A supervised extreme learning committee for food recognition

A Supervised Extreme Learning Committee for Food Recognition[☆]

Niki Martinel*, Claudio Piciarelli, Christian Micheloni

[a] *Department of Mathematics and Computer Science, University of Udine, Udine (33100), Italy*

Abstract

Food recognition is an emerging topic in computer vision. The problem is being addressed especially in health-oriented systems where it is used as a support for food diary applications. The goal is to improve current food diaries, where the users have to manually insert their daily food intake, with an automatic recognition of the food type, quantity and consequent calories intake estimation. In addition to the classical recognition challenges, the food recognition problem is characterized by the absence of a rigid structure of the food and by large intra-class variations. To tackle such challenges, a food recognition system based on a committee classification is proposed. The aim is to provide a system capable of automatically choosing the optimal features for food recognition out of the existing plethora of available ones (e.g., color, texture, etc.). Following this idea, each committee member, i.e., an Extreme Learning Machine, is trained to specialize on a single feature type. Then, a Structural Support Vector Machine is exploited to produce the final ranking of possible matches by filtering out the irrelevant features and thus merging only the relevant ones. Experimental results show that the proposed system outperforms state-of-the-art works on four publicly available benchmark datasets.

Keywords: Food Recognition; Extreme Learning Machines; Structural SVM;

1. Introduction

According to the World Health Organization, in the last years there has been a rapid increase of diseases related to excessive or wrong food intake, most notably obesity and derived issues such as diabetes, cardiovascular diseases, musculoskeletal disorders and some types of cancers. In particular, it is estimated that in 2014 about 39% of the world's adult population were overweight, including a 13% of obese people, whose number more than doubled between 1980 and 2014. Contrary to popular belief, the problem also affects many low- and middle-income countries, particularly in urban settings [1].

Despite obesity being a complex disease involving many factors, from genetics to life styles, proper actions against it necessarily include a strict control over the daily food intake. Obese people should constantly take note of their daily meals, both for self-monitoring and to acquire useful statistics for dietitians. This justifies the large amount of food diary applications for mobile devices that have recently been developed [2, 3, 4]. However, these apps typically require a manual annotation of the food intake, a tedious task that often discourages the potential users. To face this problem, many food recognition works have been recently proposed, whose aim is to automatically classify food (and possibly its amount) directly from smartphone-acquired pictures.

Apart from the main health-oriented task, food recognition techniques can be applied in several other contexts as well. The recent rise in popularity of food-related TV shows, food blogs, etc. has lead to the production and sharing of a large amount of food-based multimedia (a trend sometimes referred to as "food porn" [5]). This information deserves proper tools for automatic search and classification, e.g. for image retrieval, user profiling, targeted advertising applications and so on. For example, it has become quite common for people to share pictures of their own meals on social networks, either on generic-purpose networks such as Facebook or image-oriented ones, such as Instagram or Pinterest. Automatic food recognition could help to identify the personal tastes of the users, in order to deliver finely-tuned advertising such as the best restaurants in the nearby that match the user's taste.

Regardless of the specific application, automatic food recognition is a tough problem with many specific challenges. Differing from other common image classification tasks, in food recognition there is no spatial layout information to be exploited. While for example human body recognition can benefit from prior knowledge on the spatial relationships between the parts to be detected (e.g. the head being always over the torso [6, 7, 8]) this is rarely

[☆]The work has been accepted for publication by Elsevier and is available at doi:10.1016/j.cviu.2016.01.012

*Corresponding author

Email addresses: niki.martinel@uniud.it (Niki Martinel), claudio.piciarelli@uniud.it (Claudio Piciarelli), christian.micheloni@uniud.it (Christian Micheloni)

the case when considering food. More generally, food is typically non-rigid, and thus no structure information can be easily exploited. Intra-class variation is another source of uncertainty, since the recipe itself for the same food can vary depending on the location, the available ingredients and, last but not least, the personal taste of the cook. Finally, inter-class confusion is a source of potential problems too. Different foods may look very similar, as in many soups where the main ingredients may be hidden below the liquid level. On the other hand, food images often have distinctive properties, especially in terms of colors and textures, which humans are able to exploit to recognize foods even from a single example, thus the task is still tractable, despite the non-trivial challenges.

A possible solution to sidestep the aforementioned problems might be a system that uses as many different features as possible but exploits only a subset of those to perform the food classification task. Following this idea, a food classification system based on a supervised learning committee is introduced.

As demonstrated in [9, 10, 11], learning with a committee has two main benefits: (i) a committee might exhibit performance unobtainable by an individual committee member on its own. This is due to the fact that individual errors made by the committee members cancel out to some degree when their predictions are combined; (ii) a committee of learning machines has modularity properties. Since different members can focus on a particular region in the input space, the mapping from input to target is not approximated by one estimator but by several estimators. Despite these benefits, since many different possible visual features can be used to address the task, they cannot be just integrated in a single feature vector of very high dimensionality. Indeed, this might yield to intractable computational loads as well as to the curse of dimensionality problem. To address such issues, the Supervised Extreme Learning Committee (SELC) approach is introduced.

SELC relies on a committee of Extreme Learning Machines (ELM) [12], where each ELM is trained with a specific feature type only. In this way, each member specializes on classifying a food only by using a certain feature type. This has the advantage of both reducing computational loads as well as to keep the committee learning benefits. Among all the possible neural-based learning systems [13], Extreme Learning Machines have been chosen for their excellent performances in terms of computational burden while maintaining a classification accuracy comparable to similar learning tools.

Committee-based approaches require the selection of a supervisor to fuse the discordant members' classifications. The typical output of the supervisor is a class. However, when classification results must be presented to users, a rank could be more appropriate. While ranking information can be obtained from single committee members, none of the existing works have adopted a supervisor considering it. Motivated by this, we introduce a Structural Support Vector Machine [14] as supervisor. It automatically selects the ranking produced by the members and combines them to obtain optimal classification performance as well as an optimal ranking.

The rest of the papers is organized as follows: in Section 2 we review the main state-of-the-art works in food recognition. Section 3 describes the proposed approach, explaining how ELMs can be applied to food classification and how the committee outputs can be merged into a final rank by means of a Structured Support Vector Machine. In Section 4 we give some comparative experimental results, showing how our system performs with respect to state-of-the-art methods. Finally, conclusions are drawn in Section 5.

2. Related Work

The topic of automatic food recognition has not been deeply investigated until recent years. The first works date back to late '90s, but they are limited to very specific contexts. For example the work by Jiménez *et al.* [15] focuses on automatic spherical fruit detection by means of a laser range-finder and image-based color and shape analysis to operate a robotic arm for fruit picking.

Most of the modern works on automatic classification of generic food images, typically for health-oriented applications, have been proposed since 2010. The work by Chen *et al.* [16] introduced a system exploiting different classifiers trained on multiple features. The authors compute both SIFT and LBP features with sparse coding for each image, as well as color histograms and Gabor filter responses to model the image colors and textures. A Support Vector Machine is trained for each texture separately, and the results are fused to form a single classifier using a multi-class AdaBoost algorithm. However, no details are given on the algorithm used for results fusion. A preliminary technique to estimate the amount of food using depth information computed via stereo matching techniques is also proposed. While being similar to our work, the results are fused to obtain optimal classification performance and not a "plausibility-rank" as in this work. Farinella *et al.* [17] exploit the texture information by applying a bank of rotation and scale invariant filters to each class of food images, in order to extract texture-oriented features known as Textons. The feature space is then quantized via K-means to create a codebook of textons for each class. All the textons prototypes are collected in a single visual dictionary which is used to represent each image as visual words distributions, effectively implementing a Bag of Textons approach. Finally, a Support Vector Machine is used in the classification stage. In [18], Yang *et al.* claim that spatial relationships between different ingredients could be exploited in the recognition of some types of food, as in a sandwich, where the meat is always between the bread slices. They perform a soft pixel-level segmentation of the image into eight ingredient types using a Semantic Texton

Forest. Then, they compute pairwise statistics over the detected local ingredients, such as distance, orientation, etc. The statistics are accumulated in a multi-dimensional histogram, which is then used as the input feature vector for a χ^2 kernel Support Vector Machine. The algorithm has been evaluated on the PFID dataset (see Section 4). Bossard *et al.* [19] believe that local information is crucial in food recognition. They introduced a weakly-supervised mining method which relies on Random Forests to extract relevant image patches (components) that are typical of specific foods. Recognition is performed by scoring image superpixels according to their similarity with the mined components, with a final multi-class SVM-based classification step. Their work is also notable for introducing the *Food-101* dataset. Also the nowadays popular deep learning techniques have been applied to food recognition tasks. For example, Kagaya *et al.* [20] trained a Convolutional Neural Network on the food images acquired by the FoodLog web service. They tuned the CNN parameters such as kernel size, number of layers etc. to achieve experimental results that outperformed traditional techniques such as SVM-based classification. By analyzing the resulting convolution kernels, they also observed that color seems to be a predominant feature in the specific task of food recognition. The authors also used the same approach to train a food detector, although this required the nontrivial creation of a non-food training set.

Many works are explicitly tuned for food diary applications on smartphones and other mobile devices [21, 22, 23]. Kawano and Yanai [21], for instance, are particularly concerned with real-time performances on an Android-based smartphone. To speed up the process, the user is asked to manually select a proper bounding box delimiting the food to be recognized. The bounding box is then adjusted based on the segmentation result by the GrabCut algorithm. The user also receives hints on how to move the camera to better acquire the food pictures. Then, the system extracts both color histograms and SURF-based Bag of Features, and uses them to assign the acquired image to one of 15 possible classes using a Support Vector Machine. Kong *et al.* [22] have developed DietCam, a smartphone-based system to help assessing daily food intakes. The system requires three images of each food to be recognized, to increase the robustness against partial occlusion or lighting conditions. Classification is done using a SIFT-based Bag of Visual Words, and then searching for the best match against a database of known foods using a nearest-neighbor classifier. The three pictures are also used to estimate the total food volume, although this requires a prior camera calibration and a reference object in the scene to reconstruct the correct scale. Once the type and amount of food are estimated, the system outputs the total amount of calories of the observed food. Zhu *et al.* propose a system for calories estimation via mobile phones in [23]. Their approach consists in segmenting the images using different techniques (connected component analysis, active contours and normalized cuts). Then, they extract both color and texture features using color histograms and Gabor filters, and classify the images using a Support Vector Machine. Volume is estimated by means of a single-view calibrated camera and a reference marker. Few works also exploit the additional data that can be acquired by sensors typically found on mobile devices, such as GPS. In [24] Bettadapura *et al.* focus on restaurant food recognition, and exploit the GPS position to identify the restaurant in which the photo is taken and automatically retrieve its menu, if available online. This information is exploited to limit the possible output categories, thus substantially improving the overall system performances. Their classifier is based on a Multiple Kernel Learning approach. Also Yanai *et al.* [25] exploit additional available information, although their goal is to mine reliable food pictures from Twitter streams. In their approach both textual and geotag data are used as hints to the type of food being depicted in the mined photos. Ravì *et al.* [26] integrate a food recognition system with a daily activity and energy expenditure estimator based on the inertial sensors available in mobile devices. The food recognition part is based on multiple feature (HoG, LBP, Color) analysis. Several combinations of features are evaluated in order to organize them in a feature hierarchy with the most relevant features at the highest hierarchy levels. Final classification is done using Fisher Vectors and linear SVM classifiers.

Finally, it is worth mentioning the works by Matsuda and Yanai [27, 28] that explicitly take in consideration the problem of multiple foods in the same picture. In particular, in [27] the authors detect several candidate regions by fusing outputs of different region detectors such as DPM, a circle detector and JSEG region segmentation algorithm. Then, food is recognized independently in each bounding box of the candidate regions using several features (Bag of SIFT, HoG, Gabor textures) and Support Vector Machines. The work is extended in [28], where the classification of each region is not done independently, but exploiting co-occurrence statistics on food items.

3. The Approach

3.1. System Overview

As shown in Figure 1, the proposed food recognition system consists of three main phases: (i) image feature representation, (ii) committee training/classification, and (iii) supervisor training/decision.

Since the proposed approach introduces a committee-based learning mechanism, a training phase is required. During such a phase, each training image is given to the feature extraction module that computes discriminative visual features capturing color, shape and texture information. To reduce the high dimensionality of a subset of such features, a codebook learning submodule that provides nonlinear feature encoding is exploited. The obtained encodings and the remaining features are finally considered as the image feature representation (details in

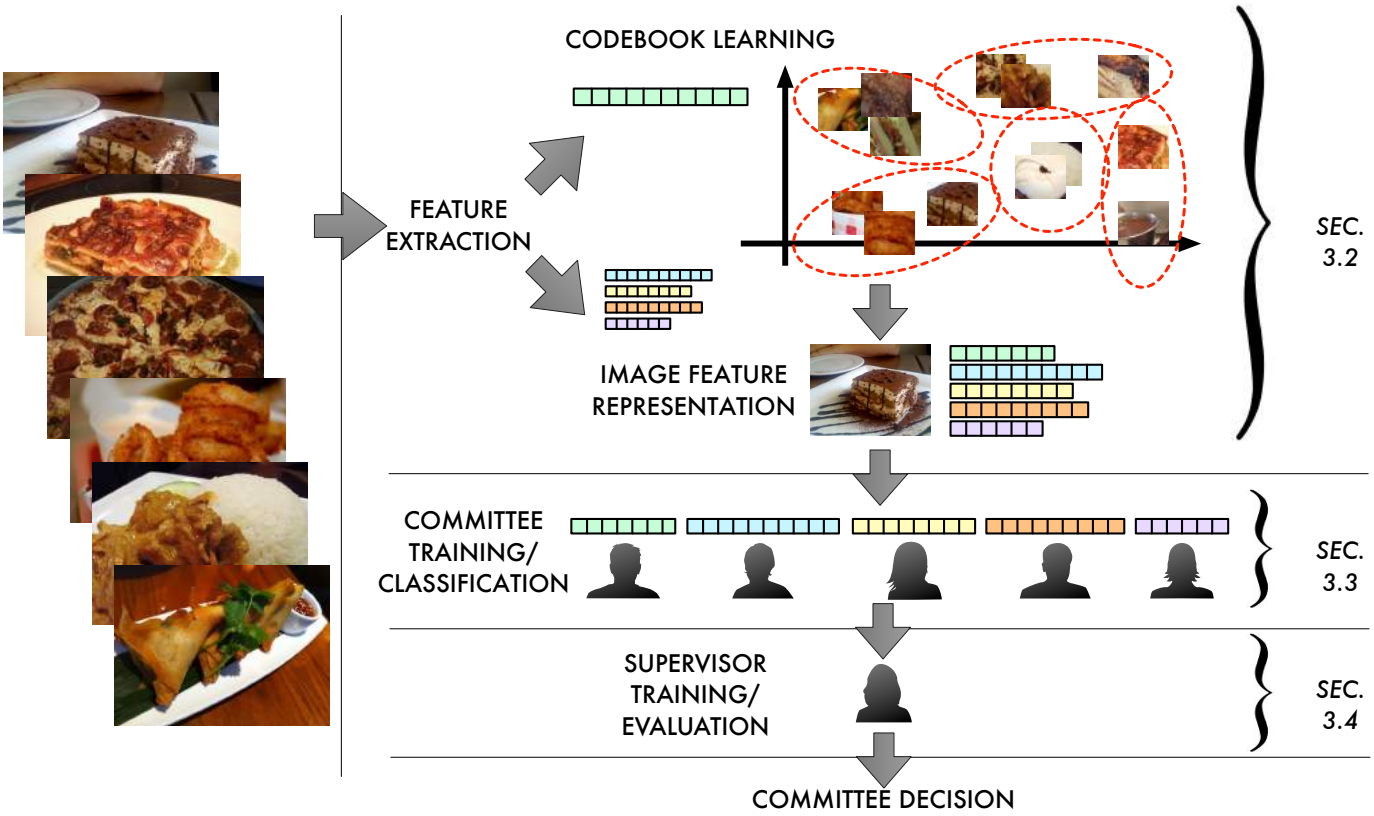


Figure 1: Proposed system architecture based on three main stages: (i) image feature representation; (ii) committee training and (iii) supervisor training. *(Best viewed in color)*

Section 3.2). Then, the whole set of such representations is given to the committee training module (Section 3.3) where each committee member specializes in classifying the food by exploiting a single type of feature only (i.e., either color, shape or texture). Once the committee members are trained, their answers are evaluated by a committee supervisor whose aim is to learn how to properly combine them such that the optimal ranking is obtained (Section 3.4).

