Real-time Food Intake Classification and Energy Expenditure Estimation on a Mobile Device

Daniele Ravì, Benny Lo, Guang-Zhong Yang
The Hamlyn Centre
Imperial College London
London, United Kingdom
Email: {d.ravi, benny.lo, g.z.yang}@imperial.ac.uk

***Abstract*—Assessment of food intake has a wide range of applications in public health and life-style related chronic disease management. In this paper, we propose a real-time food recognition platform combined with daily activity and energy expenditure estimation. In the proposed method, food recognition is based on hierarchical classification using multiple visual cues, supported by efficient software implementation suitable for real-time mobile device execution. A Fischer Vector representation together with a set of linear classifiers are used to categorize food intake. Daily energy expenditure estimation is achieved by using the built-in inertial motion sensors of the mobile device. The performance of the vision-based food recognition algorithm is compared to the current state-of-the-art, showing improved accuracy and high computational efficiency suitable for real-time feedback. Detailed user studies have also been performed to demonstrate the practical value of the software environment.**

I. Introduction

Obesity is a growing global health problem that has received increasing attention in recent years. It has been estimated that over 700 million people in the world are classified as obese. In the UK, the obese population has more than trebled in the last 25 years. Obesity is linked to many chronic diseases including diabetes, heart diseases and cancer. Public health systems are responding to this epidemic by promoting good dietary intake and weight management. Traditional methods for dietary assessments mostly rely on questionnaires or self-reporting [1]. These methods are riddled with problems due to underreporting and miscalculation of food consumption [2]. New approaches are required for objective assessment of free-living food intake linking with daily activity patterns [3]. Increasing research in this direction is performed in recent years [4]. For example, signal processing algorithms have been developed to detect and characterize food intake by capturing sound generated during chewing and swallowing of food using an in-ear microphone [5], [6]. Wearable sensors have been used for objective monitoring of ingestive behaviour [7]. Although sensor based approaches are useful for detecting eating habits, they are not suitable for detailed food classification. With increasing sophistication of smart phones, recent approaches have used vision-based methods for image classification [8]. Based on this idea, vision methods can be designed also to solve the food recognition problem directly on the smart phone. This also has the advantage of using the built-in inertial sensors to monitor the activities of daily living, thus providing detailed information in terms of energy expenditure. Although many solutions exist for physical activity monitoring (see e.g. [9] for a survey), limited work has attempted to monitor the individual's food intake at the same time. In the rest of the paper, we will mainly focus on the technical development of a vision-based food recognition system and its detailed performance evaluation. We will also demonstrate the deployment of the integrated system and illustrate the potential value of the platform for population-based assessment.

Fig. 1. Screenshot of the proposed mobile application for the food recognition problem. The bar on the left shows the first 5 classes recognized by the system for the current frame.

II. Related work

Image-based classification is a popular topic in computer vision. Local features such as the SIFT [10] or global features such as GIST [11] are frequently used for this task. However, for images of food, key point features or landmarks are either not available or representative enough for reliable classification [12]. For this reason, appearance and texture are often used as a bag of features without explicit reference to spatial distribution. Joutou et al. [13], for example, combined Bag-of-SIFT with color histograms and Gabor filters to discriminate between images of a dataset composed by 50 different food categories. Matsuda et al. [14], [15] employed a Bag-of-SIFT on spatial pyramid, histograms of gradient, color histogram and Gabor filters to train a Multiple Kernel Learning based on deformable part models. Zong et al. [16] employed a SIFT detector and Local Binary Patterns (LBP) for encoding local shape context. Concatenated features are used in many of these methods by assuming the food classes are equally abundant and complex in term of classification. This is problematic when there are biased distributions. To overcome this problem, we use a hierarchal representation that extracts a feature only if it

is relevant for the considered image. To ensure computational efficiency and generalizability, local features are excluded. As shown by Farinella et al. [17] and by Zong et al. [16], textures are important for food classification and for this reason, the Local Binary Pattern descriptor is used in our system. In the proposed method, a hierarchical classification approach is implemented. Key emphasis is placed on the real-time performance of the system without sacrificing classification accuracy.

III. Proposed approach

The key processing flow of the proposed system is computed as follows:

1) A user points a smart phone camera toward food items before eating them.
2) Using the touchscreen display, the user refines the target region, as illustrated in Fig. 1.
3) The system automatically processes the images and in real-time lists the five most likely food categories recognized.

