Snap, eat, repEat: a Food Recognition Engine for Dietary Logging

Michele Merler
IBM Research
mimerler@us.ibm.com

Hui Wu
IBM Research
wuhu@us.ibm.com

Rosario Uceda-Sosa
IBM Research
rosariou@us.ibm.com

Quoc-Bao Nguyen
IBM Research
quocbao@us.ibm.com

John R Smith
IBM Research
jsmith@us.ibm.com

ABSTRACT

We present a system to assist users in dietary logging habits, which performs food recognition from pictures snapped on their phone in two different scenarios.

In the first scenario, called *Food in context*, we exploit the GPS information of a user to determine which restaurant they are having a meal at, therefore restricting the categories to recognize to the set of items in the menu. Such context allows us to also report precise calories information to the user about their meal, since restaurant chains tend to standardize portions and provide the dietary information of each meal. In the second scenario, called *Foods "in the wild"* we try to recognize a cooked meal from a picture which could be snapped anywhere.

We perform extensive experiments on food recognition on both scenarios, demonstrating the feasibility of our approach at scale, on a newly introduced dataset with 105K images for 500 food categories.

Categories and Subject Descriptors I.4 [**Computing Methodologies**]:Image Processing and Computer Vision
Keywords Food Recognition, Mobile Application

1. INTRODUCTION

Food recognition has recently attracted a lot of attention in the multimedia community, partly due to the deluge of food pictures shared on the web and social media[1] and partly due to the rapid rise of fitness apps which has generated a need for easy logging of calories consumption on mobile devices. Our proposed food recognition engine can represent a fundamental building block of such an application. While there exists preliminary work in this area [24, 30], a reliable, comprehensive solution has yet to be achieved.

There are several challenges to a commercial-grade food visual recognition system for the purpose of logging and evaluating nutrition intake that are inherent to the problem space. First, there are thousands of foods consumed around the world. The *Nutritionix* database[2] lists 22k international foods, 107k restaurant menu items and 500k ingredients (grocery items). Second, many of these dishes are single-ingredient variations of a master recipe, like *Fettuccine Alfredo with Shrimp* and *Fettuccine Alfredo with Chicken*. Third, some ingredient substitutions (e.g. skim milk instead of whole milk) are visually indistinguishable in the finished product and yet they affect nutritional information. Another key challenge consists in the lack of curated, comprehensive data assets that can be used to build fine-grained visual models at scale.

Seeking to address the problem space challenges, we have considered two separate, but related, problems: the recognition of menu items from restaurants, which we name *context-aware recognition* and the recognition of dishes without any context, named *in-the-wild recognition*. In the first case, we can leverage the location context [4, 5, 40], which identifies the restaurant the user is in, and the menu context, which reduces the recognition problem space to a few hundred menu items in the worst case scenario. In fact, the systematic recognition of restaurant food could go a long way to help users do their food logging, especially in the US, where people spend half of their food dollars eating out and 58% of the population eats out at least once a week[3].

The main goal of this work for *context-aware recognition* is to ascertain the performance of visual models when recognizing real-life pictures of foods, versus idealized menu pictures. We have been able to produce highly performing visual models and establish the viability of our approach in the search of an industrial-strength solution.

In the case of *in-the-wild recognition*, we have also constructed a knowledge graph of semantically categorized food dishes. The semantic context will help in differentiating visually similar dishes and, in the case of close matches, will enable the system to return a semantically organized short list of options. We consider this outcome valuable to the user as it would mirror human behavior when confronted with, say, a deep fried dumpling. In this scenario, the system may offer the results of *chicken dumpling*, *beef dumpling* or *vegetable dumpling* as valid options and we believe that this is an expected and useful behavior to most users.

[1] www.foodspotting.com, www.yummly.com

MADiMa'16, October 16 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2986035.2986036

[2] www.nutritionix.com
[3] ushfc.org

The current version of our knowledge graph has 500 specific dishes (instances or leaves) and has been built by following the content links of the *Lists of foods* article in Wikipedia, cross-referenced with 15 common ethnic foods (American, Chinese, English, French, Greek, Indian, Italian, Mediterranean, Middle Eastern, etc.) and with sample menus of local restaurants and common recipes from sites like allrecipes.com and epicurius.com. The idea is to start with a wide variety of common dishes that allows us to test dishes with enough variety and similarity. The knowledge graph has been used as a guide to create a data asset containing almost 150K images of over 500 food categories (Food500).