During the recognition phase, given a test image to classify, the same features are extracted. The nonlinear encoding procedure is applied to a subset of those by using the learned codebook. Then, the obtained representation is provided to the committee members for classification. Each member produces a classification considering only the type of feature it has been trained with. Finally, the members answers are given to the committee supervisor which combines them and produces the final decision (ranking).

3.2. Image Feature Representation

Literature in food recognition [17, 28, 18, 29] usually represents a food image as a combination of different features (e.g., color, shape, spatial relationships, etc.). Recent works [30, 17] have also shown the benefits of feature encodings [31, 32, 33] for the same task.

Despite this, there is no clear statement suggesting which are the best features for food recognition. The SELC approach has been designed to tackle such interesting problem. It aims to autonomously select only the relevant features out of a large pool of given ones. Thus, to obtain the image feature representation, a large set of features has been considered. This includes (i) color, (ii) shape, (iii) texture, (iv) local and (v) data-driven features.

Color: Due to their rotation and location invariant properties, color histogram features are the most widely used features to capture the global appearance of an image. However, as shown in [34, 35], histograms are discriminative feature representations of a datum only if the input image is projected in an appropriate color space. By following the advices in [34, 35], in the current framework the HSV, CIELab, RGB, normalized RGB and Opponent color spaces have been considered. Thus, for a given image $\mathbf{I}$, a histogram is extracted from each of such color space components. Histograms belonging to the same color space are finally concatenated.

Shape: To capture the shape of a given image the Pyramid Histogram of Oriented Gradients (PHOG) [36] and the GIST features [37] are used. The PHOG feature captures the local shape and the spatial layout of the shape in a given image [36] by exploiting the pyramidal framework proposed in [38]. The GIST feature models the shape

of an image by computing the dominant spatial structure of a very low dimensional representation of the image itself (i.e., the Spatial Envelope).

Texture: As pointed out in [17], texture features are of fundamental importance for food recognition. Indeed, looking at food images, its reasonable to claim that the food recognition problem is closely related to that of texture discrimination.

To capture the texture information, Local Binary Pattern (LBP) [39], Local Phase Quantization (LPQ) [40], Local Configuration Pattern (LCP) [41], Pairwise Rotation Invariant LBP (PRICoLBP) [42], Binary Gabor Patterns [43] and textons features [31, 32, 33] are used. These have been selected on the basis of the fact that each of them captures different texture aspects which can be worth to inspect for food recognition.

In particular, to extract textons features, the MRS4 filter bank [32] has been considered for textons encoding. To achieve invariance to affine illumination transformations, the z-scores of the image intensity are computed before convolution with the ℓ_1 normalized filters. The obtained responses at each pixel location are then contrast normalized as in [17].

Differently from previous works in the field using standard textons encodings schemes like [30, 17], in this work a more robust feature encoding based on the Improved Fisher Vector (IFV) technique [44, 45, 46] is used. The IFV method suppresses the lossy process [47] of common Bag-of-Words (BoW) hard quantization methods which yield to performance degradation. In a nutshell, given a set of training images, after convolution with the MRS4 filters, the obtained filter responses are clustered by means of a Gaussian Mixture Model. Then, for each image, every feature descriptor (i.e., the filter response at a location) is assigned to a particular mode in the mixture with a strength given by the posterior probability. In addition, for each mode the mean and covariance deviation vectors are considered. The result of such a process, computed for every feature response, yields to the encoded feature representation.

Local features: Local feature are particular image regions which are different from their surroundings. Such features lead to a powerful and discriminative image representation that has been widely applied in a large range of applications. In this work, we have exploited three different types of local features, which also consider color information, namely: (i) DSP-SIFT [48], (ii) OpponentSIFT [35], and (iii) C-SIFT [35].

To obtain a fixed-length feature representation for each image, we followed [24] and used a standard Bag-of-Words approach. To avoid very high dimensional feature representations, which may yield to curse of dimensionality issues, the IFV approach has not been used in such a case.

Data-Driven: All the aforementioned features are the result of an hand-craft designing process that is conducted by humans on the basis of the a priori knowledge of the problem. However, in the recent past such a task has been widely obscured by the now well known feature learning procedure. With such a task, a machine learning algorithm is trained to learn the most suitable image representation for the given classification/regression task. Following such a motivation, we have considered feature representations that were learned from natural images. Specifically, we have followed [49] to compute the data-driven feature representation: the given image is fed to the OverFeat Convolutional Neural Network (CNN) [50]. Then, the CNN features are taken from the output of the last convolutional layer.

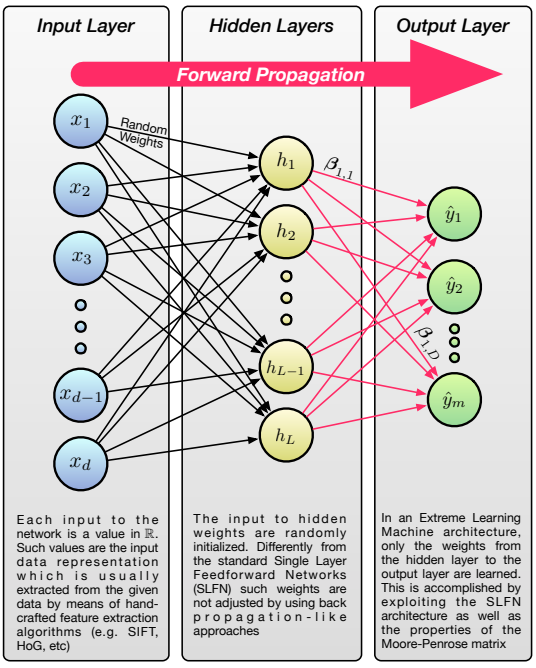


Figure 2: General architecture of a standard Extreme Learning Machine. The input-to-hidden weights are randomly generated whereas the hidden-to-output weights are learned analytically, without the need of an iterative process.

3.3. Extreme Learning Machines

An Extreme Learning Machine (ELM) [51] is a particular Single Layer Feed-forward Network (SLFN) which was inspired by biological learning and proposed to overcome the issues faced by back propagation-based learning algorithms. In particular, the hidden nodes of an ELM compute random combinations of the input values. Thus, the input-to-hidden weights do not need to be learned with computationally expensive learning algorithms and can be randomly generated, independently from the input data (see Figure 2). In such a network, it is proven that the universal approximation theorem still holds under mild assumptions, provided that the hidden layer has enough nodes [51]. Computing the final output is then just a matter of finding the optimal hidden-to-output weights, which can be conveniently done in an analytical way, without the need of computationally expensive iterative processes, such as backpropagation.

More formally, let $\mathbf{x}$ be a d-dimensional row feature vector belonging to one out of m possible classes. To such a feature vector corresponds a class label $\mathbf{y}$ represented by a m-dimensional unit row vector. Its single positive c-th component, denoted as y_c, indicates that $\mathbf{x}$ belongs to class $c \in \mathcal{C} = \{1, \ldots, m\}$. The vector $\mathbf{x}$ is the input for the ELM. Thus, the value of the j-th input neuron corresponds to the j-th component in a data sample $\mathbf{x}$.

Let the hidden layer be composed of L hidden neurons and let its connection with the input neurons be denoted as the random matrix $\mathbf{R} \in \mathbb{R}^{d \times L}$. The output of the j-th neuron in the hidden layer is computed as $h_j(\mathbf{x}) = \sigma(\mathbf{x}, \mathbf{R}_j, b_j)$, where $\mathbf{R}_j$ is the j-th column of $\mathbf{R}$, b_j is a random bias parameter associated with the j-th hidden neuron and $\sigma(\cdot)$ is a differentiable activation function, such as the Sigmoid:

$$\sigma(\mathbf{x}, \mathbf{R}_j, b_j) = \frac{1}{1 + \exp(-(\mathbf{x}^T \mathbf{R}_j + b_j))} \tag{1}$$

Let $\boldsymbol{\beta} \in \mathbb{R}^{L \times m}$ be the matrix of weights connecting the hidden layer composed of L nodes with the m output nodes and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]$ the output row vector of the hidden layer with respect to the input $\mathbf{x}$. The output of the network is then defined as:

$$\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{2}$$

The function $\mathbf{h}(\mathbf{x})$ maps the data from the d-dimensional input space to the L-dimensional hidden layer random feature space where the input-to-hidden node weights $\mathbf{R}$ are randomly generated according to any continuous sampling probability distribution. Ideally, to not lose valuable discriminative property of the input data, the random weights (i.e., the projection matrix) $\mathbf{R}$ should provide a stable embedding that approximately preserves the distance between all pairs of original features. As proved in [52], if the original points are projected onto a randomly selected subspace with suitably high dimensions, then the Johnson-Lindenstrauss lemma [53] is satisfied with high probability and thus the distances between the points in the random space are preserved. As shown [54], a matrix satisfying such restricted isometry property is the random Gaussian matrix where each element of the random projection matrix is normally distributed: $r_{i,j} \sim \mathcal{N}(0, 1)$. Under these assumptions, the hidden layer output mapping $\mathbf{h}(\mathbf{x})$ has the universal approximation capability property, i.e. for any $\epsilon > 0$ arbitrarily small, there exists a network with a proper number of hidden nodes L and a weight matrix $\boldsymbol{\beta}$ such that

$$\|\mathbf{h}(\mathbf{x})\boldsymbol{\beta} - \mathbf{y}\| < \epsilon \tag{3}$$

where $\mathbf{y}$ is the correct output value for input $\mathbf{x}$.

3.3.1. ELM Learning

Let $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$ be the set of n training samples pairs. Since the input-to-hidden weights are randomly generated, they do not require tuning, and the learning algorithm consists in finding a proper hidden-to-output weight matrix $\boldsymbol{\beta}$ such that

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}. \tag{4}$$

where

$$\mathbf{H} = [\mathbf{h}(\mathbf{x}^{(1)}), \ldots, \mathbf{h}(\mathbf{x}^{(n)})]^T \in \mathbb{R}^{n \times L}$$
$$\mathbf{Y} = [\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}]^T \in \mathbb{R}^{n \times m}$$

Since, in general, $\mathbf{H}$ is not a square matrix, an exact solution may not exist. An approximate solution however can be found by solving the following minimization problem:

$$\arg\min_{\hat{\boldsymbol{\beta}}} \|\mathbf{H}\hat{\boldsymbol{\beta}} - \mathbf{Y}\| \tag{5}$$

which is a standard least-squares problem whose solution can be found by using the orthogonal projection method [55]:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^T \left(\mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{Y} \tag{6}$$
$$= \left(\mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T\mathbf{Y} \tag{7}$$

where eq.(6) holds when $\mathbf{H}\mathbf{H}^T$ is nonsingular and eq.(7) is valid when $\mathbf{H}^T\mathbf{H}$ is nonsingular.

In addition, to achieve a stabler solution and to obtain better generalization performance [12, 56], the ridge regression theory [57] can be exploited, and a positive value $1/C$ can be added to the diagonal elements of $\mathbf{H}\mathbf{H}^T$. By considering this, and substituting eq.(6) in eq.(2), the predicted output of a new sample can be computed as

$$\hat{\mathbf{y}} = \mathbf{h}(\mathbf{x})\mathbf{H}^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{Y}. \tag{8}$$

3.3.2. Kernel ELM

Standard ELMs perform an initial projection onto the ELM random feature space by means of a linear mapping. To obtain a stable embedding that preserves the distance between original data points such a mapping is randomly generated according to any continuous sampling probability distribution. However, it is worth noting that the each element of the matrix $\mathbf{H}\mathbf{H}^T$ in eq.(8) is a dot product in the random ELM hidden layer space, i.e.

$$(\mathbf{H}\mathbf{H}^T)_{i,j} = \mathbf{h}(\mathbf{x}^{(i)}) \cdot \mathbf{h}(\mathbf{x}^{(j)}) \tag{9}$$

following the kernel methods theory, this dot product can be replaced by a *kernel function* $\phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ which implicitly computes dot products in the random ELM feature space without explicitly knowing the mapping function h. The matrix $\mathbf{H}\mathbf{H}^T$ can be replaced by the *kernel matrix* $\boldsymbol{\Phi}$ such that $\boldsymbol{\Phi}_{i,j} = \phi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$. Thus, eq.(8) can be written as

$$\hat{\mathbf{y}} = \begin{bmatrix} \phi(\hat{\mathbf{x}}, \mathbf{x}^{(1)}) \\ \vdots \\ \phi(\hat{\mathbf{x}}, \mathbf{x}^{(n)}) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \boldsymbol{\Phi}\right)^{-1} \mathbf{Y}. \tag{10}$$

The main advantage of eq.(10) over eq.(8) is that the number of nodes in the hidden layers is no more a parameter of the system and thus does not require tuning. As a side note, observe that eq.(10) is similar to the one obtained using a Least Squares SVM, and in fact, as demonstrated in [12], ELM can be interpreted as a generalization of a large group of classifiers such as LS-SVM, Proximal SVM and kernel Ridge Regression.

3.3.3. ELM Committee

As a result of the ELM training procedure carried out with the kernel extension, the set of hidden-to-output layer connection weights are learned. Such weights are learned by exploiting the set $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$ of n d-dimensional features. A feature vector $\mathbf{x}^{(i)}$ is typically computed by concatenating all the feature vectors deriving from different cues (e.g., color histograms, Histogram of Oriented Gradients, etc.). While such an approach is widely and successfully adopted, it may introduce several problems. Indeed, if many multiple features are considered to represent the input data, then the overall joint feature dimension may be very large. Thus, the computational load is increased. In addition, low dimensional features are usually dominated by high dimensional ones, hence these are somehow considered as more important for discriminating between input data patterns.

To address the aforementioned problems an approach that separately considers the different types of features is proposed. Such an approach is named ELM committee and works as follows. Let $\{(\mathbf{x}_{*}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$ be the set of n training samples pairs, where $\mathbf{x}_{*}^{(i)} = \{\mathbf{x}_{k}^{(i)} : \mathbf{x}_{k}^{(i)} \in \mathbb{R}^{d_k}, k = 1, \ldots, K\}$ denotes the set of K different feature types (e.g., color histogram, Histogram of Oriented Gradients, etc.) extracted for the i-th training data sample and d_k indicates the dimensionality of the k-the feature type. Thus, it is not necessary that, features of different type span the same space, i.e., $|\mathbf{x}_{1}^{(i)}|$ may be different to $|\mathbf{x}_{2}^{(i)}|$.