The general workflow of the algorithm developed for food classification is listed in $Algorithm1$. In essence, it takes as input the frame I and the trained parameters described in Sec. III-B and III-C. Starting from the first level of the hierarchy, the algorithm proceeds until one of the classifiers associated with a food class reaches a high value of confidence. At each level i, a new set of local descriptors V_i, obtained exploiting the feature F_i, are extracted from Frame I and the corresponding Fisher Vector FV_i is computed. The Fisher Vector outputs are concatenated according to the features combination required at the current level. Specifically, Idx_i contains the indices used to refer the features combination Fc_i involved at level i. Finally for each class that is recognized from the current combination (these classes are obtained by setting all the elements in Idx_i as an index for the vector $Cl_1, Cl_2, ..., Cl_M$), a linear classifier is deployed. If the classification confidence of one of the classifiers reaches above a predefined threshold T, the class label associated with this classifier is returned. In the following sub-sections, we will explain in more details about the processing steps involved. Specifically, in section III-A we describe the features employed in the system, in section III-B the training process for generating the required classification hierarchy, in section III-C the details of the Fisher Vector representation and finally in section III-D the classification steps .

Algorithm 1 Proposed food intake classification

Input:

I	▷ Image
L	▷ Number of levels in the hierarchy
M	▷ Number of features combinations
P	▷ Number of classes
$F_1, F_2, ..., F_N$	▷ Pool of features
$Fc_1, Fc_2, ..., Fc_M$	▷ Set of features combinations
$Cl_1, Cl_2, ..., Cl_M$	▷ Set of classes recognized by each Fc_i
$Idx_1, Idx_2, ...Idx_L$	▷ Set of indices
$Lc_1, Lc_2, ..., Lc_P$	▷ Classification models
$\Sigma_1, \Sigma_2, ..., \Sigma_L$	▷ Covariance matrix of the GMM
$\mu_1, \mu_2, ..., \mu_L$	▷ Mean vector of the GMM
$w_1, w_2, ..., w_L$	▷ Weight vector of the GMM
$Lab_1, Lab_2, ...Lab_P$	▷ Class labels
T	▷ Threshold

Output:

$result$	▷ The label classification

```
result ← −1;
for i=1 to L do
    V_i ← extract_local_descriptors(F_i, I);
    FV_i ← extract_fisher_vector(w_i, μ_i, Σ_i, V_i);
    for Each index j in Idx_i do
        c ← combine_features(Fc_j, FV_1...FV_i);
        for Each class z in Cl_j do
            s ← compute_linear_classifier(Lc_z, c);
            if s > T then
                result ← Lab_z;
                Exit();
            end if
        end for
    end for
end for
```

A. Pool of available features

The Histogram of Oriented Gradients (HoG) proposed by N. Dalal et al. [18] is used in this paper because of its simplicity for real-time extraction on a smart phone. The method counts the occurrences of gradient orientation in localised portions of an image. The HOG is similar to the edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in the fact that it is computed on a dense grid and uses overlapping local contrast normalization for improved accuracy. As suggested in [19], we extract HOG features in the following manner:

(a) The image is divided into overlapped windows of 16 × 16 pixels, densely sampled.
(b) Each window is subdivided in 2 × 2 blocks, and the gradient histogram regarding eight orientations from each block is extracted. The final dimension of each Hog features is 32 (4 blocks × 8 directions).
(c) PCA is applied to reduce the dimensions from 32 to 24.

For capturing texture features, the LBP descriptor proposed by Ojala et al. [20] has been also used in our approach. It has many attractive properties such as rotation invariance, low computational burden, and robustness against monotonic gray level transformation [21], [22]. In our approach, the LBP feature is computed as follows:

(a) The image is divided in windows of 8 × 8 pixels.
(b) Each pixel in a window, is compared to each of its 8 neighbours (on its left-top, left-middle, left-bottom, right-top, etc.), follow the pixels along a clockwise circle.
(c) Where the centre pixel's value is greater than the neighbour's value, the bit at the current position is 1 otherwise is 0. This gives an 8-digit binary number.
(d) Compute the histogram for the window; the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the centre) gives the feature vector for the window.
(e) PCA is applied to reduce the dimension to 24.