These two threads, context-aware and in-the-wild recognition, are indeed related, since dishes from restaurants are but instances of the generic recipes, and our knowledge graph categorizes restaurant menu items as instances of the abstract dish categories (e.g., soup, sandwich, dessert).

In fact, food categorization can be viewed as a specific instance of fine grained visual recognition problem [43]. Different approaches have been proposed to recognize food in pictures, from random forests [7] to structured SMVs on top of extreme learning machines [31], from using image captioning techniques [19] to directly training or fine-tuning deep convolutional neural nets from food images [15, 20, 41]. We follow the latter approach, which has produced state of the art results on the Food 101 dataset [7]. Our experimental results show that information learned by fine-tuning a CNN on more generic, "in the wild" images before further fine-tuning on the target domain can significantly help the classification performance of the model in the context of menu items.

Outside of restaurant chains, food calories estimation requires segmentation [17], distance[27] and portion estimation steps. While we recognize the need for such components in a commercial system, they are outside of the scope of this work, where we instead focus on the food classification aspect, as well as the working pipeline of a prototype architecture that has as input a single image (and possibly some context in the form of geo-location and/or restaurant information) and returns the recognized dish category as well as the associated calories count, when available.

This work thus introduces the following contributions:

- we present a system for food recognition to be used in the context of dietary assessment/logging apps
- we test the performance of the proposed visual recognition engine in two contexts (within restaurants, "in the wild") and we establish best practices for a realistic solution working on each context, given its constraints
- we introduce a dataset of 500 food categories and approximately 150K images of cooked meals (Food500), and provide an in depth study of recognition performance on it

The remainder of the paper is organized as follows. We review related works on food recognition in Section 2. We describe our proposed system architecture and food recognition engine in Section 3. Section 4 introduces the Food500 dataset and we describe in Section 5 the recognition experiments performed on such dataset as well as on menu items from six restaurant chains. We finally draw conclusions and discuss future directions in Section 6

2. RELATED WORK

Food logging is an essential tool in nutrition intake tracking and diet management. As the social awareness of the increase in worldwide obesity grows [32], accurate and convenient food logging systems are greatly in demand. This trend has been accompanied by the pervasive use of mobile devices in daily life, which allows people to perform food logging at any time and location. By tracking the various types and portions of food consumed over time, this information can be used for food balance estimation [1, 3] and meal planning [16], which can help to guide people towards a healthier diet. Conventional systems for dietary logging heavily depend on manual input of food consumption information (including food category, volume, etc) [8]. While logging the nutrition information of packaged and branded food is relatively easy, the difficulty in assessing food-in-the-wild (such as most homemade food) has been one of the critical barriers to the growth in food logging applications [12].

Due to the limitations of manual food journaling, many automated food logging algorithms based on image recognition have been proposed. A large body of work encodes the image visual information using certain predefined features, such as texture features [29], bag-of-visual-words features [21], pre-segmented image patches [11], etc., and learns visual classifiers to differentiate different food categories. With the recent progress in deep convolutional neural networks, the performance on food visual recognition has been largely improved [23, 30], similarly to many other image classification problems in computer vision.

Deep neural network based recognition methods have been shown to outperform traditional recognition paradigms which rely on extracting hand-crafted features from images and training "shallow" learners. However, food recognition remains a challenging problem due to the variation in food composition, visual similarity among different dishes, etc. Therefore, many existing approaches leverage additional contextual information to achieve better recognition performance. Herranz et al.[18] proposed a probabilistic model that incorporates the geolocation of restaurants and images with visual information; textual features extracted from the webpage where images were downloaded from were fused with visual features to identify food categories [37]; Chen et al. [9] learn a convolutional neural network that can simultaneously optimize ingredient prediction and food categorization in a multi-task learning framework; Wang et al. [38] utilize personal dietary history to aid food recognition. While these approaches focus on the food recognition performance, other approaches aim to propose end-to-end systems for automatic food logging and nutrition assessment [33, 30, 42]. Such systems all utilize some form of metadata associated with images to enhance recognition, however they only consider one type of information. In contrast, our proposed system can integrate various sources of information, including standard restaurant menu, image and restaurant geolocations.

