognition, (3) adjustment of the given bounding box, and (4) estimation of the direction of the expected food region automatically.

The rest of this paper is organized as follows: Section 2 describes related work. In Section 3, we explain the overview of the proposed system. In Section 4, we explain the detailed method for food recognition. In Section 5 describes the experimental results and user study, and in Section 6 we conclude this paper.

2. Related Work

2.1. Food Recognition

Food recognition is difficult task, since appearances of food items are various even within the same category. This is a kind of fine-grained image categorization problem. As food recognition, Yang *et al.* [14] proposed pairwise local features which exploit positional relation of eight basic food ingredients. Matsuda *et al.* [9] proposed method for recognition multiple-food images, which first detects food put regions by several detectors, next recognizes by extracted color, texture, gradient, and SIFT using multiple kernel learning(MKL).

As a food recording system with food recognition, Web application FoodLog [3] estimates food balance. It divides the food image into 300 blocks, from each blocks extracts color and DCT coefficients, next classifies to five groups such as staple, main dish, side dish, fruit, and non-food.

The TADA dietary assessment system [7] has food identification and quantity estimation, although it has some restriction that food must be put on white dishes and food photos must be taken with a checkerboard to food quantity estimation.

In all the above-mentioned systems image recognition processes perform on servers, which prevents systems from being interactive due to communication delays. On the other hand, our system can recognize food items on a client side in a real-time way, which requires no communication to outside computational resources and enables user to use it interactively. Note that our current system recognize only 50 food categories, and it requires user's assistances to estimate food volumes by touching a slider on the system screen.

2.2. Mobile Device and Computer Vision

With the spread of smartphones, some mobile applications exploiting computer vision technique have been proposed. Google Goggles [1] is one of the most well-known application which recognizes specific object, returns to user about object information. Kumar *et al.* [4] proposed recognize 184 species application Leaf snap. For Leaf image on the solid light-colored background segment and extract curvature-based shape features. Maruyama *et al.* [8] proposed a system which extracts color feature and recognizes 30 kinds of food ingredients on a mobile device. As a mobile device application emphasizing real-time image recognition, Lee *et al.* [6] proposed a system which trains templates, then decomposes descriptors such as intensity and gradient orientation. At the time of testing, template matching at multiple scales enables the system to detect and track objects in a real-time way. In the above-mentioned mobile vision systems, local-feature-based specific object recognition was performed, while we tackle category-level object recognition on foods on a mobile device.

As an interactive mobile vision system, Yu *et al.* [15] proposed Active Query Sensing (AQS) the objective of which is localization of the current position by matching of street photos. When the system fails in location search, it suggests the best viewing direction for the next reference photo to a user based on the pre-computed saliency on each location. Our system is also built as an interactive system which detects object regions based on the bounding boxes a user draws and suggests the direction of food regions where the higher evaluation output score is expected.

3. System Overview

The final objective of the proposed system is to support users to record daily foods and check their food eating habits. To do that easily, we built-in food image recognition technique on the proposed system. In this paper, we mainly describe a food image recognition part of the proposed system.

Processing flow of typical usage of the proposed system is as follows (See Figure 2 as well):

1. A user points a smartphone camera toward food items before eating them. The system is continuously acquiring frame images from the camera device in the background.

2. A user draws bounding boxes over food items on the screen. The bounding boxes will be automatically ad-

[1] http://www.google.com/mobile/goggles/

Figure 2. System process flow

justed to the food regions. More than two bounding boxes can be drawn at the same time. .

3. Food recognition is carried out for each of the regions within the bounding boxes. At the same time, the direction of the region having the higher evaluation score is estimated for each bounding box.

4. As results of food recognition and direction estimation, the top five food item candidates and the direction arrows are shown on the screen.

5. A user selects food items from the food candidate list by touching on the screen, if found. Before selecting food items, a user can indicate relative rough volume of selected food item by the slider on the right bottom on the screen. If not, user moves a smartphone slightly and go back to 3.