In the ELM committee approach, an ELM is required to learn the optimal hidden-to-output connection weights $\hat{\boldsymbol{\beta}}$ for each feature type separately. Therefore, K different sets of connection weights are learned and the K predicted outputs $\hat{\mathbf{y}}_k$ are defined as

$$\hat{\mathbf{y}}_k = \begin{bmatrix} \Phi_k(\hat{\mathbf{x}}, \mathbf{x}_k^{(1)}) \\ \vdots \\ \Phi_k(\hat{\mathbf{x}}, \mathbf{x}_k^{(n)}) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \boldsymbol{\Phi}_k \right)^{-1} \mathbf{Y} \qquad (11)$$

where $\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}$ and $\boldsymbol{\Phi}_k$ are the training data samples of type k and the corresponding training kernel matrix, respectively. Such a procedure is highly parallelizable and can easily scale to high-dimensional problems.

3.4. Committee Supervisor

In the previous sections a learning approach to separately model K different types of features has been introduced. This reduces the computational loads and allows to better handle the problems of dominating high dimensional features. Despite such benefits, since an answer is given by each committee member (i.e., each single ELM), there should be an appropriate pooling procedure that collects them to provide a final decision. Such a task is accomplished by the committee supervisor. The committee supervisor can be as simple as a pooling operator (e.g., average pooling, max pooling, etc.). The common output of the supervisor is a class label. However, we believe that when classification results have to be presented to users (as in the case of food recognition), a ranking could be more appropriate. Towards this objective, we introduce a supervisor that aims to learn the coefficients $\boldsymbol{\alpha}$ of the linear combination of the $k = 1, \ldots, K$ member answers $\hat{\mathbf{y}}_k$ such that an optimal ranking can be obtained. A Structural Support Vector Machine (i.e., the supervisor) has been selected for such a task. The overall architecture is shown in figure 3.

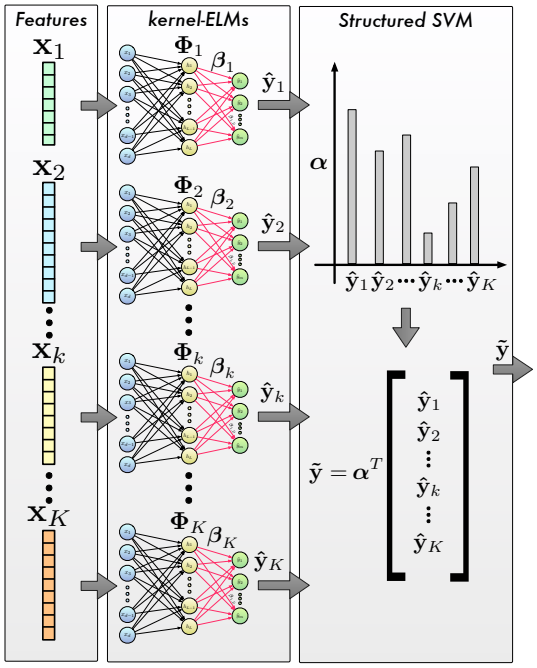


Figure 3: Architecture of the proposed Supervised Extreme Learning Committee approach.

3.4.1. The Structural Supervisor Objective

Let $\mathcal{X}$ and $\mathcal{O}$ denote the input feature (i.e., $\mathbf{x} \in \mathcal{X}$) and the output (i.e., $\mathbf{o} \in \mathcal{O}$) spaces, respectively. The idea behind Structural SVM [58, 14, 59] is to discriminatively learn a scoring function $f : \mathcal{X} \times \mathcal{O} \rightarrow \mathbb{R}$ over input/output pairs, where the space of the outputs $\mathcal{O}$ is no longer restricted to contain only numbered labels (as in common classification problems), but it is a structured output space whose elements may be sequences, strings, lattices, etc. In SELC, the structured output space consists in a ranking of the considered classes. Since the true class should be always ranked first, such structured output space also yields to optimal classification performance.

As before, let $\mathcal{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$ be the input set and $c^{(i)} \in \mathcal{C} = \{1, \dots, m\}$ denote the class of the i-th sample. For a given sample $\mathbf{x}^{(i)}$, the objective is to learn the coefficients $\boldsymbol{\alpha}$ that order relevant classes $\mathcal{C}^{(i)+} \subseteq \mathcal{C}$ (i.e., classes "similar" the same class of the sample) before irrelevant ones $\mathcal{C}^{(i)-} \subseteq \mathcal{C}$ (i.e., classes "different" from the class of the sample).

However, in common classification problems there is only knowledge of the order between the relevant and irrelevant classes, but not of the order within relevant or irrelevant ones. To overcome such an issue, we consider the query-class set of a sample $\mathbf{x}^{(i)}$ as a partially ordered set, where the partial order $\mathbf{o}^{(i)}$ is defined as

$$\mathbf{o}^{(i)} = \{o^{(i)^+,(i)^-}\}, \quad o^{(i)^+,(i)^-} = \begin{cases} +1 & \text{if} \quad c^{(i)+} \prec c^{(i)-} \\ -1 & \text{if} \quad c^{(i)+} \succ c^{(i)-} \end{cases} \tag{12}$$

where $c^{(i)+} \prec c^{(i)-}$ indicates that a relevant class $c^{(i)+} \in \mathcal{C}^{(i)+}$ is ranked before an irrelevant one $c^{(i)-} \in \mathcal{C}^{(i)-}$, and after otherwise.

Armed with the partial order definition, we can define the Structural SVM with slack rescaling objective [59] as

$$\begin{aligned} &\min_{\boldsymbol{\alpha}, \xi \geq 0} \frac{1}{2}\|\boldsymbol{\alpha}\|^2 + \frac{\gamma}{n}\sum_{i=1}^{n}\xi_i \\ &\text{s.t.} \quad \forall i, \forall \tilde{\mathbf{o}}^{(i)} \in \mathcal{O} \setminus \mathbf{o}^{(i)} : \\ &\langle \boldsymbol{\alpha}, \Psi(\mathbf{x}^{(i)}, \mathcal{C}; \mathbf{o}^{(i)}) - \Psi(\mathbf{x}^{(i)}, \mathcal{C}; \tilde{\mathbf{o}}^{(i)}) \rangle \geq 1 - \frac{\xi_i}{\Delta(\mathbf{o}^{(i)}, \tilde{\mathbf{o}}^{(i)})} \end{aligned} \tag{13}$$

where γ is a parameter that controls the trade-off between the norm of the coefficients $\boldsymbol{\alpha}$ and the average of the slack variables ξ_i. $\Psi(\cdot)$ is the combined feature representation (details in the following) and $\mathcal{O}$ is the space consisting of all possible partial orders. Within such a space, $\mathbf{o}^{(i)}$ denotes a correct partial order that ranks all relevant classes before irrelevant ones while $\tilde{\mathbf{o}}^{(i)}$ is an incorrect partial order that violates some of the pairwise relations. Finally, $\Delta(\mathbf{o}^{(i)}, \tilde{\mathbf{o}}^{(i)})$ is a suitable loss function that quantifies the loss associated with predicting a wrong partial order.

The constraints in eq.(13) state that, for each sample, the score $\langle \boldsymbol{\alpha}, \Psi(\mathbf{x}^{(i)}, \mathcal{C}; \mathbf{o}^{(i)}) \rangle$ of a correct order $\mathbf{o}^{(i)}$ must be greater than the score $\langle \boldsymbol{\alpha}, \Psi(\mathbf{x}^{(i)}, \mathcal{C}; \tilde{\mathbf{o}}^{(i)})) \rangle$ of all incorrect orders $\tilde{\mathbf{o}}^{(i)}$ by a required margin. This margin equals 1 in the slack-rescaling formulation.

3.4.2. Supervisor Learning

As shown in eq.(13), three main components have to be defined in order to properly achieve the final objective:

(i) The combined feature representation for inputs and outputs: $\Psi(\cdot)$;

(ii) The function used to compute the loss between a wrong and a correct partial order: $\Delta(\cdot, \cdot)$;

(iii) A separation oracle to optimize the given objective by means of the cutting plane approach.

The Combined Feature Representation

The flexibility in designing $\Psi(\cdot)$ has strongly pushed the adoption of Structural SVMs to attack a wide plethora of problems like natural language parsing [58], object detection [60] and segmentation [61], just to name a few. Therefore, its choice is closely dependent to the task that one wants to address.

Since in the current approach only relevant and irrelevant pairs relationships are known, a modification of the commonly adopted partial order feature [62, 63] is used. This, denoted $\Psi(\mathbf{x}^{(i)}, \mathcal{C}; \mathbf{o}^{(i)})$, is computed as

$$\sum_{i^+=1}^{|\mathcal{C}^{(i)+}|}\sum_{i^-=1}^{|\mathcal{C}^{(i)-}|} o^{(i)^+,(i)^-} \frac{(\boldsymbol{\psi}(\mathbf{x}^{(i)}, c^{(i)+}) - \boldsymbol{\psi}(\mathbf{x}^{(i)}, c^{(i)-}))}{|\mathcal{C}^{(i)+}||\mathcal{C}^{(i)-}|}. \tag{14}$$

In the proposed case the order should be optimized over the committee members decisions, thus the feature $\boldsymbol{\psi}$ is represented by the output of the K committee members

$$\boldsymbol{\psi}(\mathbf{x}^{(i)}, c^{(i)}) = \left[\hat{y}^1_{c^{(i)}}, \dots, \hat{y}^K_{c^{(i)}}\right]^T \tag{15}$$

where $\hat{y}^k_{c^{(i)}}$ is the output computed by the k-th member with respect to class label $c^{(i)}$. Such partial order feature is suitable for the proposed objective because it only depends on the difference between relevant and irrelevant pairs, not the entire list. By adding the differences of correct orders and subtracting that of incorrect orders, the partial order feature emphasizes the directions in feature space which are closely related to correct ordering.

The Loss Function

Similarly to the selection of a suitable combined feature representation, the choice of the loss function for Structural SVMs is also highly dependent on the task. Among all the possible loss functions that can be used, the area under curve (AUC) measure [63, 64] is the more appropriate for the proposed approach. Indeed, it allows to characterize the difference between relevant and and irrelevant pairs with only partial order available.

As shown in [63, 64], computing the AUC requires computing a ranking. This can be naturally obtained by ordering each example according to $\langle \boldsymbol{\alpha}, \boldsymbol{\psi}(\mathbf{x}^{(i)}, c^{(i)}) \rangle$, for all $c^{(i)} \in \mathcal{C}$. From such a ranking, the partial ordering $\tilde{\mathbf{o}}^{(i)}$ can be computed and the AUC loss calculated as

$$\Delta(\mathbf{o}^{(i)}, \tilde{\mathbf{o}}^{(i)}) = \sum_{i^+}^{|c^{(i)+}|}\sum_{i^-}^{|\mathcal{C}^{(i)-}|} \frac{\mathbf{1}_{\left(o^{(i)^+,(i)^-} \neq \tilde{o}^{(i)+,(i)-}\right)}}{|\mathcal{C}^{(i)+}||\mathcal{C}^{(i)-}|} \tag{16}$$

where $\mathbf{1}_{(\cdot)}$ is the indicator function. Thus, the AUC loss function tells, on average, how many incorrect orders are obtained with the current partial ordering $\tilde{\mathbf{o}}^{(i)}$.

The Separation Oracle

As shown in [63], learning a ranking function that optimizes an upper bound on the AUC loss on the training set requires a constraint for every possible wrong output

$\tilde{\mathbf{o}}^{(i)}$. Unfortunately, the number of possible wrong outputs is exponential in the size of $\mathcal{C}$. Such a problem can be addressed by adopting a cutting plane algorithm which iteratively introduces constraints until the original problem is solved within a desired tolerance [14]. In such a case, one key step is to efficiently determine the separation oracle. Given a fixed $\boldsymbol{\alpha}$, for each example $\mathbf{x}^{(i)}$ the separation oracle aims to find the output $\dot{\mathbf{o}}^{(i)}$ associated with the most violated constraint, i.e.

$$\dot{\mathbf{o}}^{(i)} = \arg\max_{\dot{\mathbf{o}}^{(i)} \in \mathcal{O}} \langle \boldsymbol{\alpha}, \Psi(\mathbf{x}^{(i)}, \mathcal{C}; \mathbf{o}^{(i)}) \rangle + \Delta(\mathbf{o}^{(i)}, \dot{\mathbf{o}}^{(i)}). \quad (17)$$

For a fixed $\boldsymbol{\alpha}$, the argument maximizing eq.(17) can be found by sorting the committee answers by $\langle \boldsymbol{\alpha}, \boldsymbol{\psi}(\mathbf{x}^{(i)}, c^{(i)}) \rangle$ in descending order. This strongly reduces the computational complexity as the maximization objective in eq.(17) only requires $O(n \log n)$ processing time.

3.5. The Supervised Extreme Learning Committee Decision

Once the training procedure is completed, the learned parameters can be adopted to obtain the ranking for a new test data sample $\hat{\mathbf{x}}$. First, the committee members are asked to produce K answers $[\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K]$. Then, the learned supervisor coefficients $\boldsymbol{\alpha}$ are used to weights such answers. As a result, the output of the Supervised Extreme Learning Committee is computed as

$$\tilde{\mathbf{y}} = \boldsymbol{\alpha}^T \begin{bmatrix} \hat{\mathbf{y}}_1^T \\ \vdots \\ \hat{\mathbf{y}}_K^T \end{bmatrix}. \quad (18)$$

The ranking for the test data sample is finally obtained by sorting in descending order the elements in $\tilde{\mathbf{y}}$.

4. Experimental Results

To validate the proposed SELC approach, results on four benchmark datasets for food recognition have been computed. For each of them, an analysis of the performances of the selected features as well as on the benefits of the proposed approach with respect to standard ELMs is conducted first. Then, the advantages of the proposed supervisor with respect to other approaches are shown. Finally, comparisons with state-of-the-art methods are presented to show the superior performance of SELC.

As commonly performed in the evaluation of food recognition approaches [65, 30, 17], the achieved results are shown in terms of recognition accuracy and by means of the *top-n* criterion [30]. The *top-n* criterion defines the chance of obtaining a correct recognition within the first n retrieved images.

The performances achieved by the existing methods have been taken from the corresponding works or have been directly provided by the authors.

4.1. Experimental Settings

To evaluate the performance of our approach we have adopted the following settings. All the given parameters have been selected through 4-fold cross validation.

4.1.1. Image Feature Representation:

Color:
For every color channel a 32 bin histogram is extracted. Thus, the set of color histogram features consists of 5 histograms (i.e., one for each color space) each of which has dimensionality equal to 96.

Shape:

1. PHOG: features have been extracted from each color channel of a given image which has been projected onto the HSV color space first [66]. HOG quantized considering 9 bins have been extracted considering 3 levels of the spatial pyramid. The resulting PHOG feature vector consists of 2295 elements.
2. GIST: the same implementation settings used in [37] have been adopted to get the 512-D feature vector.