Although a myriad of approaches have been proposed to address the food recognition problem, the effort on acquiring large-scale, fine-grained food image databases has been lacking. Most existing food databases either cover a narrow range of food categories (e.g., UEC-Food100 [25] focuses on asian-style dishes), or the food categories are not sufficiently fine-grained. Chen et al. [10] introduced an image dataset collected from popular fast food restaurant, with 11 brands of fast food, and 101 food categories. While

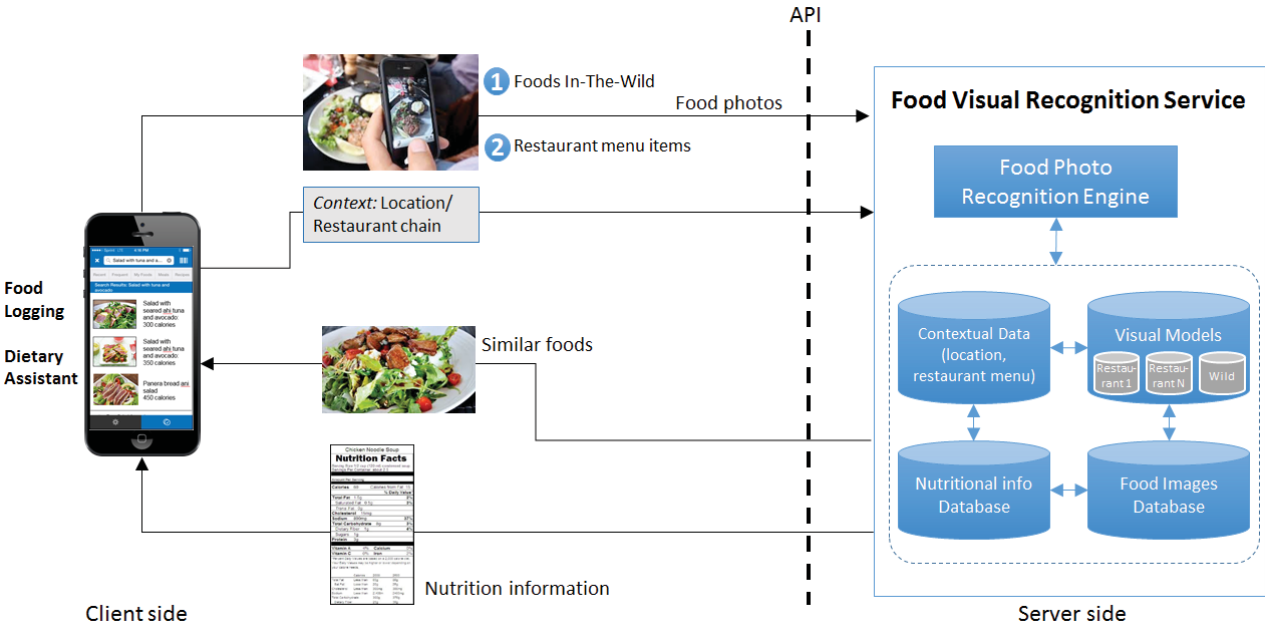

Figure 1: Architecture of the proposed food recognition system.

the UNICT-FD889 dataset [13] has a wilder range of food dishes, the distribution of images in each category is as low as four on average. With very limited training data per category, UNICT-FD889 is not suitable for learning deep learning models that require large amount of training data. The largest available food image datasets are Food-101 [7] and UPMC Food-101 [37]. Both datasets contain 101 categories of food-in-the-wild by selecting the most popular dishes from Foodspotting[4]. Our Food500 dataset, to the best of our knowledge, covers the most amount of categories with over 500 distinct dishes, and each category has on average almost 300 images, which makes it valuable to assess the performance of a food visual recognition engine at scale.

3. SYSTEM ARCHITECTURE

Figure 1 illustrates the architecture of our food recognition-dietary assistant pipeline. The system consists in an interactive web client and a server side recognition engine. The communication between the two components is managed by a specifically designed API.

The client sends two types of information to the server (as a multi-part FormData object sent using XMLHttpRequest):

- one image (mandatory)
- contextual information (optional), in the form of GPS coordinates or restaurant name

On the server side each API request coming from the client is managed with WebSphere Liberty[5], coupled with Java code designed to handle the specific API requests. On the server are stored:

- a database of restaurant chains, with their menu items
- a nutritional information database containing caloric information associated with known menu dishes
- a reference set of food images, on top of which the visual recognition models were built
- a set of visual recognition models: one to filter non-food images, one for each known restaurant chain, and one for recognizing foods "in the wild"

The server returns to the client the top N recognized foods, together with a reference image representing the category, and the calories information of the meal, when available. In the current version of system, $N = 9$, as explained in Section 3.2.

