# FoodCam: A Real-Time Mobile Food Recognition System Employing Fisher Vector

Yoshiyuki Kawano and Keiji Yanai

Department of Informatics, The University of Electro-Communications
Chofu, Tokyo 182-8585, Japan
{kawano-y,yanai}@mm.cs.uec.ac.jp

**Abstract.** In the demo, we demonstrate a mobile food recognition system with Fisher Vector and liner one-vs-rest SVMs which enable us to record our food habits easily. In the experiments with 100 kinds of food categories, we have achieved the 79.2% classification rate for the top 5 category candidates when the ground-truth bounding boxes are given. The prototype system is open to the public as an Android-based smartphone application.

## 1 Introduction

In recent years, food habit recording services for smartphones 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.

Most of the existing mobile image recognition systems such as Google Goggles 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. On the other hand, image recognition on the client side, that is, on a smartphone is much more promising in terms of availability, communication cost, delay, and server costs. Due to recent rapid progress of smartphones such as iPhone and Android phones, they have obtained enough computational power for real-time image recognition. Then, by taking advantage of rich computational power of recent smartphones as well as recent advanced object recognition techniques, in this demo, we propose a real-time food recognition system which runs on a common smartphone. To boost accuracy and speed of food image recognition, we adopt Fisher Vector and a linear SVM, and implement a system as a multi-threaded system for using quad CPU cores effectively.

Since the recognition process on the proposed system 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. This is a big advantage of a real-time image recognition system on a mobile device.

C. Gurrin et al. (Eds.): MMM 2014, Part II, LNCS 8326, pp. 369–373, 2014.

In the experiments, we have achieved the 79.2% classification rate for the top 5 category candidates which outperformed the classification rate of our previous server-side food image recognition system [1].

To summarize novelties of the proposed system, it consists of three folds: (1) An interactive and real-time food recognition and recording system running on a consumer smartphone, (2) using Fisher Vector on a mobile device (3) automatic adjustment of the given bounding box.

## 2 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 embedded food image recognition engine on the proposed system.

Processing flow of typical usage of the proposed system is as follows (See Figure 1 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 adjusted to the food regions by a GrabCut-based method [2]. 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 using HoG patches and color patches with a linear SVM and Fisher Vector.
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 for calorie estimation. If not, user moves a smartphone slightly and go back to 3.
6. The name, calorie and nutrition of the recognized food items are shown on the screen and are recorded in the system.

In addition, a user can see his/her own meal record and its detail including calories and nutrition of each food items not only on the screen but also on the Web by sending the records to the server regularly.

## 3 Implementation and Evaluation

In this section, we describe the implementation of the prototype system we will demonstrate at the conference.

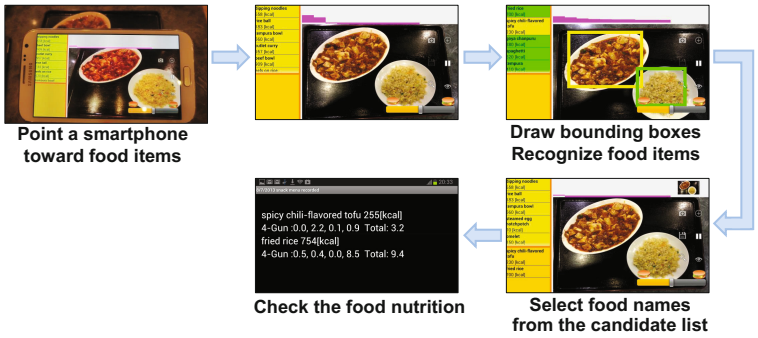


**Fig. 1.** System process flow

**Image Feature.** In this paper, we use the following local descriptors for Fisher Vector encoding: HOG Patch and Color Patch.

Histogram of Oriented Gradients(HOG) was proposed by N. Dalal *et al*[3]. Since HOG description is very simple, it is able to describe much faster than popular local descriptor such as SIFT and SURF. This is important characteristic to carry out real-time recognition on a smartphone. In addition, it is able to extract local feature more densely. As a result, it improves recognition accuracy.

We extract HOG features as local features. We divide a local patch into $2 \times 2$ blocks, and extract gradient histogram regarding eight orientations from each block. Totally, we extract 32-dim HOG Patch features. Then the 32-dim HOG Patch is L2 normalized to an L2 unit length, not adopt HOG-specific normalization by sliding fusion. PCA is applied to reduce dimensions from 32 to 24.

We use mean and variance of RGB value of pixels as Color Patch feature. We divide a local patch into $2 \times 2$ blocks, and extract mean and variance of RGB value of each pixel within each block. Totally, we extract 24-dim Color Patch features. PCA is applied without dimension reduction. The dimension of a Color Patch feature are kept to 24-dim.

**Fisher Vector.** Fisher Vector [4] can decrease quantization error than BoF[5] by using of a high order statistic and GMM-based soft assignments. Moreover, while BoF needs larger dictionary to improve recognition accuracy, larger dictionary brings increase of computational cost for searching nearest visual words. On the other hand, Fisher Vector is able to achieve high recognition accuracy with even small dictionary, and low computational complexity. This is an advantage for mobile devices. Thus adopting Fisher Vector as encoding method is better in terms of recognition accuracy and processing time for a mobile object recognition. But a system on a smartphone does not exist so far that carries out rapid and high precision image recognition with Fisher Vector. Then we propose a recognition method for a smartphone which is rapid and accurate by making good use of computational resource of the smartphone with Fisher Vector.