Texture:

1. LBP: the uniform rotation invariant descriptor [39] has been extracted considering 8-neighbors and a radius of 1. The resulting vector consists of 59 elements.
2. LPQ: the basic LPQ version [40] with decorrelation and SIFT uniform 3×3 window for local frequency estimation has been used. The resulting 256-D vector contains the histogram of the LPQ codewords.
3. LCP: a feature vector of 81-D has been extracted considering 8 neighborhoods and a radius of 2.
4. BGP: The 216-D vector has been computed using the same settings in [43].
5. PRICoLBP: The code released with [42] has been used to extract the 1770-D feature vector.
6. Textons: 300 Gaussian Mixtures have been considered to quantize the filter responses, hence to learn the codebook.

Apart from textons, which have been separately extracted from each channel of the the RGB color space, all other texture features have been computed from grayscale image representations.

Local:

1. DSP-SIFT: The DSP-SIFT descriptor [48] introduces a simple modification of the original SIFT one. Specifically, gradient orientations of a grayscale image are pooled across different domain sizes in addition to the usual spatial locations. The descriptor computed for each detected keypoint lies in a 128-D space.
2. OppSIFT: The 384-D descriptor computed for each detected interest point describes all of the channels in the opponent color space. Due to the normalization of the SIFT descriptor, such a feature is invariant to changes in light intensity [35].

3. C-SIFT: The O1 and O2 components of the opponent color space (see [35]) contain intensity information. To obtain a descriptor which is scale-invariant with respect to light intensity changes, the 384-D C-SIFT descriptor was proposed [35].

Each of these feature has been encoded using a BoW approach with 1000 codewords.

Data-Driven:
Following [49], the output for the last convolutional layer has been taken as the image feature representation. As a result each image is described by a 4096-D feature vector.

When jointly considered, the resulting feature vector lies in a 19770-D feature space.

4.1.2. Kernels

When kernel-ELMs are used, their performances are evaluated using four different kernels, namely the: (i) linear kernel; (ii) cosine kernel; (iii) exponential χ^2 kernel; (iv) radial basis function kernel (with free parameter set to 1);

4.1.3. Datasets

To validate the proposed method four publicly available benchmark datasets have been considered. The PFID [65], UNICT-FD889 [30], the UECFood100 [27] and the Food-101 [19] datasets have been selected because they provide different food recognition challenges:

1. The PFID dataset has images acquired under different lighting conditions and from different viewing angles. Therefore, it is useful to understand if the proposed method is robust to such challenges.
2. The UNICT-FD889 dataset has images of 889 different real food plates acquired by mobile devices in uncontrolled scenarios. Hence, results on this dataset provide an estimate on how well an algorithm scales to a real scenario.
3. The UECFood100 dataset contains about 14000 images, corresponding to 100 different food categories.
4. Similarly, the Food-101 dataset has images of 101 different foods. However, in such case 1000 images for each category are available. Due to the large number of images, these two datasets are well suited to evaluate the learning performance of the proposed approach.

More details regarding each dataset are given in the following.

4.1.4. Experimental Scenarios

Three main different scenarios have been selected to analyze the performance of the proposed approach. These are the followings:

1. To see how single features perform on the food recognition task, performances obtained by separately exploiting each considered type of feature have been computed.
2. To demonstrate the benefits of kernel ELM over standard ELM, results will be also given for the case when kernel is not used. Under such a scenario, the number of hidden neurons has been set to $L = 1000$.
3. To demonstrate the benefits of the proposed kernel-ELM committee supervisor, the achieved performance are compared to those obtained by using common fusion schemes: (a) *low*-level consists in feature concatenation; (b) *mid*-level, where the kernels computed for different features are combined. Results have been obtained by kernel averaging [67], kernel product [67] and by exploiting the Sparse and Non-Sparse version of Multiple Kernel ELMs (MKELMs) [68]; (c) *high*-level, where committee members outputs are fused using the weights learned by means of score averaging, Lasso and Logistic Regression (our method belongs to such a category).

In all the following results, the performances shown for both the *mid* and *high*-levels fusion schemes (SELC included) have been computed using the $\chi^2 - exp$ kernel for every feature type. Notice that, the kernels could have been separately selected for each dataset to obtain better recognition performance. However, to provide a more general framework, the choice of the kernel has been kept fixed.

4.2. PFID Dataset

The Pittsburgh Fast-food Image Dataset[1] (PFID) was the first dataset build exclusively for food recognition [65]. The PFID contains data of three instances of 61 different food items which were purchased on different dates from the same restaurant or at different branches of the same fast food chain. Each instance has 6 images collected under different lighting conditions, with different background, and sensed from different viewing angles (see Figure 4). As a result, the whole dataset contains 1089 images of 61 different classes of food.

Following the protocol in [18, 17], performance evaluations have also been conducted by re-organizing the 61 PFID food categories into 7 major classes: Sandwiches, Salads&Sides, Chicken, Breads&Pastries, Donuts, Bagels, and Tacos. In both the cases, 3-fold cross-validation has been conducted using 12 images from two instances of each original class for training, and the 6 remaining images of the third instance of each original class for testing [18, 17].

4.2.1. Performance Analysis

In Figure 5a the performances achieved by the single features using different kernels are shown. Let consider the case when no kernel is used. Under such scenario, results demonstrate that PRIcoLBP features are the best performing ones with an accuracy of 30.62%. In general, color histograms and texture features perform better than the CNN ones. Despite this, when the Cosine kernel is

[1] Available at http://pfid.intel-research.net/

Figure 4: 15 randomly selected samples from the PFID dataset. Each column corresponds to a different type of food (i.e., a different class). Rows show three different instances. In the PFID dataset some food classes (i.e., items on the 5th, 6th and 7th columns) are very similar to each other.

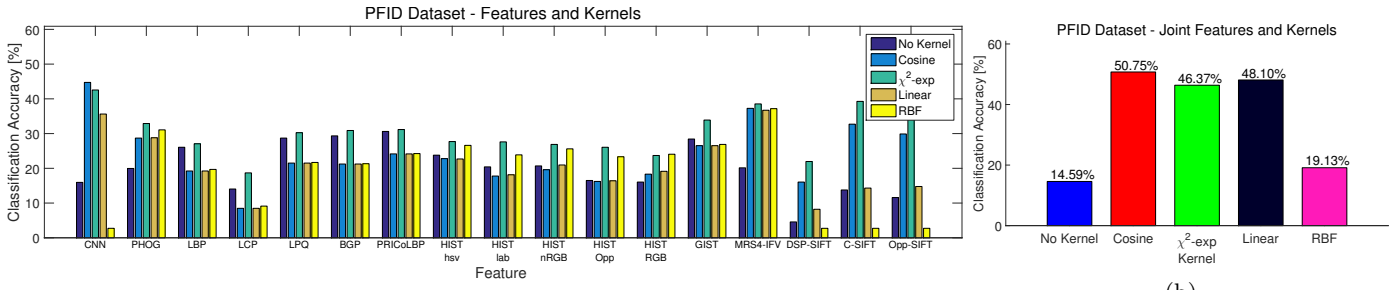


Figure 5: Accuracy performances obtained by the proposed method on PFID dataset. In (a) performances achieved by using single features with different kernels are given. In (b) performances achieved by considering the joint feature space and different kernels are shown.

used, the opposite occurs and CNN features yield to the best performance with an accuracy improvement of about 30%.

In Figure 5b the performances achieved by considering the joint feature space (i.e., low level fusion scheme) and using different kernels are shown. The depicted results show that, when no kernel is used, the accuracy is of 14.59%, which is even lower than the one achieved by using some features alone. If the Cosine kernel is adopted, performance reaches an accuracy of 50.75%, which is slightly better than the single performance achieved by CNN features only. This shows that, as for the single feature scenario, if a kernel is used performance will improve. However, dependently on the adopted kernel, the performances of the joint feature approaches are very close or even worse than the ones achieved by using the best single feature. This highlights the fact that, if jointly considered, there is no guarantee that adding more features to tackle the problem of food recognition improves the performance.

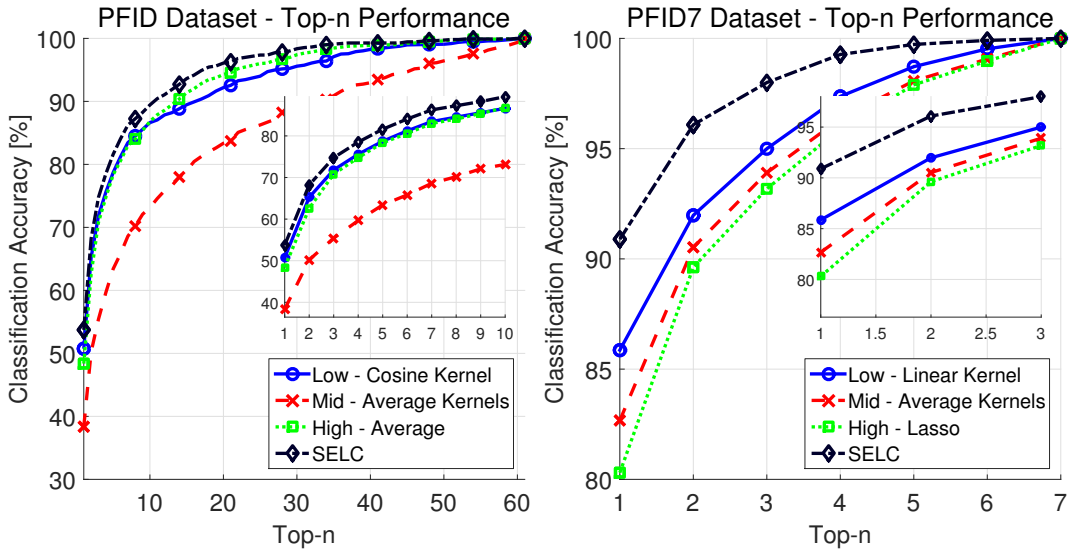
To show the benefits of the proposed committee-based method, in Figure 6 the achieved *top–n* performances are compared to the ones obtained by the low/mid/high level fusion schemes having the best accuracy results on the considered dataset. Results have been computed considering both the 61 categories and the 7 major classes.

Results computed considering all the 61 categories (see Figure 6a) show that by using the proposed SELC approach the accuracy performance is of 53.73%. Thus, it SELC improves the best results obtained by considering the joint features and cosine kernel by about 3%. Such a gap increases more with larger values of n. A similar difference in performance is shown with respect to the average high level fusion scheme. Kernel averaging yields to a significant decrease in the performance. Indeed, in such a case the accuracy is of 38.36%, only.

When the 7 major classes are considered only, a similar behavior is achieved (see Figure 6b). Considering the accuracy reached by using the linear (85.86%) and the cosine (85.22%) kernels for the low level fusion scenario, the accuracy improves by 5.03% and by 5.67% respectively. When the best performing mid and high level fusion schemes used for comparisons are considered, SELC improves the corresponding accuracies by more than 7% and 10%, respectively.

To better analyze the performance of the proposed method, the confusion matrices shown in Figure 7 have been computed. The lighter the diagonal line, the more effective the approach, because it has a higher probability of classifying food of a given category as itself. When the 7 major classes are considered (Figure 7b), the correct classification percentages are shown together with the class labels.

For both the scenarios, the diagonal elements show high probabilities, thus reflecting the capacities of the proposed approach in correctly classifying most of the plates. More interestingly, Figure 7b have a light vertical band showing that in a large number of cases a food is assigned to the Sandwich class. The motivation behind such be-



(a) (b)

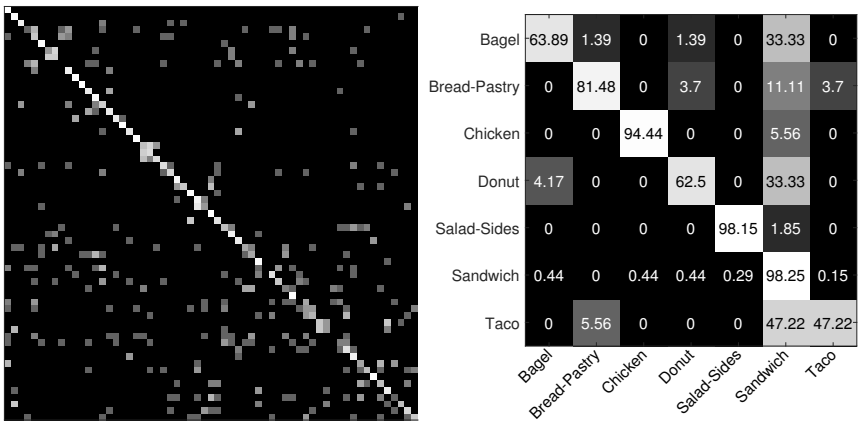
Figure 6: *Top–n* Performances on the PFID dataset using the proposed approach are compared to the results achieved considering the best performing low/mid/high level fusion schemes. In (a) results are computed considering all the 61 classes. In (b) results are computed considering only the 7 major classes. The inside pictures show the performance on a reduced range of *Top–n*.

havior is that in the 7 major classes dataset, the majority of the samples belong to such a class. Hence the dataset is not well balanced among all the classes as it was in the case all the 61 categories were considered. Therefore, when the algorithm is trained with many Sandwich samples and very few Taco samples that shares similar characteristic as those, the classifier is not able to find a good decision boundary separating the two classes.

In Figure 8 the performances achieved by the proposed method are shown for 6 query images (see caption for additional details). The depicted results demonstrate that proposed approach is able to well capture the global appearance of the images and it also has the capacity to reliably find the true match under challenging conditions. When the query image is not correctly classified, or the considers cases are very challenging, the resulting scores are very close to each other, thus meaning there is uncertainty in the given answer.

4.2.2. State-of-the-art Comparisons

In Figure 9, the performance of the proposed SELC approach is compared to the state-of-the-art ones. The comparison is given with respect to the PFID dataset with all the 61 classes. Results demonstrate that the proposed approach improves the state-of-the-art performance of CTX-MKL [24] by more than 5.2% and outperforms recent approaches like Class-BoT [17] and OM [18] by more than 20%.



(a) (b)

Figure 7: Confusion matrices computed for both the standard PFID dataset (a) and its 7 major classes (b). The lighter the diagonal, the more effective the approach. In (b), the class labels and the correct classification percentages are shown.