The steps taken by the server once a request is received are detailed in Algorithm 1. For every request image x, the server applies the Food vs Non-Food classifier (FNF) in order to filter out erroneous requests containing images that do not represent food. If $FNF(x)$ is smaller than a threshold t (0.55 in our system), the image is rejected as not containing any recognizable food item. If the image is labeled as food instead, a visual model VM will be scored on it. The choice of VM depends on the contextual information R_i. If no context is provided as part of the client request, the system by default employs the "wild" model VM_w, trained on the 500 food categories described in Section 4. If the restaurant information is available, the system will instead apply the visual classifier associated to that restaurant, and pull the calories information $c(x)$ associated to the recognized dish. For every recognized category, the server returns also a reference image, as a visual feedback to the user.

Currently the server is a single 64 bit Linux node running Redhat 7.2, with 12 CPUs Intel(R) Xeon(R) CPU E5-2683 v3 @ 2.00GHz, each with 4GB of RAM. The full cycle of a request, from the moment a picture is submitted to when the result is returned and displayed on the client side, takes on average 1.5 seconds.

[4]www.foodspotting.com
[5]https://developer.ibm.com/wasdev/websphere-liberty/

Algorithm 1 Server Request Management

1. **Input**:
   (a) Image x (mandatory)
   (b) Restaurant/GPS information R_i (optional)

2. **Process**:
   (a) **if** $(FNF(x) > t)$
   (b) **if** $(\exists\ R_i)$
   (c) $p(x) = VM_i(x)$
   (d) retrieve calories info associated with food $p(x)$
   (e) **else**
   (f) $p(x) = VM_w(x)$

3. **Output**:
   (a) Food class prediction $p(x)$
   (b) Calories info $c(x)$ (if available)
   (c) Reference image for class $p(x)$

3.1 Image Recognition Engines

As mentioned earlier, our food recognition engine contains two types of models: one to discriminate between food and not-food pictures, and one to classify food items.

3.1.1 Food vs Non-Food Model

For the Food vs. Non-Food binary problem, we tested two different models. The first is an ensemble of binary SVMs with linearly approximated χ^2 kernel, each trained on a random bag of positive and negative examples, on top of various low level descriptors related to color, edge, shape and texture. The parameters of the SVMs and the ensemble weights were optimized on cross-validation splits.

The second model (which proved to be superior in our experiments) is a deep convolutional neural network, obtained by fine-tuning a GoogleNet[36] model pre-trained on ImageNet to an appositely created dataset containing 3.2 million food and non-food pictures.

3.1.2 Food Item Classifiers

Once we are sure that an image contains food, we need to recognize which dish is represented. We treat this problem as a standard multiclass classification problem, where the number and nature of the classes depends on the context. We trained a separate model for each Restaurant chain in our dataset, in order to distinguish among the items on the menu, and another one for the 500 foods "in the wild".

Since the number of training images for the menu items in certain food chains might be limited, we adopted first a simple K-NN model, using the 4,096 dimensional deep features extracted by the last non-fully connected layer (fc7) of an AlexNet CNN [26].

When the number of available training images increases, the risk of overfitting diminishes and therefore we were able to train a more sophisticated classifier. Following standard practices [30], we fine-tuned a GoogleNet inception CNN [36] pre-trained on ImageNet. As reported in the experiments in Section 5.2.2, as we evaluated different fine-tuning settings, we registered an improvement in performance when performing a first round of fine-tuning on a food-specific dataset, followed by a second round of fine-tuning on the target restaurant chain menu set.

Each fine-tuned CNN model is relatively compact (42MB), thus ensuring the scalability of the system to a reasonable number of restaurant chains. However, some weights sharing mechanism across models might be required to scale to a functioning commercial solution.