The last feature that we decide to take into account is the color. We extract the color features in the following manner:

(a) The image is divided in overlapped windows of 16 × 16

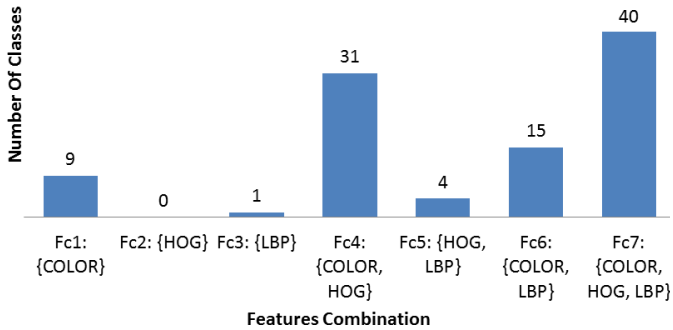


Fig. 2. Distribution of food classes recognized using the different features combinations.

pixels, densely sampled with a step of 6 pixels.

(b) Each window is subdivided in 2×2 blocks.

(c) Mean and variance on each of the RGB channel are extracted from the blocks. The final dimension of each local feature vector related to the color is 24 (4 blocks $\times$ 3 channels $\times$ 2 statistics).

(d) PCA is applied without dimension reduction with the aim to improve the classification.

B. Hierarchy of features

The aim of the proposed hierarchy is to find, for each class, the combination of features for optimal classification. The hierarchy of features is generated using an off-line learning process. Given a pool of N features, the total number of possible features combinations is:

$$M = \sum_{i=1}^{N} C(N,i) = \sum_{i=1}^{N} \frac{N!}{(i! \times (N-i)!)} \tag{1}$$

Each image of the training-set can be represented in M different ways (one for each features combination). We train a set of classifiers with a one-vs-rest strategy for all the food categories P and for each of the proposed M representations. The total number of classifiers generated is $P \times M$. All the images in the evaluation-set are classified using the generated classifiers and finally the per-class accuracy is analysed. Specifically, for each class, the feature combination that produces the maximum per-class accuracy is selected producing a histogram H containing the numbers of classes better recognized by each feature combination. An example of this histogram is shown in Fig. 2. The system creates the final hierarchy by applying a greedy selection strategy on this histogram. It selects first the feature that on its own is capable to classify the maximum number of classes, then the feature that combined with the previous one obtains the second maximum number of classes. This process repeats for the rest of the features. For each selected feature, the corresponding class indices are saved in the array Idx_i. For the histogram H of Fig. 2, the system extracts, in order, the following features: Color, HOG then LBP and the feature combinations generate at each level are $Idx_1 = \{1\}$, $Idx_2 = \{4\}$ and $Idx_3 = \{7,6,5,3\}$ with $Fc_1 = \{Color\}, Fc_2 = \{Hog\}, Fc_3 = \{LBP\}, Fc_4 = \{Color, Hog\}, Fc_5 = \{Hog, LBP\}, Fc_6 = \{Color, LBP\}, Fc_7 = \{Color, Hog, LBP\}$. By using this approach, simple classes are classified using just the feature in the first level, instead more complex classes will be classified using more features; eventually arriving at the leaf node of the hierarchy.

C. Representation

As mention earlier, the Fisher Vector [23] is used for representing a set of the local descriptors. Fisher Vector has been introduced to combine the benefits of generative and discriminative approaches. Let $X = \{x_t, t = 1...T\}$ be the set of T local descriptors extracted from an image. We assume that X can be modelled by a probability density function u_λ with parameters λ. X can be described by the gradient vector:

$$G_\lambda^X = \frac{1}{T} \nabla_\lambda \log u_\lambda(X) \tag{2}$$

This gradient vector can be classified by using any discriminative classifier. A natural kernel on these gradients is:

$$K(X,Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y \tag{3}$$

where F_λ is the Fisher information matrix of u_λ:

$$F_\lambda = E_{x \sim u_\lambda}[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)'] \tag{4}$$

Following [23] we used Fisher kernels on visual vocabularies, where the vocabularies of visual words are represented a Gaussian Mixture Model (GMM) as follows:

$$u_\lambda(x) = \sum_{i=1}^{K} w_i u_i(x) \tag{5}$$

We denote $\lambda = \{w_i, \mu_i, \Sigma_i, i...K\}$ where w_i, μ_i and Σ_i are respectively the mixture weight, mean vector and covariance matrix of Gaussian u_i. We assume that the covariance matrices are diagonal and we denote by σ_i^2 the variance vector. The GMM u_λ is trained on a large number of images using Maximum Likelihood (ML) estimation. It is supposed to describe the content of any image. We assume that x_t's are generated independently by u_λ. The probability that x_t belongs to the i-th component (estimation posterior probability) is given as follows:

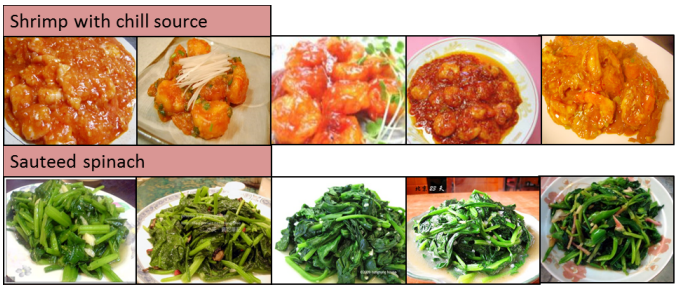
$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{j=1}^{K} w_j u_j(x_t)} \tag{6}$$

We consider the gradient with respect to the mean and standard deviation parameters that are defined as follows:

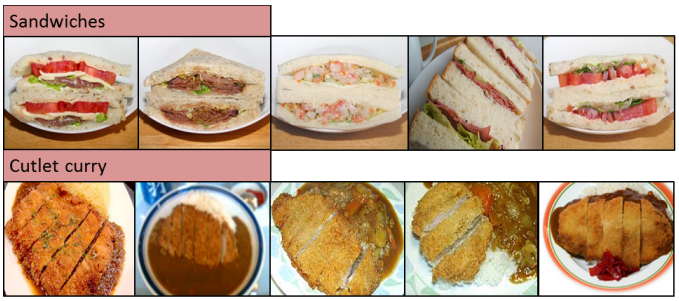
$$\Gamma_{\lambda,i}^X = \frac{1}{T\sqrt{w_i}} \sum t = 1^T \gamma_t(i) (\frac{x_t - \mu}{\sigma_i}) \tag{7}$$

$$\Gamma_{\sigma,i}^X = \frac{1}{T\sqrt{2w_i}} \sum t = 1^T \gamma_t(i) [\frac{(x_t - \mu)^2}{\sigma_i^2} - 1] \tag{8}$$

where the division between vectors is as a term-by-term operation. The final gradient vector Γ_λ^X is the concatenation of $\Gamma_{\lambda,i}^X$ and $\Gamma_{\sigma,i}^X$ vectors for $i = 1...K$ and is therefore 2KD-dimensional. Since its introduction, Fisher Vector has been improved in different ways to enhance its performance. One way is to apply an element-wise power normalization function, $f(z) = sign(z)|z|^\alpha$ where $0 \leq \alpha \leq 1$ is a parameter of the normalization. Another is to apply a L2 normalization on the Fisher Vector after applying the power normalization function. Finally PCA can be applied to the features before encoding them in the Fisher Vector. The PCA is crucial to obtain good classification performance [24]. By applying these



(a) Example of classes selected from the proposed approach and recognized using color information.



(b) Example of classes selected from the proposed approach and recognized using edge information.

(c) Example of classes selected from the proposed approach and recognized using a combination of color and texture information.

Fig. 3. Example of classes recognized by different cues

operations, Fisher Vector with a linear classifier obtain superiority compared to the popular combination of bag-of-features (BoF) and a non-linear SVM [24]. Moreover, since BoF needs a larger-size dictionary to improve recognition accuracy, (increasing computational cost for searching the nearest visual code words from the visual word dictionary), Fisher Vector is able to achieve a high recognition accuracy with even a small-size dictionary, and low computational complexity [19]. This property is a significant advantage for mobile devices. In the proposed approach, all the aforementioned improvements are considered. Moreover, each feature vector is reduced to 24 dimensions by PCA and the number of Gaussian used to compute the Fischer Vectors is 32.