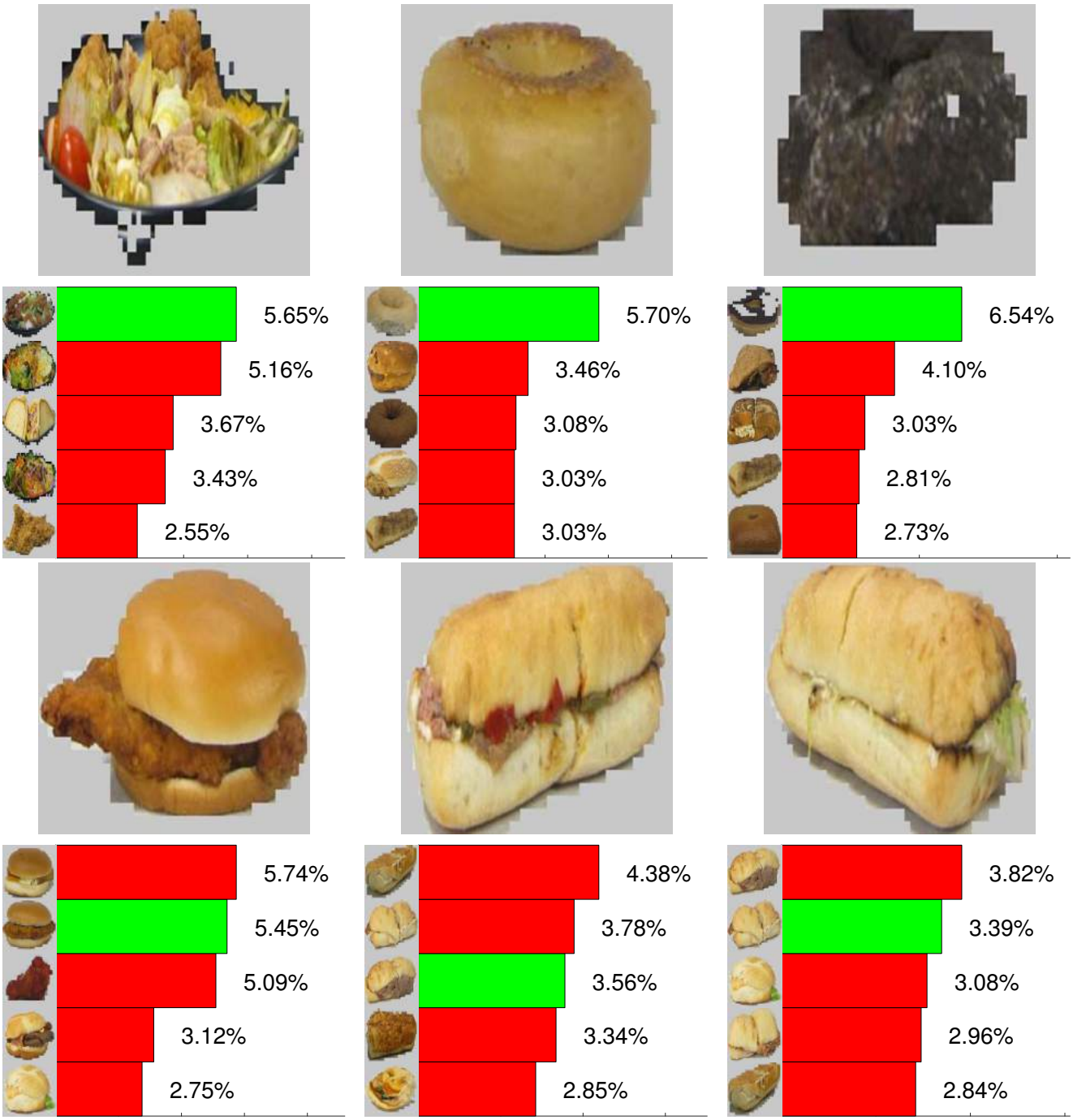


Figure 8: Performances achieved on the PFID datasets by the proposed method are shown for 6 query images (organized in two rows). The bar histograms show the score (in percentage) of the proposed approach for the true match (in green) and for the remaining top 4 matches (in red). On the y-axis of each bar histogram a randomly selected training image corresponding to the food class is depicted. *(Best viewed in color)*

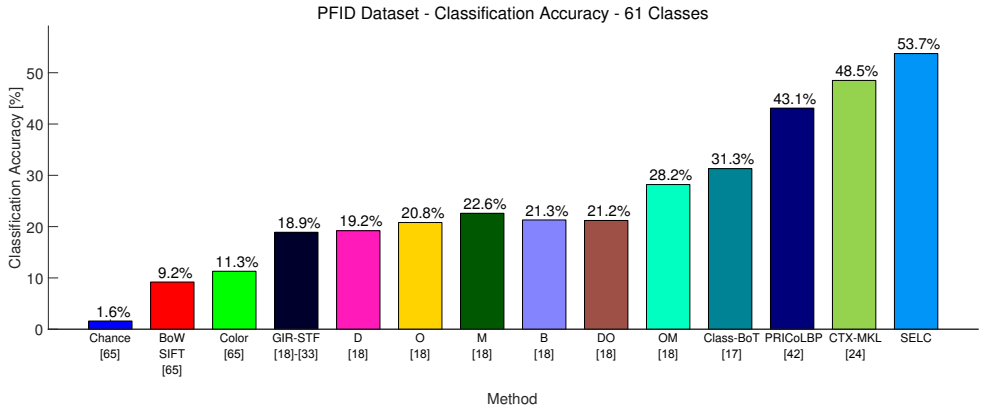


Figure 9: Comparisons with state-of-the-art methods computed considering all the 61 categories in the PFID dataset. Results are shown as classification accuracy.

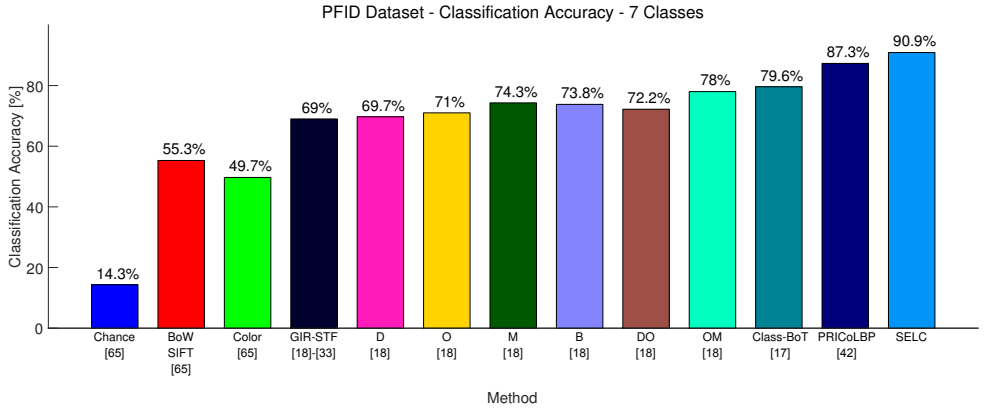


Figure 10: Comparisons with state-of-the-art methods on the PFID dataset considering only the 7 major classes. Results are shown as classification accuracy.

Table 1: Classification performances achieved by state-of-the-art methods the 7 major classes of the PFID dataset. Since the number of images belonging to the different classes is not balanced, for each class the per-class accuracy is given together with the corresponding number of images. Best results for each class are highlighted in boldface font.

Images per class	Sandwich	Salad & Sides	Bagel	Donut	Chicken	Taco	Bread & Pastry
On each test run	228	36	24	24	24	12	18
Per class accuracy [%] (Number of Images)	Sandwich	Salad & Sides	Bagel	Donut	Chicken	Taco	Bread & Pastry
Color [65]	69.0 (157.3)	16.0 (5.8)	13.0 (3.1)	0.0 (0)	49.0 (11.8)	39.0 (4.7)	8.0 (1.4)
BoW SIFT [65]	75.0 (171)	45.0 (16.2)	15.0 (3.6)	18.0 (4.3)	36.0 (8.6)	24.0 (2.9)	3.0 (0.5)
GIR-STF [18]-[33]	79.0 (180.1)	79.0 (28.4)	33.0 (7.9)	14.0 (3.4)	73.0 (17.5)	40.0 (4.8)	47.0 (8.5)
OM [18]	86.0 (196.1)	93.0 (33.5)	40.0 (9.6)	17.0 (4.1)	82.0 (19.7)	65.0 (7.8)	67.0 (12.1)
Class-Based BoT [17]	87.6 (199.7)	84.3 (30.3)	**70.8 (17.0)**	43.1 (10.3)	66.7 (16.0)	**69.4 (8.3)**	53.7 (9.7)
SELC	**98.25 (224.1)**	**98.15 (35.3)**	63.89 (15.3)	**62.50 (15.0)**	**94.44 (22.7)**	47.22 (5.7)	**81.48 (14.7)**

In Table 1 and Figure 10 the accuracy performance comparison between the proposed approach and state-of-the-art ones on the 7 major classes of the PFID dataset are given. In addition, following in [17], to better understand the results in Table 1, the number of images belonging to the different classes are reported together with the per-class accuracy.

Results depicted in Figure 10 show that the accuracy performance of SELC (90.9%) outperforms all the considered algorithms, with PRI-CoLBP [42] being the closest ones with an accuracy of 87.3%.

Results in Table 1 show that the main source of errors is the Taco food category. As already discussed, this is due to the imbalanced conditions of the dataset. Despite this, the proposed SELC approach performs better than existing ones in classifying 5 out of 7 categories. In the two remaining case, Class-based BoT [17] performs better. Notice that Class-based BoT requires that an encoding is computed for each different class of food. While such an approach could have been exploited in our work as well, we decided not to use it to limit the computational requirements.

4.3. UNICT-FD889 Dataset

The UNICT-FD889 Dataset[2] has been recently introduced in [30]. The UNICT-FD889 dataset is the one that has the largest number of different classes to recognize. It comes with 3583 images related to 889 distinct food categories belonging to different nationalities (e.g., Italian, English, Thai, Indian, Japanese, etc.). Images have been collected in a real and uncontrolled scenario (e.g., different backgrounds and light environmental conditions) by means of smartphones. Hence, the UNICT-FD889 dataset is a collection of food images acquired by users in real cases of meals. Each food belonging to a particular class has been acquired multiple times (four on average) to ensure geometric and photometric variabilities (see Figure 11 for a few examples).

To provide a fair comparison with existing methods, the following results have been computed by averaging the performance on the same three splits adopted in [30].

4.3.1. Performance Analysis

In Figure 12a the performances achieved by the single features using different kernels are shown. Differently from the results of single feature on the PFID dataset, in such a case the best classification accuracies are achieved using color histogram features. In particular, when no kernel is used, an accuracy of 53.44% is achieved using color histogram features extracted from the normalized RGB color space. CNN features do not perform well. Their classification accuracy obtained without using a kernel is of 14.03% only. However, when kernels are introduced, their performance increases significantly and a classification accuracy of 66.30% is reached using the Cosine kernel. Thus, as shown for the PFID dataset, results demonstrate that the choice of an appropriate kernel may strongly improve the recognition performance.

In Figure 12b the performances achieved by considering the joint feature space (i.e., low level fusion scheme) and using different kernels are shown. The depicted results show that, when no kernel is used on the joint feature space the obtained accuracy (5.15%) is much less than the one obtained by using color histogram features only. While the usage of a kernel drastically improves the accuracy performance, these are still on the same line as the ones achieved by color histograms. Indeed, using the $\chi^2 - exp$ kernel an accuracy of 68.95% is reached (which is very similar to the one obtained using the same kernel on normalized RGB histogram features, i.e. 72.16%).

In Figure 13 the *top–n* performance of the proposed SELC approach is compared to the ones achieved by the best performing low/mid/high level fusion schemes. Under such scenario, results show that significant benefits can be obtained by using the proposed high level fusion committee-based approach.

In particular, let us consider the results obtained by the low level fusion approach using the $\chi^2 - exp$ kernel. When $n = 1$, a classification accuracy of 68.95% is obtained using the joint feature space. SELC reaches a classification accuracy of 88.85%, thus yielding to a 20% performance improvement. Such a gap reduces to 1%, only when n=300. Similarly, when compared to the mid and high level fusion schemes, results show that a significant improvement is achieved. Specifically, an accuracy improvement of about 17% and 3.5% is obtained with respect to the Product Kernel [67] and the Average fusion schemes, respectively.

In Figure 14 the performances achieved by the proposed method are shown for 6 query images (organized in two rows). The reported cases show that the proposed approach is able to well capture both the global appearance of the images and the little details that differentiate two very similar classes (e.g., see the first query on the second row).

4.3.2. State-of-the-art Comparisons

In Figure 15 the performance of the proposed SELC approach is compared to the state-of-the-art ones given in [30, 69]. The depicted results demonstrate that the proposed method strongly outperforms the existing ones by improving the best previous performance by more than 28%. In particular, the PRI-CoBP [42] approach that has similar performance to SELC on the PFID dataset, is getting the second worst accuracy on the UNICT-FD889 dataset.

4.4. UECFood100 Dataset

The UECFood100 Dataset [3] is one of the largest food recognition datasets [27]. This dataset contains approxi-

[2] Available at http://iplab.dmi.unict.it/UNICT-FD889

[3] Available at http://foodcam.mobi/dataset100.html

Figure 11: 15 randomly selected samples from the UNICT-FD889 dataset. Columns correspond to a different type of food (i.e., to a different class). Rows show the appearance variations between samples belonging to the same class.

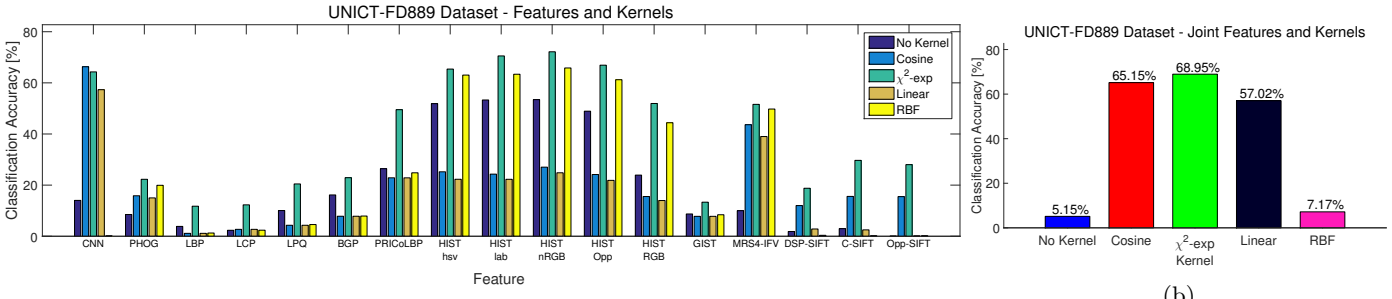


Figure 12: Accuracy performances on the UNICT-FD889 dataset achieved by: (a) using single features with different kernels and (b) considering the joint feature space and different kernels.

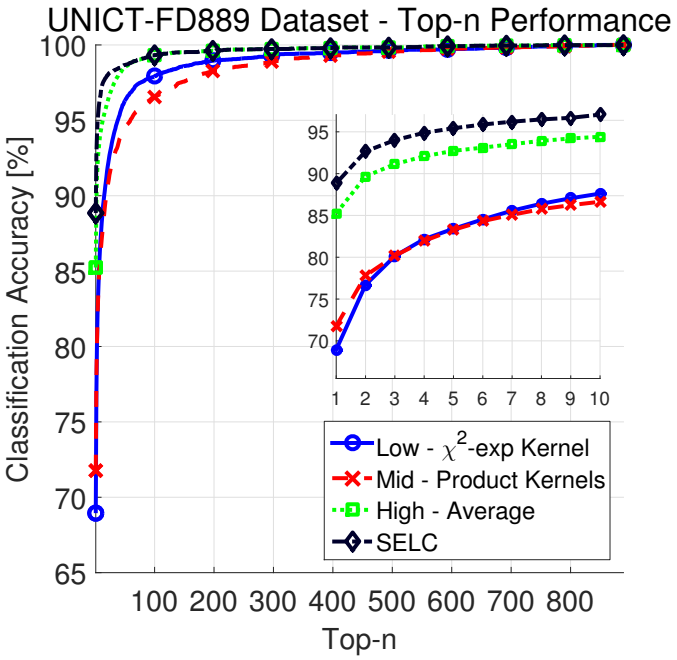


Figure 13: Performances on the UNICT-FD889 dataset using the proposed approach are compared to the results achieved considering the best performing low/mid/high level fusion schemes. Performance are given using the *top–n* criterion.

mately 14000 real-world food images belonging to 100 different categories. The UECFood100 dataset was built to implement a practical food recognition system [70] which was intend to be used in Japan. Because of this, it was collected in such a way that multiple food items were present in a single image, thus with the objective to perform both the detection and the recognition tasks. However, since the proposed system is designed to focus only on the recognition task, the given ground truth bounding boxes have been used to obtain a dataset of images containing single food items only (see Figure 16). Despite this, the same protocol in [27] has been followed to fairly compare the obtained performance with existing methods.