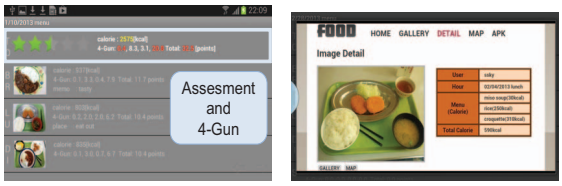
6. The calorie and nutrition of each of the recognized food items are shown on the screen.

In addition, a user can see his/her own meal record and its detail including calories and proteins of each food items on the screen as shown in Figure 3(a). Meal records can be sent to the server, and a user can see them on the Web (Figure 3(b)).

4. A Method of Food Recognition

Before starting to recognize food items for the frame images taken by a smartphone camera, first the system requires that a user draws bounding boxes which bounds food items on the screen by dragging along the diagonal lines of the boxes.

In this section, we explain the following three processing: (1) Adjustment of bounding boxes, (2) Recognition of food items with the given bounding boxes, and (3) Estimation of the direction of the possible food region.



(a) Meal records on the screen.

(b) Meal records on the Web.

Figure 3. Meal records.

4.1. Bounding Box Adjustment

First, a user draws bounding boxes roughly on the screen by dragging. The bounding boxes a user draws are not always accurate and sometimes they are too large for actual food regions. Therefore, we apply well-known graph-cut-based segmentation algorithm GrabCut [12], and then modify the bounding boxes so as to fit them to food regions segmented by GrabCut. GrabCut needs initial foreground and background regions as seeds. Here, we provide GrubCut the regions within bounding boxes as foreground and the areas out of the doubly-extended boxes of the original bounding boxes as background. Since the computational cost of GrabCut is relatively high in the real-time recognition system, the bounding box adjustment is performed only once after it was drawn.

4.2. Food Recognition

Food recognition is performed for each of the window images within the given bounding boxes. Firstly, image features are extracted from each window, secondly feature vectors are built based on the pre-computed codebook, and finally we evaluate the feature vectors with the trained linear SVMs on 50 food categories. The top five candidates which have the top five SVM scores over 50 categories and all the bounding boxes are shown on the screen as food item candidates.

4.3. Image Features

Since we aims for implementing a mobile recognition system which can run in the real-time way, image features to be extracted should be minimum requisites. Then we evaluated and compared performance and computational costs of various kinds of common image features including global and local features such as color histogram, color moment, color auto correlogram, Gabor texture feature, HoG, PHoG, and Bag-of-SURF. Finally we chose the combination of color histogram and Bag-of-SURF. To save computational cost, if the longer side of a bounding box is more than 200, the image extracted from the bounding box is resized so that the longer side becomes 200 preserving its aspect ratio.

Color Histogram: We divide an window image into 3×3 blocks, and extract a 64-bin RGB, color histogram from each block. Totally, we extract a 576-dim color histogram. Note that we examined HSV and La*b* color histograms as well, and RGB color histogram achieved the best among them.

Bag-of-SURF: As local features, we use dense-sampled bag-of-SURF. SURF [1] is an invariant 64-dim local descriptor for scale, orientation and illumination change. We sample points by dense sampling in scale 12 and 16 with every 8 pixel with the window. To convert bag-of-features vectors, we prepared a codebook where the number of codeword was 500 by k-means clustering in advance. We apply soft assignment [11] by 3 nearest-neighbor assigned reciprocal number of Euclid distance to the codeword, also we use fast approximated-nearest-neighbor search based kd-tree to search the codebook for the corresponding codeword for each sampled point. Finally, we create a 500-dim bag-of-SURF vector.

4.4. Classification

As a classifier, we use a linear kernel SVM, and we adopt the one-vs-rest strategy for multi-class classification. In the experiment, since we prepared 50 food categories, we trained 50 linear SVM classifiers.