D. Classification

Bag-of-features (BoF) with non-linear SVM is a common strategy used for the problem of food recognition [17], [14], [15], [13]. However, evaluation of non-linear SVM needs kernel computation between all the support vectors and the given feature vector, making it computationally complex. In addition, all the support vectors needed to be available in

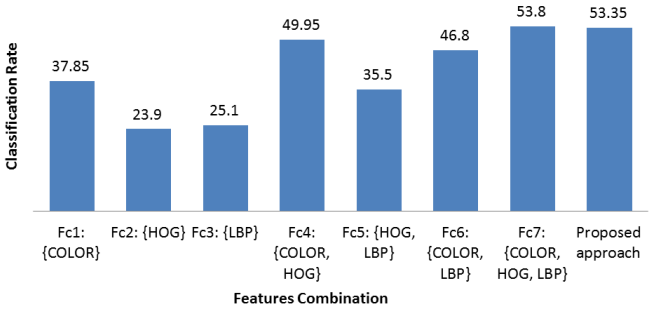


Fig. 4. Classification rate obtained when the images are represented using the different features combinations.

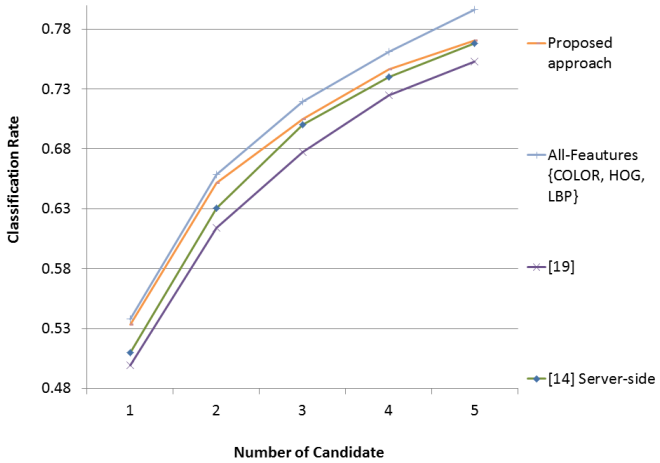


Fig. 5. Classification rate withing the top-n candidates obtained by the different approaches.

memory in order to compute the classification. For this reason, linear SVM is used as it only requires to calculate inner products, which is fast and has a low memory requirement. The computational cost of a linear SVM is in fact O(N). We adopt the one-vs-rest strategy for multi-class classification trained off-line with LIB-LINEAR [25]. For each class, we extract the best features combination (as described in section III-B) and then we train the linear classifiers by using all the image features in the corresponding category as positive samples and the remaining image features as negative samples.

IV. Experiments and Results

A. Classification Accuracy

In this section, we describe the experiments carried out in order to demonstrate the practical value of our method. In our experiments, we used the dataset "UEC-FOOD100" containing 100 food categories with more than 100 images per category. All the food items are marked with bounding boxes. Following the same protocol described in [19], we first extract the sub-region contained the food portion (through the bounding boxes marks) and then apply the food classification algorithm. The total number of food images in the dataset is 12,905. Moreover, we set the validation data and the test data for each category as 20 images while the rest of images are used for training the system. A cross-validation with five folders randomly selected is performed for each experiment. We used a Samsung Galaxy S5 (Quad Core with 2.5GHz and

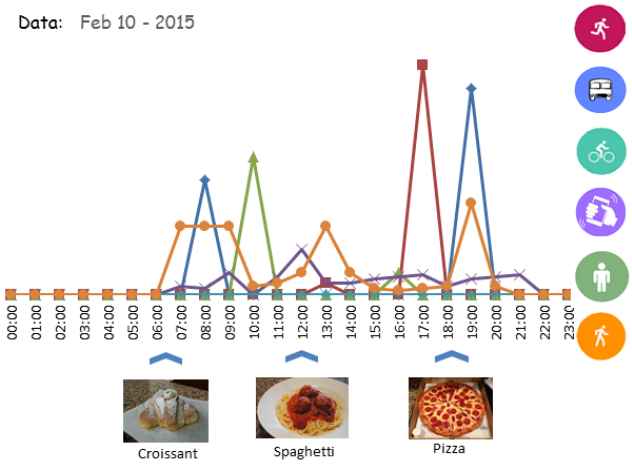


Fig. 6. Outputs obtained by integrating activity recognition with daily food intake recognition.