4.4.1. Performance Analysis

In Figure 17, the accuracy performance on the UEC-Food100 dataset are shown for single features (Figure 17a) and joint features (Figure 17b) both with different kernels.

Results in Figure 17a show the performances achieved by single features exploiting different kernels. When no kernel is used to model each single feature space, local features are well discriminating between the 100 categories. Data-driven features (i.e., CNN) yield to the best performance with a classification accuracy of about 53.54%. Color histogram features extracted from the normalized RGB color space achieve the lowest classification accuracy (i.e., 0.96%). As for the other two considered datasets, when kernels are used performances strongly improve. In particular, the CNN feature performance increases to 66.99% and to 74.30% when the cosine and the $\chi^2 - exp$ kernels are adopted, respectively. Using the same kernels, MRS4 encoded features yield to a classification accuracy of 34.21%

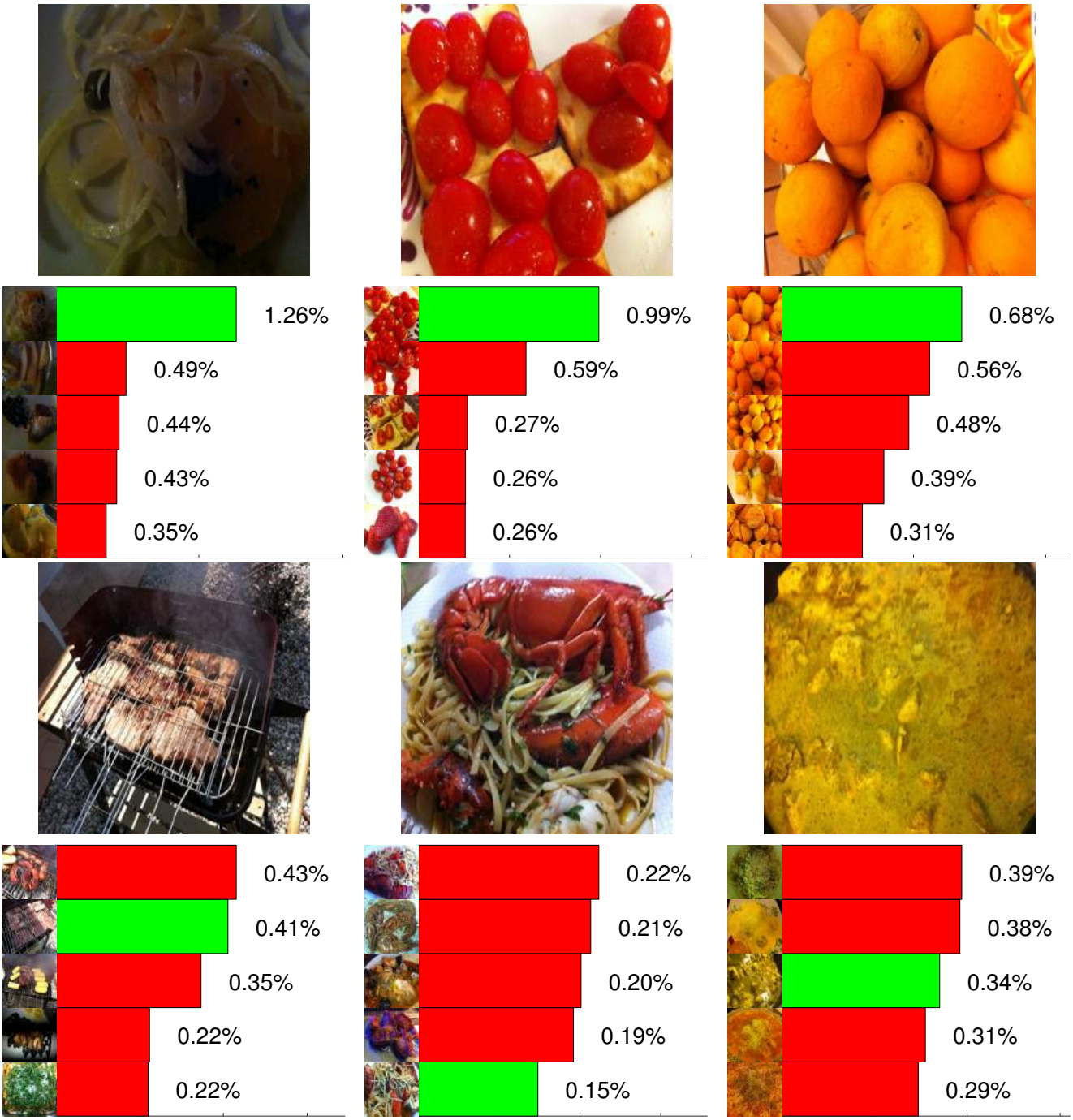


Figure 14: Performances achieved by the proposed method on the UNICT-FD889 dataset are shown for 6 query images (organized in two rows). At the bottom of each of those, the bar histograms show the score (in percentage) of the proposed approach for the true match (in green) and for the remaining top 4 matches (in red). On the y-axis of each bar histogram a randomly selected training image corresponding to the food class is depicted. *(Best viewed in color)*

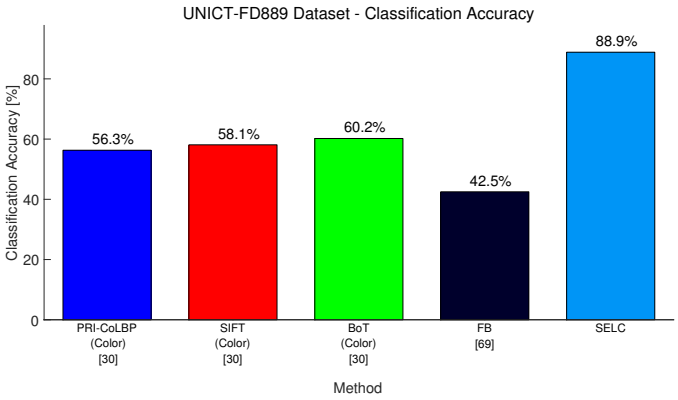


Figure 15: Comparisons with state-of-the-art methods on the UNICT-FD889 dataset. Results are shown in terms of classification accuracy.

and 47.96%. More interestingly, when the linear and the RBF kernels are used the CNN feature performance significantly drop down to 2.11% and 2.78%.

Performance achieved by considering the joint feature space and using different kernels are depicted in Figure 17b. Results show that an accuracy of 58.63% is reached when no kernel is used to model the joint feature space. Performance improves by 16.15% if the cosine kernel is adopted. Such an improvement is more significant when the exponential χ^2 kernel is used. Indeed, in that particular case performance increases by more than 20%.

In Figure 18 the *top–n* performance of the proposed SELC approach is compared to the ones achieved by the existing low/mid/high level fusion schemes that have the highest accuracy on the considered UECFood100 dataset. Results show that, for low values of n, the performance achieved by SELC is very close to the ones obtained by modeling the joint features space with the $\chi^2 - exp$ kernel. In particular, when n=1, the gap between the two approaches is of about 3%. Such a difference remains stable and reduces to 0.5% only when n=32. The mid and high level fusion schemes used for comparison, namely Product Kernels [67] and Lasso, have an accuracy of 46.71% and 61.80%, respectively. Thus, such methods are strongly outperformed by the proposed fusion scheme.

Qualitative performances achieved by the proposed SELC approach on the UECFood100 dataset are shown in Figure 19. Results are shown for 6 query images (organized in two rows). The depicted images demonstrate that the proposed approach is able to model the appearance of the 100 categories and can well generalize to even very challenging test samples (e.g., first row, 3rd query).

4.4.2. State-of-the-art Comparisons

In Figure 20 the performance of the proposed SELC approach is compared to the state-of-the-art ones both in terms of classification accuracy and by using the *top–n* criterion. Results are compared to the ones provided in [27] and [25]. Methods like Circle, JSEG, DCR, DPM, and Whole (all from [27]), use a detector to identify the location of the food, while GTBB [27] and PMTS [25] uses the same ground truth as SELC. The reported results demonstrate that state-of-the-art performance are significantly improved from 60.2% (PMTS [25]) to 84.3%. Such a difference reduces little when n increases. When n=5 the proposed method is the only one that achieves a classification accuracy of more than 95%.

To conclude, while results of other approaches that use the detector are not directly comparable, we can hypothesize that since GTBB uses the same features and learning algorithm as those to perform the classification, it is plausible to assume that SELC outperforms such methods as well if the same detector is used.

4.5. Food-101 Dataset

The Food-101 Dataset[4] is the largest food recognition dataset [19]. It has been collected by downloading images from foodspotting.com The top 101 most popular and consistently named dishes were selected. Then, for each category 750 training and 250 test images were collected and manually cleaned. On purpose, the intense colors and sometimes wrong labels included in the training images were not cleaned. As a result the dataset contains 101'000 real-world food images (see Figure 21).

The same splits introduced in [19] have been used to compute all the following results.

4.5.1. Performance Analysis

Results in Figure 22 show the accuracy performances on the Food-101 dataset of single features and joint features (i.e., low-level fusion scheme) with different kernels.

Results, in Figure 22a show that the performances of single features are similar to the ones obtained for the other datasets with CNN ones largely dominating the others. Such features achieve a 49.54% accuracy when the $\chi^2 - exp$ kernel is used. The second runner up is the MRS4-IFV feature with an accuracy that is less than half of the aforementioned one (i.e., 24.80%). Regardless the considered kernel, color histograms, LBP and LCP barely achieve an accuracy higher than 15%. Thus, these are the worst performing ones for such a dataset.

When all such features are jointly considered (see Figure 22b) the performances improve and an accuracy of 52.90% is obtained using the $\chi^2 - exp$ kernel. In all the other cases, the performances considerably reduce, even with respect to single features. In particular, when the RBF and linear kernels are used the accuracy never gets higher than 20%. Thus, showing that CNN and MRS4-IFV features used alone yield to better performance.

In Figure 23, the *top–n* performance achieved by SELC is compared to the ones obtained by using different fusion schemes. Specifically, the results are shown for the

[4] Available at http://www.vision.ee.ethz.ch/datasets/food-101

Figure 16: 15 randomly selected samples from the UECFood100 dataset. Images of single food items were obtained by using the given ground truth bounding boxes. Columns correspond to a different type of food (i.e., to a different class). Rows show the appearance variations between samples belonging to the same class.

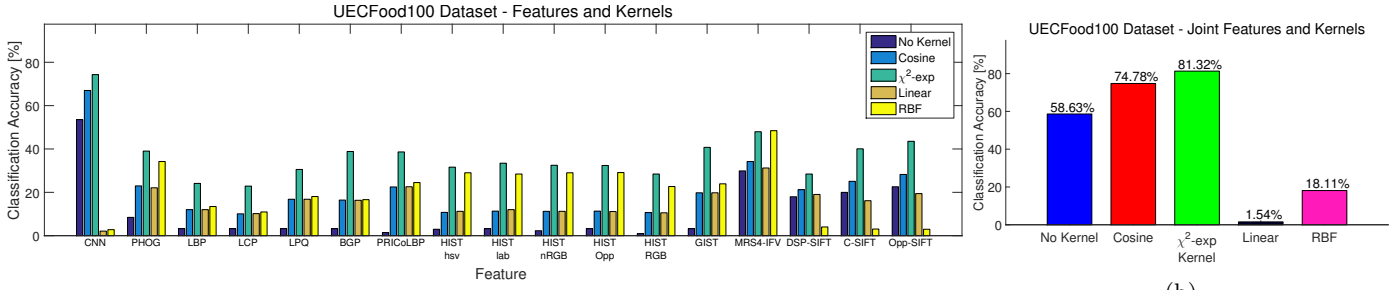


Figure 17: Accuracy performances on the UECFood100 dataset obtained by single features and joint ones, both with different kernels. In (a) the results of single features with different kernels are depicted. In (b) the accuracy performances obtained by considering the joint feature space and different kernels are shown.

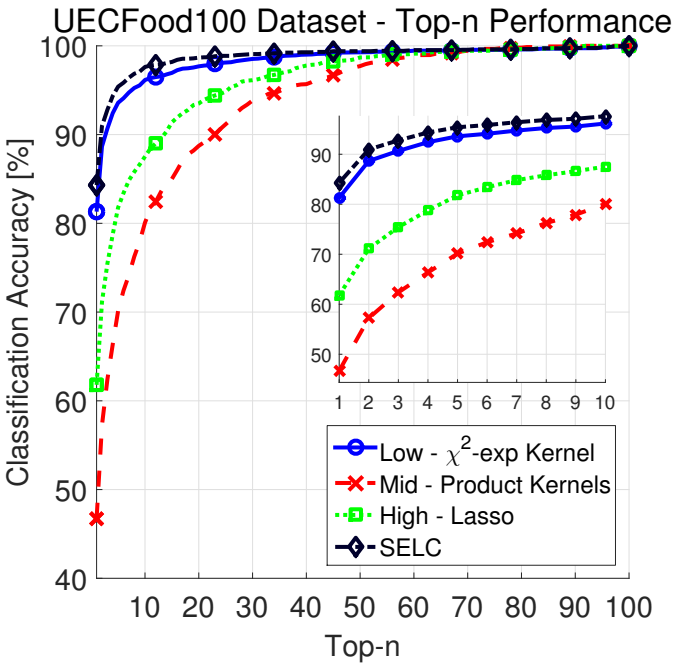


Figure 18: Results on the UECFood100 dataset using the proposed approach are compared to the best performing low/mid/high level fusion schemes. Performance are shown using the *top–n* criterion.

low/mid/high level fusion schemes that have the highest accuracy on this dataset. Results show that the proposed approach obtains the highest accuracy (i.e., 55.89%) and it also yields to the best ranking with respect to other methods. More interestingly, the best high fusion scheme, i.e., Lasso, has the worst performance both in terms of accuracy and ranking.

4.5.2. State-of-the-art Comparisons

In Table 2, the accuracy performance of SELC is compared to the ones obtained by existing methods on the Food-101 dataset. Results show that, with an accuracy of 56.40%, the best performing approach on such dataset is achieved by employing a convolutional neural network which has been trained using the AlexNet architecture. Such results reflects the performance achieved by our single CNN features which achieves an accuracy of 49.54%. However, notice that in such a case the adopted OverFeat [50] has been trained on natural images and not on this dataset specific samples. Despite this, the performance of SELC is only 0.51% less than the best existing one. All the other approaches are significantly outperformed. In particular, the performance of the very recent RFDC [19] work is improved by more than 5%.

4.6. Computational Performance

To show the computational performance of the proposed approach, we have computed the results in Table 3 and Figure 24.