3.2 User Interface

The front end client of the system is implemened in html and javascript. It is a web portal which is optimized to work seamlessly on mobile. The desktop version of the interface presents a drag and drop area for the user to submit an image (Figure 2(a)), while the mobile version accesses the camera or picture library on the user's phone directly. In the displayed example the restaurant information is also available, and the system is notified to use the *Panera Bread* restaurant chain classifier. Once the request is processed, the resulting information is returned as a json file and displayed in the following manner: first the top three general categories are displayed, with their name and reference image. When the K-NN classifier is employed, the reference image is the actual closest neighbor from the reference database. We opted not to display the classification score of each category, since we have found that users tend to have difficulties interpreting the absolute numerical results, which in turn did not add any value. Showing general categories (Figure 2(b)) before specific ones (once the user taps or clicks on one of the general classes, Figure 2(c)) proved instead to be highly beneficial, especially when the system makes mistakes. In that case, even if the proper fine-grained category of *Panera Bread Ancient Grain Arugola and Chicken Salad* might not get recognized, the system at least knows that it's looking at a salad, and not a soup for example. This helps build the confidence of the user in the results returned by the system, even when they are imperfect. Finally, when the user selects the correctly recognized food item, its caloric content is displayed, as illustrated in part (d) of the Figure.

4. FOOD 500 DATASET

A commercial system performing Food recognition "in the wild" necessarily has two hard requirements: coverage and accuracy. A system that does not have a given dish in its database will never be able to recognize it (at best it can provide a similar alternative). On the other hand it's useless to have hundreds of thousands of food items if the system is not able to accurately recognize them. In order to bring our first system prototype into a scale that sufficiently compromises between those two needs, we created a dataset of 508 food dishes, which we call Food500. With almost 150K images, to the best of our knowledge Food500 is the largest existing food recognition dataset.

4.1 Data Collection and Filtering

Our goal was to collect a set of dish categories reflective of the most common foods eaten by consumers in North America, outside of restaurants and chains. Those categories would complement/integrate the menu items that we already had from restaurant chains. We therefore looked at the intersection of the most common dishes according the *Lists of foods* article in Wikipedia, cross-referenced with 15 common ethnic foods (American, Chinese, English, French, Greek, Indian, Italian, Mediterranean, Middle Eastern, etc.) and with sample menus of local restaurants and common recipes from cooking websites[6].