2GB of Memory) for measuring the processing time of the food recognition algorithm. Fig. 4 demonstrates the classification accuracy when the images are represented using the different features combinations. For food recognition, unsurprisingly, color is one of the most important visual cue. Fig. 3(a) illustrates some of the classes that are recognized by using this feature. Fig. 3(b) shows some of the classes that can be recognized using the edge information (HOG features) and in Fig. 3(c) those by using a combination of textures and colors. It is evident that in the first case, the system selects the classes that have a discriminative color pattern (like "Sauteed spinach" where the green is the predominant color or "Shrimp with chill source" where the red is the predominant color). In the second case, the classes that have discriminative edge patterns (like "Sandwich" with strong horizontal and oblique edges or "Cutlet curry" with oblique edges caused by the cuts) are determined. From the results in Fig. 3(a), we can also see that the best classification accuracy is obtained when the 3 features are always used in the system. However, using the proposed hierarchy, the accuracy is only reduced by 0.5% with a system 32% faster (see Table I). In Fig. 5, we show the classification accuracies achieved by the different approaches. It can be seen that our approach outperformed the classification results of [19] (+3.35% in case of the top-1 candidates) and obtained better results compared to a server-side system proposed in [14] (+2.3% in case of the top-1 candidates). Finally, in Table I, we show the computational time achieved by all different approaches. These results are obtained by averaging the classification time for all the images. Table I shows that our approach, although increases the number of features employed in the system (3 features respect 2 employed by [19]), it has similar computation time. The last column of Table I shows also the time performances obtained when multi-core optimization is enabled. This optimization is achieved by using the Intel TBB library [26] compiled for the Android operating system. The Android application that implements the proposed food recognition engine can be downloaded at the following URL: https://play.google.com/store/apps/details?id= org.imperial.amfoodrecognition

B. Activity recognition

To demonstrate how the proposed method can be combined with daily activity recognition, the mobile app is implemented

TABLE I. AVERAGE PROCESSING TIME OBTAINED ON THE DATABASE UEC-FOOD100

	Time using 1 core [sec]	Time using 4 core [sec]
Proposed approach	0.897	0.557
All Features	1.112	0.767
Our implementation of [19]	0.729	0.438

TABLE II. AVERAGE CONFUSION MATRIX FOR 6 DAILY ACTIVITIES OBTAINED BY THE PROPOSED SYSTEM

	Run.	Walk.	Cycl.	Cas.Mov.	Pub.Trans.	Stand.
Run.	**98.05**	1.36	0.00	0.59	0.00	0.00
Walk.	0.07	**96.68**	0.40	1.19	1.66	0.00
Cycl.	0.37	3.72	**93.63**	1.49	0.41	0.38
Cas.Mov.	0.50	5.40	3.80	**88.25**	1.37	0.69
Pub.Trans.	0.00	0.40	0.53	0.00	**94.15**	4.93
Stand.	0.00	0.00	0.00	0.00	15.03	**84.97**

with activity recognition for 6 common daily activities. Since the activity recognition should be executed continuously in the background of a smart phone, we develop an algorithm that requires a restricted amount of resources and minimal phone battery usage. The activity recognition is obtained by exploiting machine learning techniques on the inertial sensors data of the smart phone (i.e., gyroscope, accelerometer and magnetometer data). For learning the system, we record more than 70 sections of different physical activities collected by 5 people. The final database contains around 3 hours of data. The sensor data are captured using the SensorLog app available in the Android app store. The 6 daily activities taken into account in the proposed approach are: Public Transport, Running, Standing, Casual Movement, Cycling and Walking. There are more than 180 possible features related to the activity recognition that can be extracted from a time segment containing inertial sensor data. These features include statistical measurement (median, mean, variance, maximum, minimum etc.), first derivate, axes correlation, zero cross, mean cross, peak-to-peak distance, amplitude and even more complex features like the Skewness and Kurtosis of the signal. Due to resource constrains, we cannot use all of these features for real-time implementation. Therefore, we applied a wrapper feature selection approach [27] to identify the first 12 most discriminative features. These features are used to represent the data that we finally classify using a linear classifier. To test the validity of the proposed approach, we applied a five-cross validation on the considered database. The average confusion matrix obtained by the proposed solution is shown in Table II. In Fig. 6 we show the outputs obtained by integrating activity recognition with daily food intake recognition.