Linear kernel is defined as the inner product of two vectors. In advance, we computed the inner product of a support vector and the weight of the corresponding support vector, then Linear SVM can be written as follows:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{M} y_i \alpha_i K(\mathbf{x}, \mathbf{x_i}) + b = \sum_{i=1}^{M} y_i \alpha_i \langle \mathbf{x}, \mathbf{x_i} \rangle + b \\ &= \langle \sum_{i=1}^{M} y_i \alpha_i \mathbf{x_i}, \mathbf{x} \rangle + b = \langle \mathbf{w}, \mathbf{x} \rangle + b \end{aligned} \quad (1)$$

where $\mathbf{x}$ is an input vector, $f(\mathbf{x})$ is an output SVM score, $\mathbf{x_i}$ is support vector, $y_i \in \{+1, -1\}$ is a class label, α_i is a weight of the corresponding support vector, b is a bias vector, and M is the number of support vector. By this transformation, we can save memory to store support vectors as well as calculation of kernels. Therefore, when N is the dimension of feature vector, calculation of a SVM score requires $\mathcal{O}(N)$ operations and $\mathcal{O}(N)$ memory space. We train SVMs with LIBLINEAR [2] in off-line.

Although a linear SVM is fast and can save memory, classification accuracy is not as good as a non-linear kernel SVM such as a SVM with χ^2-RBF kernel. To compensate weakness of a liner SVM, we use explicit embedding technique. In this work, we adopt kernel feature maps. Vedaldi *et al.* [13] proposed homogeneous kernel maps for χ^2, intersection, Jansen-Shanon's and Hellinger's kernels, which are represented in the closed-form expression. We choose mapping for χ^2 kernel and we set a parameter so that the dimension of mapped feature vectors are 3 times as many as the dimension of original feature vectors. This mapping can be applied for L1-normalized histogram [13]. Then we apply this to L1-normalized color histograms and Bag-of-SURF vectors. In the experiments, we compared this χ^2 kernel mapping with the square-rooting feature vector which corresponds to an exact mapping for Hellinger's kernel the effectiveness of which was demonstrated by Perronnin *et al.* [10].

In the experiment, we use both a RGB color histogram and bag-of-SURF and integrity them by a linear weighting with the weights estimated by cross-validation.

4.5. Estimation of the more reliable direction

In case that no categories with high SVM scores are obtained, camera position and viewing direction should be changed slightly to obtain more reliable SVM evaluation scores. To encourage a user to move a smartphone camera, the proposed system has a function to estimate the direction to get the more reliable SVM score and show the direction as an arrow on the screen as shown in Figure 1.

To estimate the direction of the food regions with more reliable SVM score, we adopt an effective window search method for object detection, Efficient Sub-window Search (ESS) [5], which can be directly applied for a combination of a liner SVM and bag-of-features.

The weighting factor $\mathbf{w}$ of an input vector of the SVM classifier represented in Equation (1) can be decomposed into a vector $\mathbf{w}^+$ including positive elements and a vector $\mathbf{w}^-$ including negative elements.

$$\mathbf{w} = \mathbf{w}^+ + \mathbf{w}^- \quad (2)$$

Therefore, an SVM output score for one rectangular region can be calculated in $\mathcal{O}(1)$ operation by making use of $\mathbf{w}^+$ and $\mathbf{w}^-$ integral images according to the ESS method [5]. In case of the above-mentioned soft assignments to codewords, the output score can be calculated by a product of w and assigned values to codewords. We search for the window which achieves the maximum SVM score by Efficient Sliding Windows search so as to keep more than 50% area being overlapped with the original window. Finally the relative direction to the window with the maximum score from the current window are shown as an arrow on the screen (See Figure 1).

5. Experiments

In this section, we describe experimental results regarding recognition accuracy and time. In addition, we also explain the evaluation result by user study.

In the experiments, we prepared a fifty-category food dataset which has more than 100 training images per category all the food item in which are marked with bounding boxes. The total number of food images in the dataset is 6,781. Figure 4 shows all the category names and their sample photos.

5.1. Evaluation on classification accuracy

In this subsection, we evaluate the effectiveness of feature map-based fast χ^2 kernel, bounding box adjustment,

Figure 4. 50 kinds of food images which are recognition targets in the this paper.

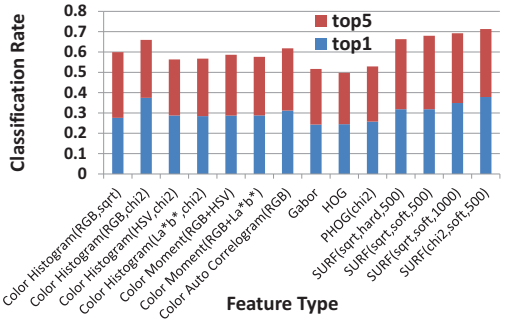


Figure 5. Classification rates with single features within top 1 and top 5.

and estimation of the expected food regions for mobile food recognition.