[6]allrecipes.com, epicurius.com

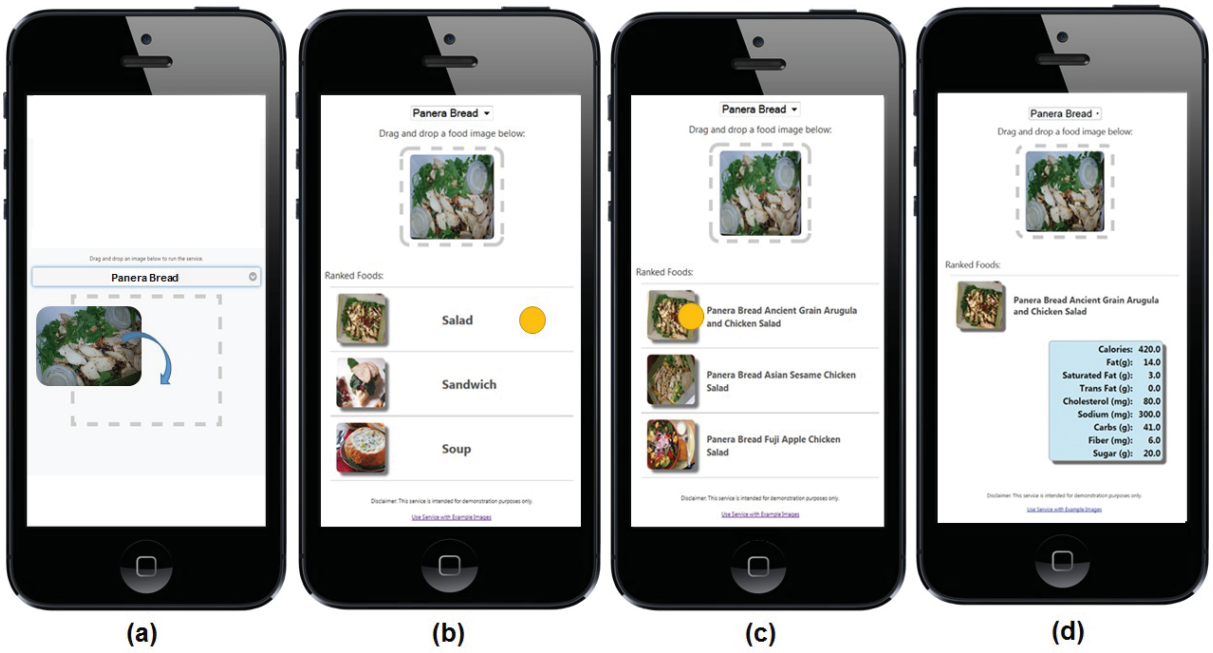

Figure 2: Example of the system interface. The user submits an image (a) and is presented with the top three food categories retrieved by the system, first general (b) and then detailed (c). The system displays also the retrieved calories count associated with the recognized food menu item (d). Note that the interface is designed to work seamlessly on mobile and the web. The yellow dot represents a user click/tap.

The data acquisition, cleaning and annotation pipeline is illustrated in Figure 3. Starting from a list of 540 dish names, we crawled images from multiple web sources and social media sites: namely Google, Bing, Flickr, Instagram and Foodspotting. The result of the initial crawl was a set of almost 3M images. Directly using the crawled images to train a visual classifier would be detrimental, since it is well known that images queried from the web are noisy. In the Figure we show some examples obtained from the query "bacon". Besides the desired food images, the crawls returned some unusable images (blank or too small), as well as non-food images representing the actor Kevin Bacon. We therefore built an automatic filtering pipeline comprising: duplicate detection (based on md5 hash collisions [34]), empty or blank image removal and small images removal (with width or height smaller than 200 pixels). We then employed the Food vs Non-Food classifier described in Section 5.1 to eliminate images not representing food dishes (such as Kevin Bacon). Since the scores of such classifier are normalized to probabilities between 0 and 1, we kept only images above a 0.55 threshold. Finally, we sent the approximately 250K of the top ranked images (according to the food classifier) to Amazon Mechanical Turk[7] for a final verification by humans. This step was necessary since given a food query, some images containing another dish might be returned by a search engine (the burger picture, in the Figure example). For each image, we asked workers to determine if it represented a particular dish by providing the dish name and three reference images. We assigned three workers per image and kept as valid only the images where at least two workers agreed.

[7]https://www.mturk.com/

4.2 Comparison with existing datasets

At the end of the filtering and annotation process we obtained a dataset of 508 food meals categories, with an average of 292 positive images per category, a minimum of 26 (for *yellow corn chips*) and a maximum of 489 (*gulab jaamun*), for a total of almost 150K images.

In Table 1 we compare our Food500 dataset with existing food classification sets, in terms of total number of classes and number of images, as well as average number of images per class. Food500 is currently the largest food classification dataset for total number of images, and ranks among the highest ones also for number of classes and images per class. It must be noted that that all the images in Food500, unlike Food 101 [7] and ETHZ Food 101 [37], have been verified by humans and contain a single food item. In Food 101 [7] only the 250 test images per class were manually annotated, while the remaining 750 training ones were simply collected from Foodspotting by query search, and are therefore noisy. Similarly, ETHZ Food 101 [37] collected the first 1K results of a Google image search, without any further verification. While both datasets provide an interesting challenge in learning from noisy data, our intended commercial application warrants extremely accurate models, thus requiring clean training data.

5. FOOD RECOGNITION EXPERIMENTS

In this Section we describe the image classification experiments that we conducted to validate the effectiveness of the proposed models on state of the art datasets.

5.1 Food vs Non-Food

As mentioned earlier, the very first visual recognition com-

Figure 3: Image acquisition and labeling pipeline.

Dataset	N. Classes	N. Images	Avg.
Food 101 [7]	101	101,000	**1,000**
ETHZ Food 101 [37]	101	101,000	**1,000**
UEC FOOD 256 [22]	356	31,651	89
UNICT-FD889 [14]	889	3,583	4
PFID [10]	101	4,545	45
Food-10k [39]	N/A	12,614	N/A
Geolocalized [40]	**3,832**	117,504	30
Food 500	508	**148,408**	292

Table 1: Comparison of existing food recognition datasets in terms of total number of classes, images, and average images per class.

ponent which is necessary for a food recognition system to work in practice is the ability to distinguish between images representing food and pictures portraying other subjects. In order to develop such component, we collected and annotated (either from the existing food datasets listed in Table 1 or from the web) approximately 1.5 million images of food as positives for our training, and a similar amount of negatives (using queries ranging from people to objects and scenes). We randomly split this collection on 80% training and 20% test, and learned the models described in Section 3.1.1 on the training partition.