C. User study

In order to evaluate the performance of the proposed mobile app in real scenarios, we asked 5 participants to test our solution and compare it to the solution developed by [19] (available at the following URL: http://foodcam.mobi/FoodCam2.apk). These comparisons are all made in a free-living condition. In each session, the user will assign 2 points if the app finds the correct class within the top 5 food candidates, 1 point if a similar class is present although the correct class is not, and 0 point if none of the proposed classes in the output is related to the correct class. The cumulative scores of each user are showed in Fig. 7. These results confirmed the same finding obtained in performance evaluation that our proposed achieves

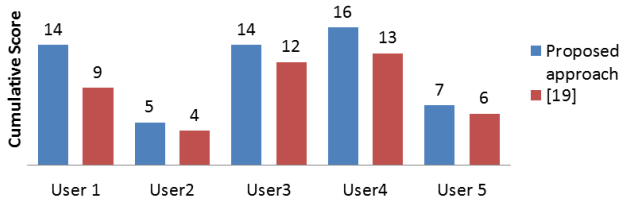


Fig. 7. Quality assessment of the proposed algorithm versus the solution in [19]. Each user evaluate the apps verifying the food classification output in real scenarios.

a higher accuracy. In order to make the app as reliable as possible, we trained the final model adopted in this experiment (also included in the released version of the app) through a variant of the original database where all the initial images, the images flipped horizontally, rotate by 90 and by 45 degree have been considered.

V. Conclusions

In this paper, we proposed an integrated framework for real-time food recognition by exploiting a hierarchy of visual features extracted from the smart phone. The proposed software environment is further integrated with daily activity recognition, allowing combined assessment of food intake and energy expenditure estimation by using a single app. The proposed approach has been compared to the state-of-the-art algorithms, demonstrating improved accuracy and ease of usability. Future work will be devoted to the measurement of the rate of food consumption and the incorporation of reinforcement learning for online adaptation of the classification algorithm for user-specific training and performance enhancement.

References

[1] F. E. Thompson and A. F. Subar, "Dietary assessment methodology," *Nutrition in the Prevention and Treatment of Disease*, vol. 2, pp. 3–39, 2008.

[2] A. E. Black and T. J. Cole, "Biased over-or under-reporting is characteristic of individuals whether over time or by different assessment methods," *Journal of the American Dietetic Association*, vol. 101, no. 1, pp. 70–80, 2001.

[3] F. E. Thompson, A. F. Subar, C. M. Loria, J. L. Reedy, and T. Baranowski, "Need for technological innovation in dietary assessment," *Journal of the American Dietetic Association*, vol. 110, no. 1, pp. 48–51, 2010.

[4] G.-Z. Yang, "Body sensor networks," 2nd Edition, Springer, 2014, ISBN 978-1-4471-6374-9.

[5] S. Passler and W. Fischer, "Food intake activity detection using a wearable microphone system," in *Intelligent Environments (IE), 2011 7th International Conference on*. IEEE, 2011, pp. 298–301.

[6] J. Liu, E. Johns, L. Atallah, C. Pettitt, B. Lo, G. Frost, and G.-Z. Yang, "An intelligent food-intake monitoring system using wearable sensors," in *Wearable and Implantable Body Sensor Networks (BSN), 2012 Ninth International Conference on*. IEEE, 2012, pp. 154–160.

[7] J. M. Fontana, M. Farooq, and E. Sazonov, "Automatic ingestion monitor: a novel wearable device for monitoring of ingestive behavior," 2013.

[8] G. Farinella, D. Ravì, V. Tomaselli, M. Guarnera, and S. Battiato, "Representing scenes for real-time context classification on mobile devices," *Pattern Recognition*, vol. 48, no. 4, pp. 1086–1100, 2015.

[9] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 6, pp. 790–808, 2012.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[12] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2249–2256.

[13] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning." in *ICIP*. IEEE, 2009, pp. 285–288.

[14] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Multimedia and Expo (ICME), 2012 IEEE International Conference on*. IEEE, 2012, pp. 25–30.

[15] Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2017–2020.

[16] Z. Zong, D. T. Nguyen, P. Ogunbona, and W. Li, "On the combination of local texture and global structure for food classification," in *Multimedia (ISM), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 204–211.

[17] G. M. Farinella, M. Moltisanti, and S. Battiato, "Classifying food images represented as bag of textons," in *IEEE International Conference on Image Processing*, 2014.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[19] Y. Kawano and K. Yanai, "Foodcam: A real-time food recognition system on a smartphone," *Multimedia Tools and Applications*, pp. 1–25, 2014.

[20] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[21] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[22] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern recognition*, vol. 42, no. 3, pp. 425–436, 2009.

[23] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[24] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 143–156.

[25] 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*, vol. 9, pp. 1871–1874, 2008.

[26] "Intel threading building blocks," https://www.threadingbuildingblocks.org/.

[27] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.