We evaluate food classification accuracy with the 50-category food dataset by 5-fold cross-validation, which corresponds to food recognition under the condition that ground-truth bounding boxes are given.

Firstly, we evaluated and compared performance in food recognition with various kinds of common image features including global and local features such as color histogram, color moment, color auto correlogram, Gabor texture feature, HoG, PHoG, and Bag-of-SURF. Figure 5 shows the classification rate using single features. From this table, RGB color histogram and Bag-of-SURF with fast χ^2 are regarded as being effective. RGB color histogram with fast χ^2 achieved the classification rate 65.90% within the top five, while bag-of-SURF with fast χ^2 achieved 71.34% within the top five, which outperformed the result with bag-of-SURF with Hellinger's kennel with the twice number of codewords. For all the cases, bag-of-SURF by soft assignment are better than by hard assignment. We also tried recent proposed local binary descriptor based BoF, but very poor accuracy, consider cause by bigger quantization error for little information. Finally we chose the combination of color histogram and Bag-of-SURF as image features for the proposed system.

Figure 6 shows the classification rate with RGB color histogram, Bag-of-SURF and their combination with a fast χ^2 liner SVM. Regarding the combined feature, the rates with a standard linear SVM and a non-linear RBF-χ^2 SVM are also shown in the Table. In case of two feature combination with a fast χ^2 liner SVM and a standard linear SVM, the classification rates were 53.5% and 39.1%, and the classification rates within the top five candidate was 81.6% and 71.0%, respectively. Moreover, in case of a standard non-linear RBF-χ^2 kernel SVM which is much more time-consuming and is not appropriate for real-time recognition, the rate was 57.3%, and the rate within the top five was 83.2%. From these results, the results by a fast χ^2 linear SVM outperformed the results by a standard linear SVM by more than 10 points, and a fast χ^2 SVM is almost comparable to a non-linear SVM, which means that a fast χ^2 linear SVM is very effective for both speed and accuracy which are requirements for a mobile image recognition system. Since the proposed system shows the top five candidate on the screen, 81.6% can be regarded as the rate that a true food name are shown in the candidate list.

Next, we made an experiments to examine effectiveness of bounding box adjustment. We magnified ground-truth bounding boxes of test images with 25% in terms of bounding box size. In fact, we used only 1912 food images as test images in the 5-fold cross-validation classification experiment, since for some food photos their size are almost the same as the size of attached bounding boxes and they do not includes 25% backgound regions. We compared groundtruth bounding boxes, 25%-magnified bounding boxes, and adjusted bounding boxes after 25% magnification in terms of classification rate under the same condition as the previous experiment. Figure7 shows the results, which indicates that 25% maginifation of groundtruth bounding boxes degraded the classification rate within the top five by 10.1%, while 25% maginification with bounding box adjustment degraded the rate by only 4.4%. From this results, GrubCut-based bounding box adjustments can be regarded as being effective.

Finally, we made an experiment on estimation of the direction of a food window. We evaluated error in the direction estimation in case of sifting the ground-truth bounding boxes by 10, 15, 20, and 25% to each of eight directions around the original boxes. Figure 8 show cumulative classification rates of the estimated direction in case of using soft and hard assignment, which proved that soft assignment was better than hard assignment. Figure9 shows cumulative classification rates of the estimated direction with different shifts. The rates with less than $\pm 20^\circ$ error and $\pm 40^\circ$ error were 31.81% and 50.34% in case of 15% shift, and are 34.54%, and 54.16% in case of 25% shift. From these results, when the difference between the ground-truth and the given bounding box is small, estimation of the direction of the ground-truth bounding box is more difficult. This is because the difference of SVM scores between them is small in case that the difference in the location of the bounding boxes is small.