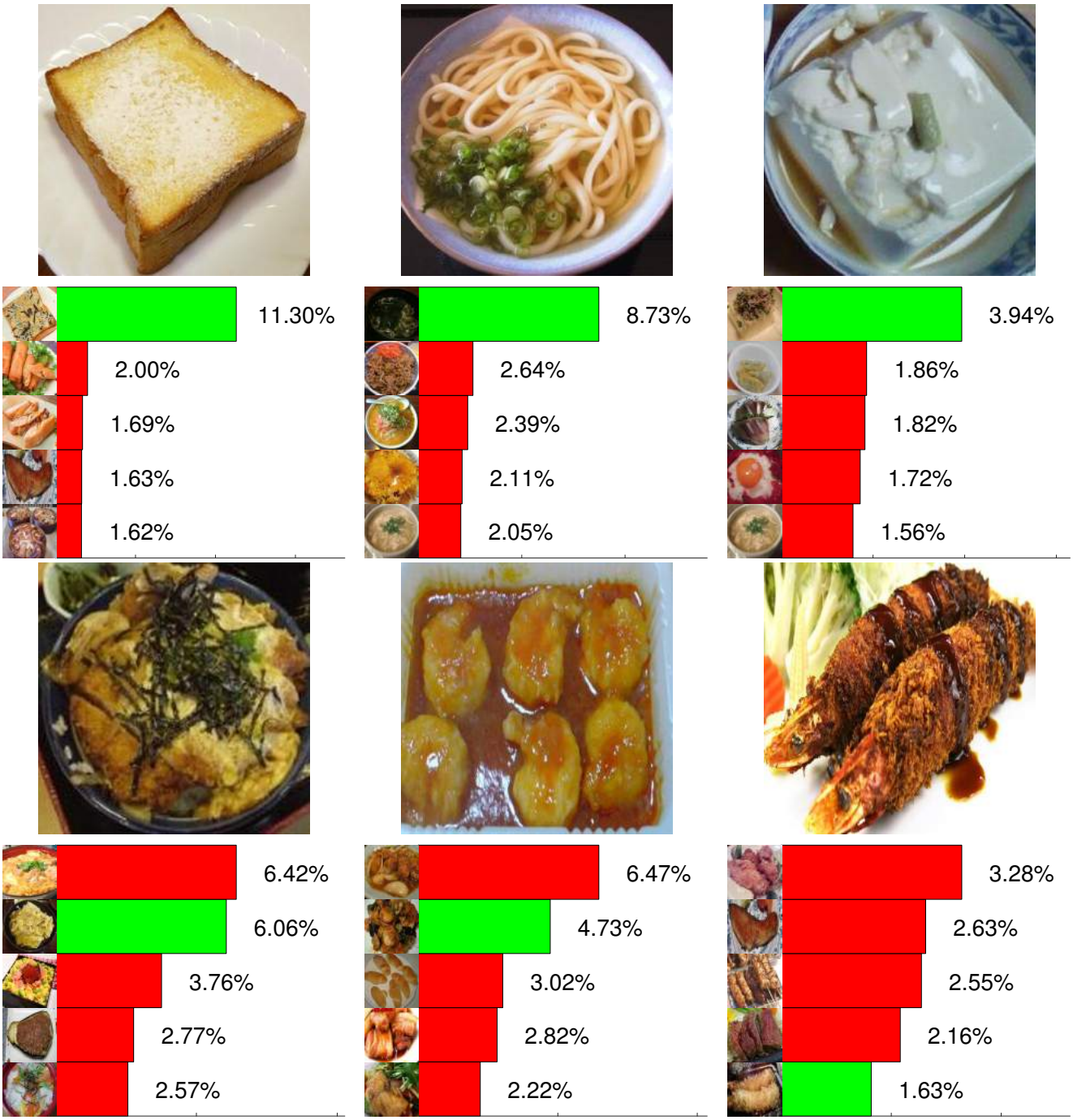


Figure 19: Performances achieved by the proposed method on the UECFood100 dataset are shown for 6 query images (organized in two rows). At the bottom of each of those, the bar histograms show the score (in percentage) of the proposed approach for the true match (in green) and for the remaining top 4 matches (in red). On the y-axis of each bar histogram a randomly selected training image corresponding to the food class is depicted. *(Best viewed in color)*

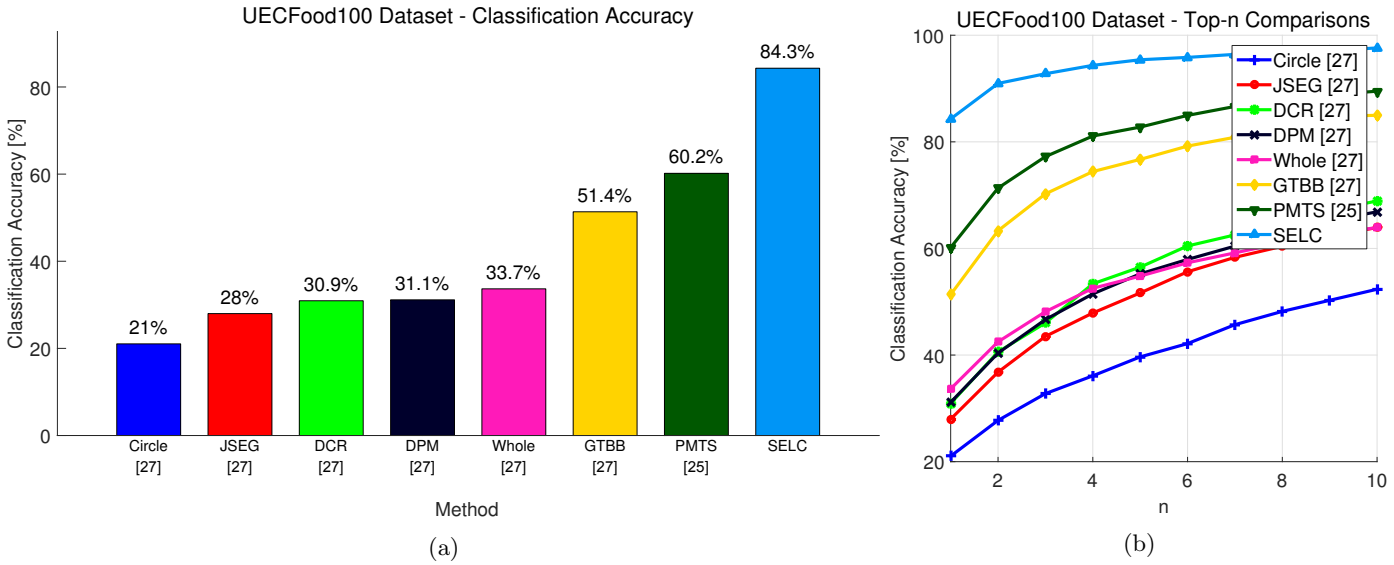


Figure 20: Comparisons with state-of-the-art methods on the UECFood100 dataset. Results are shown in terms of classification accuracy.

Figure 21: 15 randomly selected samples from the Food-101 dataset. Columns correspond to a different type of food (i.e., to a different class). Rows show the appearance variations between samples belonging to the same class.

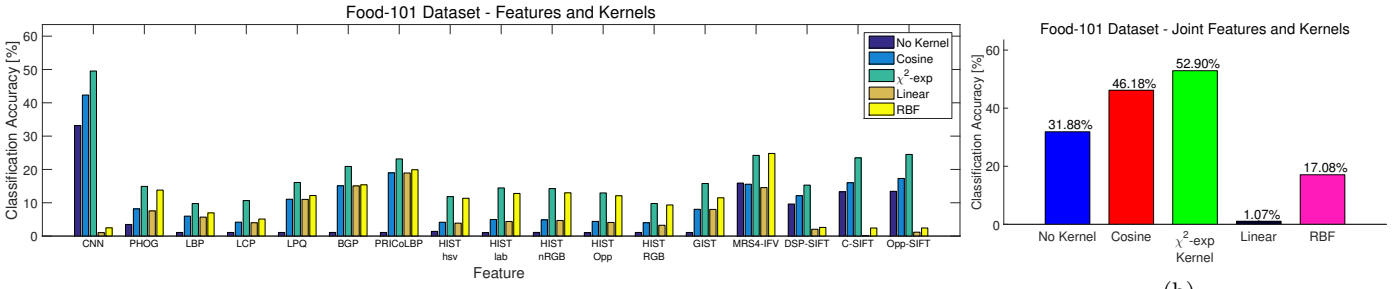
Results in Table 3 report on the classification accuracy and the processing times required to run a non optimized MATLAB implementation of the proposed approach on a Intel Xeon E5-v2660 machine equipped with 256GB of RAM. The results are shown for all the datasets and for the different fusion schemes. The processing reported for PFID training has been average over all the three trials.

Results demonstrate that the low level fusion schemes are less demanding in terms of computational times. In particular, when no kernel is exploited, only 1 second is required for training. When kernels are introduced the processing time increases and reaches a maximum of about 16 seconds ($\chi^2 - exp$ kernel).

Mid and high level fusion schemes have similar performances both in terms of accuracy as well as in processing times. Specifically, it should be noticed that the SELC approach requires more processing time than Average and Product Kernels [67], and the Average high fusion methods only. These only require a sum or a product over kernels or committee answers. Thus, the proposed fusion scheme not only produces the best accuracy but it is also competitive in terms of computational performance.

Finally, it is a matter of fact that nowadays, food recognition algorithms are very attractive for mobile devices. As regards a possible deployment of the SELC approach on such devices, we can state the following. As shown in Table 3, the kernel computation is computationally demanding, especially if the training set is very large. On the contrary, the classification and fusion operations can be performed in fractions of a second. Despite this, the proposed approach uses many different features which requires more than a couple of second to be extracted.

To verify if all the proposed features are necessary for a correct classification we have performed the following experiment on the Food-101 dataset. We have run the proposed approach by subsequently eliminating a single committee member output from the recognition phase (i.e., the fusion weight assigned to the feature has been zeroed). The process is conduced by eliminating the features following an ordering given by the sorted (ascending) supervisor weights. Thus, the feature assigned with the lowest weight is eliminated first. Results in Figure 24 show that by eliminating 12 features out of 17, the performance decreases by about 1%. If two more features are removed, such a reduction is more evident and the accuracy goes below 50%. Results demonstrate that only few features



(a) (b)

Figure 22: Feature and kernel performance analysis on the Food-101 dataset. In (a) classification accuracies achieved by using single features with different kernels are shown. In (b) the accuracy performances obtained by considering the joint feature space (low-level fusion scheme) and different kernels are depicted.

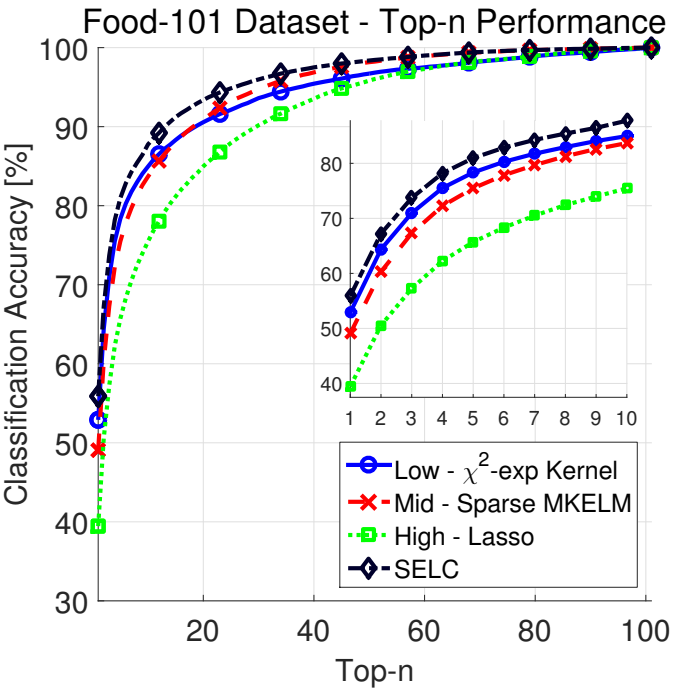


Figure 23: *Top–n* performances on the Food-101 dataset. The results achieved by the proposed fusion scheme are compared to the ones obtained by using the best low/mid/high ones.

Table 2: Accuracy performance achieved by state-of-the-art methods on the Food-101 dataset. Best results is highlighted in boldface font.

Method	Accuracy [%]
HoG [19]	8.85
SURF BoW-1024 [19]	33.47
SURF BoW-1024 + Color BoW-256 [19]	38.83
SURF IFV-64 [19]	44.79
Color IFV-64 [19]	14.24
SURF IFV-64 + Color Bow-64 [19]	49.40
BoW [19]	28.51
IFV [19]	38.88
AlexNet-CNN [19]	**56.40**
RF [19]	37.72
RCF [19]	28.46
MLDS [19]	42.63
RFDC [19]	50.76
SELC	55.89

are very discriminative and the supervisor is able to capture such information. Thus, to reduce the computational complexity, features that are assigned a low fusion weight can be excluded from the test phase without significant performance loss.

To summarize, the proposed method has significant complexity both in terms of memory and time for the computation of the considered features. However, this additional computational burden is justified by the better performances that are obtained with respect to other fusion schemes and approaches in the literature.

4.7. Discussion

On the basis of the results obtained for the four considered datasets we can state the following considerations.

Features:

Results on single features have shown a large inconsistency across the different datasets. For instance, color histogram features yield to excellent performance on the UNICT-FD889 dataset while they perform very poorly on the UECFood100 and Food-101. The opposite occurs when

Table 3: Computational and accuracy performances [%] of the proposed method. The PFID training time performances [s] are shown in the 7th column. The given values have been averaged over the three considered trials. The last column shows the time required to classify a single image (averaged over all the datasets). Best results are highlighted in boldface font.

	PFID	PFID7	UNICT-FD889	UECFood100	Food-101	PFID Training Time [s]	Average Test Time [s]
No Kernel	14.59	33.22	5.15	58.63	31.88	**0.99**	**0.01**
Cosine	50.75	85.22	65.15	74.78	46.18	2.97	0.08
$\chi^2 - exp$	46.37	81.22	68.95	81.32	52.90	15.48	0.29
Linear	48.10	85.86	57.02	1.54	1.07	2.69	0.07
RBF	19.13	18.33	7.17	18.11	17.08	2.91	0.07
Average Kernels [67]	38.36	82.67	59.74	11.06	37.26	68.75	1.67
Product Kernels [67]	36.99	78.03	71.77	46.71	38.61	68.07	1.64
Sparse MKELM [68]	35.63	80.31	57.33	10.96	49.07	74.82	1.71
Non-Sparse MKELM [68]	35.81	80.49	57.44	12.25	39.88	74.65	1.73
Average	48.38	78.76	85.24	52.40	31.09	69.11	1.74
Lasso	35.54	80.31	57.29	61.80	39.46	72.86	1.76
Logistic-Regression	38.47	68.94	55.86	51.71	32.35	102.40	1.77
SELC	**53.73**	**90.89**	**88.85**	**84.31**	**55.89**	70.57	1.76

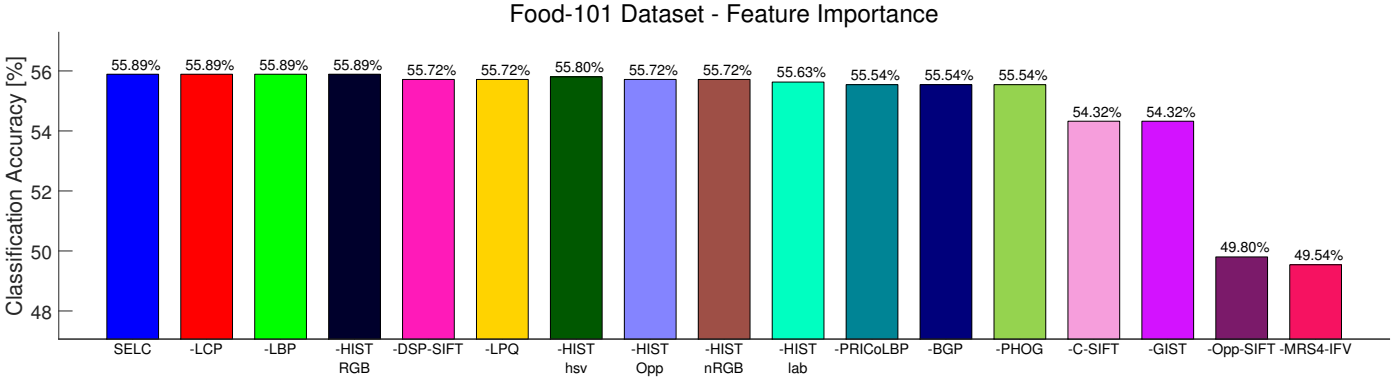


Figure 24: Accuracy performance achieved by the proposed approach with subsequent elimination of a committtee member. First bar shows the SELC approach perormance. Following ones show the results obtained by eliminating a particular feature and all the other ones appearing on its left, from the test phase. The last column shows the accuracy performance obtained by eliminting all the features other than CNN ones.