In this paper, the number of component of Gaussian is 32 and local descriptors reduced to 24 dimensions by PCA. Thus each feature vector is 1536-dimensional.

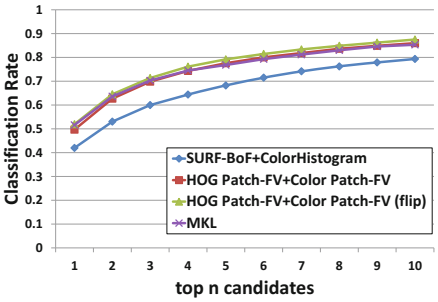


**Fig. 2.** Classification rate by proposed method, client side[6] and server side[1]

**Table 1.** Classification Rate the top1 and top5 candidates by proposed method, client side[6] and server side[1]

| method | top1 | top5 |
|---|---|---|
| SURF-BoF+ColorHistogram[6] | 42.0 | 68.2 |
| HOG Patch-FV+Color Patch-FV | 49.7 | 77.6 |
| HOG Patch-FV+Color Patch-FV(flip) | 51.9 | 79.2 |
| MKL[1] | 51.6 | 76.8 |

To improve recognition accuracy, we apply power normalization ($\alpha = 0.5$) and L2 normalization[4].

**Classification.** As a classifier, we use a linear kernel SVM the computational cost of which is $\mathcal{O}(N)$, and we adopt the one-vs-rest strategy for multi-class classification.

We trained SVMs in off-line. And all the parameter values using recognition steps are loaded on main memory (eigenvalue and eigenvector for PCA, created lookup table, mean of GMM, weight vectors of SVMs). Although all the values can be stored on main memory in advance, Fisher Vector is able to bring better recognition result with even smaller dictionary than that of BoF. We also set the dimension of feature vectors smaller reduced by PCA. As a result, memory space required for Fisher Vector is smaller than the space for codebook for conventional BoF, and in respect of memory Fisher Vector is also superior to BoF.

**Experiments.** To implement the system, we prepared one-hundred category food dataset which has more than 100 training images per category. All the food items in the dataset photos are marked with bounding boxes. The total number of food images in the dataset is 12,905

In this experiments, we compare with 2 types of the existing food recognition system. One is a client-side recognition system by Kawano *et al*[6], and the other is a server-side recognition system by Matsuda *et al*[1].

We compared recognition accuracy with the existing food recognition systems. Our proposed method (HOG Patch-FV+Color Patch-FV) and (HOG Patch-FV+Color Patch-FV(flip)), client side recognition method[6](SURF-BoF+color histogram) and server side recognition method[1]. Note that "flip" means adding horizontally flipped images of training images to training data. [1] adopt total 5 features including hard assignment BoF and global feature based feature and classify with nonlinear $\chi^2 RBF$ kernel MKL-SVM, which is very high computational cost. Figure 2 shows classification rate of each recognition method and Table 1 shows classification rate with top1 and top5. Our HOG Patch-FV+Color Patch-FV achieved 49.7% and 77.6% classification rate with top1 and top5,

**Table 2.** Average Process Time

| | time[sec] | number of food categories |
|---|---|---|
| SURF-BoF+ColorHistogram[6] | 0.26 | 50 |
| HOG Patch-FV+Color Patch-FV | 0.065 | 100 |

respectively. In case of adding flipped images, it achieved 51.9% and 79.2% classification rate with top1 and top5. Our approach is better than Matsuda *et al.*'s food recognition system [1] which was very high cost recognition system running on the server side. system. Therefore we show efficacy our recognition method and it suggests that it is able to carry out high precision image recognition on smartphone.

Next, we measured recognition time by repeating 20 times and averaging them. Table 2 shows the average recognition time. We used Samsung Galaxy NoteII (1.6GHz 4 cores, 4 threads, Android 4.1). We also show the recognition time of [6] for 50 kinds of image recognition. Our approach takes 0.065 seconds for 100 kinds. On the other hand, [6] takes 0.26 seconds for 50 kinds. Thus proposed method is much faster than [6], and this result indicates that the proposed method is more suitable for real-time object recognition.

According to the experiments of recognition accuracy and processing time, we showed efficacy our proposed method. And we achieved better result than server side very high cost recognition method[1]. Therefore we showed rapid image recognition and high precision on smartphone is possible.

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

## References

1. Matsuda, Y., Hoashi, H., Yanai, K.: Recognition of multiple-food images by detecting candidate regions. In: Proc. of IEEE International Conference on Multimedia and Expo (2012)
2. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: SIGGRAPH, pp. 309–314 (2004)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893. IEEE (2005)
4. Perronnin, F., Sánchez, J., Liu, Y.: Large-scale image categorization with explicit data embedding. In: CVPR, pp. 2297–2304 (2010)
5. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Proc.of ECCV Workshop on Statistical Learning in Computer Vision, pp. 59–74 (2004)
6. Kawano, Y., Yanai, K.: Real-time mobile food recognition system. In: Proc. of IEEE CVPR International Workshop on Mobile Vision, IWMV (2013)