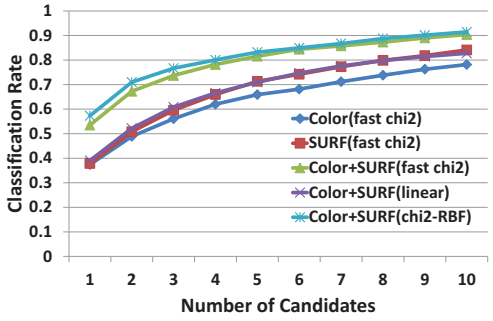


Figure 6. Classification rates of RGB color histogram, bag-of-SURF and their combination with a fast χ^2 linear SVM, and rates of the combination with a standard linear SVM and a non-linear RBF-χ^2 SVM within the top n candidates. Note that the graphs of single SURF with fast χ^2 and combination with a standard linear SVM are overlapped and difficult to distinguish.

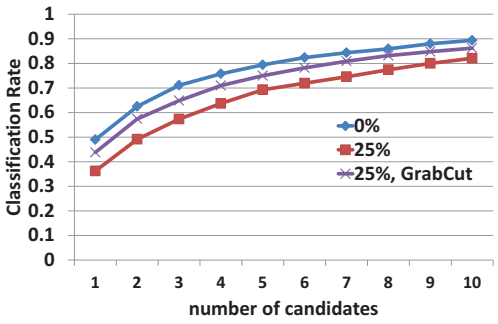


Figure 7. Classification rates in case of the ground-truth bounding box (BB), 25% magnified BB and adjusted BB after 25% magnification (shown as "25%, GrabCut").

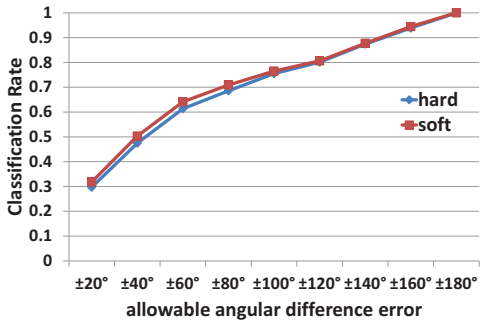


Figure 8. Cumulative classification rate of the estimated orientation with soft assignment and hard assignment. The horizontal axis expresses allowable angular difference error and the vertical axis expresses classification rate in terms of estimated directions.

5.2. Evaluation of processing time

We implemented the proposed system as Android application assuming that quad core is available. High-cost

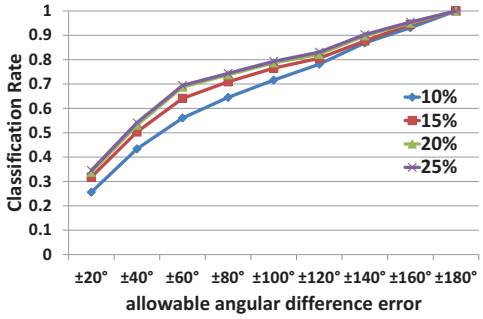


Figure 9. Cumulative rate of the estimated orientation in case of 10, 15, 20, and 25% shifted bounding boxes.

Table 1. Average processing time.

	average time [sec]
Bounding Box Adjustment	0.70
Recognition	0.26
Search Food Region	0.091
Recognition + Search Food Region	0.34

SURF descriptor extraction, assignment to codewords, evaluation of 50 kinds of fast χ^2 linear SVMs and direction estimation are carried out over four cores in parallel, while low-cost color histogram extraction is performed on a single core.

We measured processing times on the latest smartphone, Samsung Galaxy Note II (1.6GHz Quad Core with Android 4.1). The results are shown in Table 1. Processing time for recognition and direction estimation are 0.26 seconds, 0.09 seconds, while bounding box adjustment is relatively high-cost, which takes 0.70 seconds. This is why bounding box adjustment are carried out once after drawn. Among 0.26 seconds for recognition, color histogram extraction and linear SVM classification takes only 0.003 seconds, and most of the time are taken for extraction of SURF descriptors and searching of the codebook to create bag-of-features vectors. Because the total time of recognition and estimation of the direction is 0.34 seconds, three bounding boxes can be processed in around one second at the same time.