CNN features are considered. In addition, despite a common belief, texture features generally yield to unsatisfying results over all the datasets. Such a behavior is largely driven by the intrinsic challenges of each specific dataset.

More specifically, let us consider the UNICT-FD889 dataset. Images of a same category are acquired by taking picture of the same dish under different conditions (e.g. by rotating the plate, zooming in, etc.). Since the considered food images are highly textured and generally present similar gradients, features considering such information (local features included) tend to perform poorly, especially if color information is not considered. Results for the other datasets follow similar motivations.

Kernels:
Results have shown that using a kernel function instead of computing a random mapping between the input and hidden ELM neurons has significant benefits in terms of classification accuracy. In particular, for low values of n the choice of an appropriate kernel matters and can significantly affect the performance. However, different kernels yield to different improvements which also depend on the type of considered feature. Such difference in the results is driven by the specific kernel computation and its parameters.

For instance, when computing the RBF kernel we have set the free parameter to 1 (optimal for all features/datasets on average). However, since this controls the radius of influence of each sample and depends on the magnitude of the considered feature components, it is reasonable that for some specific features the kernel computation yields to a new feature space which is highly separable into the specific food categories.

A similar reasoning could be extended to the joint feature cases. Despite this, it should be noticed that the cosine and the $\chi^2 - exp$ kernels generally improve the recognition performances.

Supervisor:
The deep comparison with other low/mid/high-level fusion scheme has shown that for every dataset the proposed supervisor yields to better performance than other approaches. This substantiate the benefits of the proposed committee-based approach. More specifically, results have demonstrated that the Structural SVM is able to correctly capture the feature importance and can exploit this information to produce better results both in terms of classification accuracy as well as in terms of ranking performance. Moreover, it has negligible impact on the computational burden.

Overall Performance:
The results obtained conducting an extensive analysis and the comparisons with state-of-the-art approaches have shown that, regardless of the considered dataset, the proposed SELC approach can be successfully applied for food recognition purposes. It also scales very well to real-world widely different scenarios. Finally, the computational analysis and the feature importance evaluation have shown that SELC can be easily extended for mobile devices usage.

5. Conclusion

In this paper, a system for automatic food recognition based on a learning committee has been introduced. The committee-based approach has been conceived with the idea that existing ad-hoc image representations based on *a priori* knowledge of the problem might not be sufficient to correctly handle the task. Therefore, a system that uses as many different features as possible but exploits only a subset of those to perform the food classification task has been proposed. The approach has been named Supervised Extreme Learning Committee (SELC). In SELC, each ELM is presented a particular feature type only, hence it highly specializes on classifying food by using a certain feature type. The classification results obtained by the committee members are later fused into a single output by means of a Structural SVM. This produces an optimal plausibility rank.

To demonstrate the benefits of the proposed SELC approach extensive evaluations on four benchmark datasets have been conducted. These demonstrated that SELC has superior performance to the single members taken separately, as well as to other existing fusion schemes. Comparisons with existing methods have shown that SELC is able to outperform the state-of-the-art results on all the considered datasets.

References

[1] World Health Organization, Obesity and overweight - fact sheet n. 311 (2015).
URL http://www.who.int/mediacentre/factsheets/fs311/en/

[2] P. J. Stumbo, New technology in dietary assessment: a review of digital methods in improving food record accuracy, Proceedings of the Nutrition Society 72 (01) (2013) 70–76.

[3] M. C. Carter, V. J. Burley, C. Nykjaer, J. E. Cade, Adherence to a Smartphone Application for Weight Loss Compared to Website and Paper Diary: Pilot Randomized Controlled Trial, Journal of Medical Internet Research 15 (4) (2013) e32.

[4] K. Aizawa, M. Ogawa, FoodLog: Multimedia Tool for Healthcare Applications, IEEE MultiMedia 22 (2) (2015) 4–8.

[5] S. Rousseau, Food "Porn" in Media, in: P. B. Thompson, D. M. Kaplan (Eds.), Encyclopedia of Food and Agricultural Ethics, Springer Netherlands, 2014, pp. 1–8.

[6] N. Martinel, C. Micheloni, Classification of Local Eigen-Dissimilarities for Person Re-Identification, IEEE Signal Processing Letters 22 (4) (2015) 455–459.

[7] N. Martinel, A. Das, C. Micheloni, A. K. Roy-Chowdhury, Re-Identification in the Function Space of Feature Warps, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (8) (2015) 1656–1669.

[8] N. Martinel, C. Micheloni, G. L. Foresti, Kernelized Saliency-Based Person Re-Identification Through Multiple Metric Learning, IEEE Transactions on Image Processing 24 (12) (2015) 5645–5658.

[9] V. Tresp, Committee Machines, in: Y. H. Hu, J.-N. Hwang (Eds.), Handbook for Neural Network Signal Processing, CRC Press, 2001, pp. 1–21.

[10] V. Tresp, A Bayesian Committee Machine, Neural Computation 12 (2000) 2719–2741.

[11] A. Schwaighofer, V. Tresp, The Bayesian Committee Support Vector Machine, in: International Conference on Artificial Neural Networks, 2001, pp. 411–417.
[12] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification., IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics 42 (2) (2012) 513–29.
[13] N. Martinel, C. Micheloni, G. L. Foresti, Evolution of Neural Learning Systems, System Man and Cybernetics Magazine (2015) 1–6.
[14] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Large Margin Methods for Structured and Interdependent Output Variables, Journal of Machine Learning Research 6 (2005) 1453–1484.
[15] a.R Jimenez, a.K Jain, R. Ceres, J. Pons, Automatic fruit recognition : a survey and new results using Range / Attenuation images, Pattern Recognition 32 (1999) 1719–1736.
[16] M.-Y. Chen, Y.-H. Yang, C.-J. Ho, S.-H. Wang, S.-M. Liu, E. Chang, C.-H. Yeh, M. Ouhyoung, Automatic Chinese food identification and quantity estimation, SIGGRAPH Asia (2012) 1–4
[17] G. M. Farinella, M. Moltisanti, S. Battiato, Classifying Food Images Represented as Bag of Textons, in: International Conference on Image Processing, 2014, pp. 5212—-5216.
[18] S. Yang, M. Chen, D. Pomerleau, R. Sukthankar, Food recognition using statistics of pairwise local features, in: International Conference on Computer Vision and Pattern Recognition, 2010, pp. 2249–2256.
[19] L. Bossard, M. Guillaumin, L. Van Gool, Food-101 Mining Discriminative Components with Random Forests, in: European Conference Computer Vision, 2014, pp. 446–461. ,
[20] H. Kagaya, Food Detection and Recognition Using Convolutional Neural Network, in: ACM International Conference on Multimedia, 2014, pp. 1085–1088.
[21] Y. Kawano, K. Yanai, Real-Time Mobile Food Recognition System, Computer Vision and Pattern Recognition Workshops (2013) 1–7
[22] F. Kong, J. Tan, DietCam: Automatic dietary assessment with mobile camera phones, Pervasive and Mobile Computing 8 (1) (2012) 147–163.
[23] F. Zhu, M. Bosch, I. Woo, S. Kim, C. J. Boushey, D. S. Ebert, E. J. Delp, The Use of Mobile Devices in Aiding Dietary Assessment and Evaluation., IEEE journal of selected topics in signal processing 4 (4) (2010) 756–766.
[24] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, I. Essa, Leveraging Context to Support Automated Food Recognition in Restaurants, in: Winter Conference on Applications of Computer Vision, 2015, pp. 580–587.
[25] K. Yanai, T. Kaneko, Y. Kawano, Real-Time Photo Mining from the Twitter Stream: Event Photo Discovery and Food Photo Detection, in: International Symposium on Multimedia, 2014, pp. 295–302.
[26] D. Ravì, B. Lo, G.-z. Yang, Real-time Food Intake Classification and Energy Expenditure Estimation on a Mobile Device, in: IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, 2015, pp. 1–6.
[27] Y. Matsuda, H. Hoashi, K. Yanai, Recognition of multiple-food images by detecting candidate regions, in: International Conference on Multimedia and Expo, 2012, pp. 25–30.
[28] Y. Matsuda, K. Yanai, Multiple-Food Image Recognition Considering Co-occurrence, in: International Conference on Pattern Recognition, 2013, pp. 1724–1730.
[29] Taichi Joutou, Keiji Yanai, A food image recognition system with Multiple Kernel Learning, in: International International Conference on Image Processing, 2009, pp. 285–288.
[30] G. M. Farinella, D. Allegra, F. Stanco, A Benchmark Dataset to Study the Representation of Food Images, in: European Conference Computer Vision Workshops, 2014, pp. 584–599.
[31] B. Julesz, Textons, the elements of texture perception, and their interactions., Nature 290 (1981) 91–97.
[32] M. Varma, A. Zisserman, A Statistical Approach to Texture Classification from Single Images, International Journal of Computer Vision 62 (2005) 61–81.
[33] J. Shotton, M. Johnson, R. Cipolla, Semantic texton forests for image categorization and segmentation, in: International Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[34] J.-M. Geusebroek, R. van den Boomgaard, A. Smeulders, H. Geerts, Color invariance, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (12) (2001) 1338–1350.
[35] K. E. a. van de Sande, T. Gevers, C. G. M. Snoek, Evaluating color descriptors for object and scene recognition., IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1582–96.
[36] A. Bosch, A. Zisserman, X. Munoz, Image Classification using Random Forests and Ferns, in: International Conference on Computer Vision, Ieee, 2007, pp. 1–8.
[37] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.
[38] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in: International Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2169–2178.
[39] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.
[40] E. Rahtu, J. Heikkilä, V. Ojansivu, T. Ahonen, Local phase quantization for blur-insensitive image analysis, Image and Vision Computing 30 (8) (2012) 501–512.
[41] Y. Guo, G. Zhao, M. Pietikäinen, Texture Classification using a Linear Configuration Model based Descriptor, Procedings of the British Machine Vision Conference 2011 (2011) 119.1–119.10
[42] X. Qi, R. Xiao, C.-G. Li, Y. Qiao, J. Guo, X. Tang, Pairwise Rotation Invariant Co-occurrence Local Binary Pattern, IEEE Transactions on Pattern Analysis and Machine Iintelligence 36 (11) (2014) 2199 – 2213.
[43] L. Zhang, Z. Zhou, H. Li, Binary Gabor pattern: An efficient and robust descriptor for texture classification, in: International Conference on Image Processing, Ieee, 2012, pp. 81–84.
[44] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Advances in neural information processing systems, 1999, pp. 487–493.
[45] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher Kernel for Large-Scale Image Classification, in: European Conference on Computer Vision, Vol. 6314, 2010, pp. 143–156.
[46] J. Sanchez, F. Perronnin, T. Mensink, J. Verbeek, Image Classification with the Fisher Vector: Theory and Practice, International Journal of Computer Vision 105 (3) (2013) 222–245.
[47] O. Boiman, E. Shechtman, M. Irani, In defense of Nearest-Neighbor based image classification, in: International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[48] J. Dong, S. Soatto, Domain-Size Pooling in Local Descriptors: DSP-SIFT, in: International Conference on Computer Vision and Pattern Recognition, 2015.
[49] A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition, in: Computer Vision and Pattern Recognition Workshops, 2014.
[50] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, OverFeat: Integrated Recognition , Localization and Detection using Convolutional Networks, in: International Conference on Learning Representations, 2014, pp. 1–15.
[51] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1-3) (2006) 489–501.
[52] D. Achlioptas, Database-friendly random projections: Johnson-Lindenstrauss with binary coins, Journal of Computer and System Sciences 66 (4) (2003) 671–687.
[53] W. B. Johnson, J. Lindenstrauss, G. Schechtman, Extensions of lipschitz maps into Banach spaces, Israel Journal of Mathe-

matics 54 (2) (1986) 129–138.
[54] L. Liu, P. Fieguth, Texture classification from random features, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (3) (2012) 574–586.
[55] C. R. Rao, S. K. Mitra, Generalized Inverse of a Matrix and Its Applications, in: Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, 1972, pp. 601–620.
[56] G.-B. Huang, An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels, Cognitive Computation 6 (3) (2014) 376–390.
[57] A. E. Hoerl, R. W. Kennard, Ridge Regression: Biased Estimation for Nonorthogonal Problems, Technometrics 12 (1) (1970) 55–67.
[58] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, International conference on Machine Learning (2004) 104
[59] T. Joachims, T. Finley, C. N. J. Yu, Cutting-plane training of structural SVMs, Machine Learning 77 (1) (2009) 27–59.
[60] L. Zhu, Y. Chen, A. Yuille, W. Freeman, Latent hierarchical structural learning for object detection, in: International Conference on Computer Vision and Pattern Recognition, 2010, pp. 1062–1069.
[61] L. Bertelli, T. Yu, D. Vu, B. Gokturk, Kernelized structural SVM learning for supervised object segmentation, in: International Conference on Computer Vision and Pattern Recognition, 2011, pp. 2153–2160.
[62] B. Mcfee, G. Lanckriet, Metric Learning to Rank, International Conference on Machine Learning (2010) 775–782.
[63] T. Joachims, A Support Vector Method for Multivariate Performance Measures, International Conference on Machine Learning 440 (2005) 377–384.
[64] Y. Yue, T. Finley, F. Radlinski, T. Joachims, A Support Vector Method for Optimizing Average Precision, in: ACM SIGIR conference on Research and development in information retrieval, 2007, pp. 271–278.
[65] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, J. Yang, PFID: Pittsburgh Fast-food Image Dataset, International Conference on Image Processing (2009) 289–292
[66] N. Martinel, G. Foresti, Multi-signature based person re-identification, Electronics Letters 48 (13) (2012) 764–765.
[67] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: International Conference on Computer Vision, 2009.
[68] X. Liu, L. Wang, G.-B. Huang, J. Zhang, J. Yin, Multiple kernel extreme learning machine, Neurocomputing 149 (2012) (2015) 253–264.
[69] N. Martinel, C. Piciarelli, C. Micheloni, G. L. Foresti, On Filter Banks of Texture Features for Mobile Food Classification, in: International Conference on Distributed Smart Cameras, Seville, Spain, 2015, pp. 11–16.
[70] Y. Kawano, K. Yanai, FoodCam: A real-time mobile food recognition system employing Fisher Vector, Multimedia Tools and Applications (2014) 369–373