5.3. User Study

We asked five student subjects to evaluate quality of the proposed system in five step evaluation regarding food recognition, how easy to use, quality of direction estimation, and comparison of the proposed system with the baseline which has no food recognition and requires selecting food names from hierarchical menus by touching. The evaluation score 5, 3 and 1 means good, so-so, and bad, respectively. At the same time, we measured time for selecting food items with food recognition, and compared it with the time for selecting food items from the hierarchical menu by hand.

Figure 10 shows the spent time for selecting each food

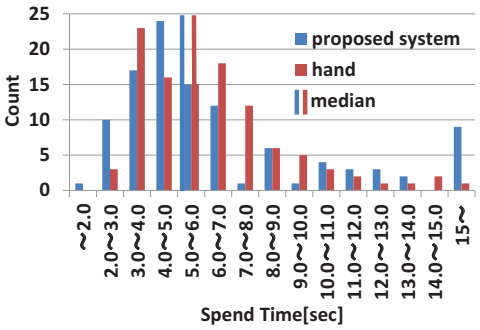


Figure 10. Time comparison: food recognition vs. hierarchical menu.

Table 2. User study results which are the average of five-step evaluation scores.

Outcome Measure	average score
Recognition quality	3.4
Facility to use	4.2
Quality of Direction Suggestion	2.4
Proposed System Quality (compared with hand-selection)	3.8

item. The median time were 5.1 second with food recognition, and 5.7 second by hand. This means the proposed system can help a user select food names faster than from a hierarchical menu by hand. However, for some food items which not able to be recognized, it spent long time to find food names using food recognition.

Table 2 shows the system evaluation by the five grade evaluation. Except for suggest direction, more than three points are obtained. Especially, usability of the system is good, since recognition is carried out in a real-time way. On the other hand, estimation of the expected food region is not evaluated as being effective, since the classification accuracy is not so good for practical use. We will improve it as a future work.

6. Conclusions and Future Work

In this paper, we propose a real-time food recognition system on a smartphone. The system adopts a liner SVM with a fast χ^2 kernel, bounding box adjustment and estimation of the expected direction of a food region. In the experiment, we have achieved 81.55% classification rate with the top five candidates when ground-truth bounding boxes are given. In addition, we obtained positive evaluation by user study compared to the food recording system without object recognition.

As feature works, we plan to extend the system regarding the following issues:

- Touch just a point instead of drawing bounding boxes to specify food regions.
- Use multiple images to improve accuracy of food item recognition.
- Improve accuracy of estimation of expected food regions or move the bounding boxes automatically instead of only showing the direction.
- Take into account additional information such as user's food history, GPS location data and time information.
- Increase the number of food categories to make the system more practical.

Note that Android application of the proposed mobile food recognition system can be downloaded from http://foodcam.mobi/ .

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 4

[2] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. 4

[3] K. Kitamura, T. Yamasaki, and K. Aizawa. Food log by analyzing food images. In *Proc. of ACM International Conference Multimedia*, pages 999–1000, 2008. 2

[4] N. Kumar, P. Belhumeur, A. Biswas, D. Jacobs, W. Kress, I. Lopez, and J. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Proc. of European Conference on Computer Vision*, 2012. 2

[5] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008. 1, 4

[6] T. Lee and S. Soatto. Learning and matching multiscale template descriptors for real-time detection, localization and tracking. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2011. 2

[7] A. Mariappan, M. Bosch, F. Zhu, C. Boushey, D. Kerr, D. Ebert, and E. Delp. Personal dietary assessment using mobile devices. In *Proc. of the IS&T/SPIE Conference on Computational Imaging VII*, volume 7246, pages 72460Z–1–72460Z–12, 2009. 2

[8] T. Maruyama, Y. Kawano, and K. Yanai. Real-time mobile recipe recommendation system using food ingredient recognition. In *Proc. of ACM Multimedia Workshop on Interactive Multimedia on Mobile and Portable Devices*, pages 27–34, 2012. 2

[9] 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*, pages 1554–1564, 2012. 2

[10] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 2297–2304, 2010. 4

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2008. 4

[12] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, pages 309–314, 2004. 1, 3

[13] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–492, 2012. 1, 4

[14] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2010. 2

[15] F. Yu, R. Ji, and S. Chang. Active query sensing for mobile location search. In *Proc. of ACM International Conference Multimedia*, pages 3–12, 2011. 2