We then evaluated our Food vs Non-Food binary classifiers on the test partition, which contains approximately 660K images (43% containing food, 57% not representing food). As shown in Figure 4 and Table 2, the fine-tuned binary GoogleNet model significantly outperforms the Ensemble SVM one.

We tested our models also on the UNICT-FD889 Dataset [14], which has two standard evaluation protocols specifically designed to evaluate food versus non food classifiers. The dataset contains 3,583 images (we'll call them *Foood889*) of 889 food categories snapped with a cellphone in restaurants, as well as 4,805 *Food* and 8,005 *No-Food* images collected from Flickr. We verified via md5 hash that no image in the UNICT-FD889 is contained in the dataset that we collected to train our Food vs Non-Food classifier.

In the first evaluation protocol, a portion of the *Food889* images are used for training as positives, and the remaining as test positives, while the test negatives come from the Flickr *No-Food* portion. In the second evaluation protocol, all *Food889* is used for training, while both Flickr *Food* and *No-Food* images are part of the test set. Note that Farinella et al. [14] use only positive examples to train a

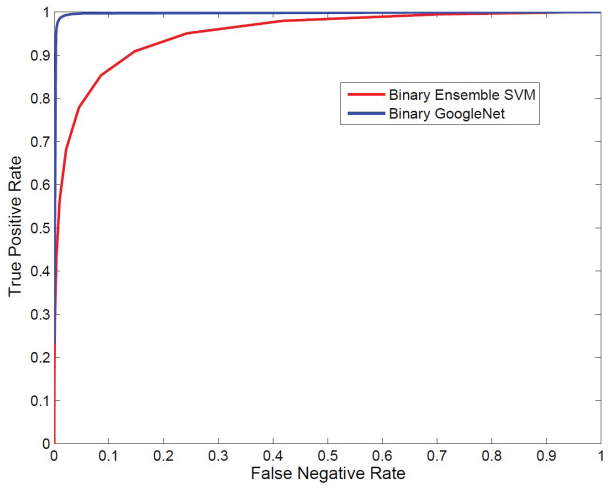

Figure 4: ROC curves for the Food vs Non-Food experiments on the our 660K images Test Set.

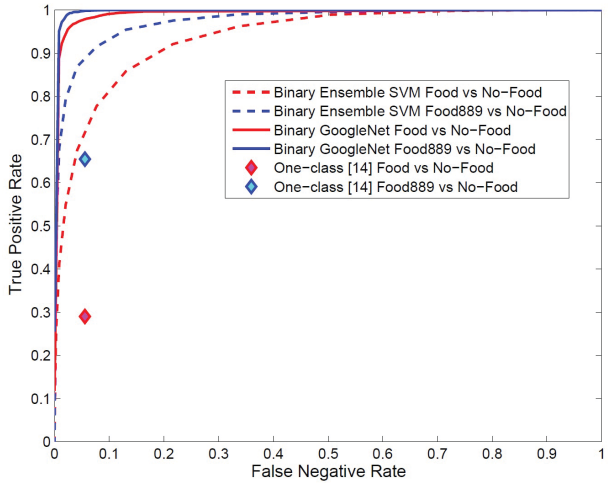

Figure 5: ROC curves for the Food vs Non-Food experiments on the UNICT-FD889 Dataset [14].

one-class SVM, while we trained a binary classifier on a separate dataset, and therefore do not use the training fraction of the UNICT-FD889 Dataset in neither of the evaluation protocols. Figure 5 reports the ROC curves obtained by

Dataset	Metric	One-class SVM[14]	Binary Ensemble SVM	Binary GoogleNet
UniCT	*Food889* True Positives Rate	0.6543	0.8685	0.9711
	Flickr *Food* True Positives Rate	0.4300	0.6744	0.9417
	Flickr *No-Food* True Negative Rate	0.9444	0.9589	0.9817
	Overall Accuracy	0.9202	0.9513	**0.9808**
660K Test Set	Accuracy	-	0.8877	**0.9895**

Table 2: Food vs Non-Food detection rates on the UNICT-FD889 Dataset [14]. The binary Ensemble SVM results are reported with a threshold on the prediction score of 0.55.

our binary classifier as the threshold on its prediction score is changed. The first evaluation protocol is represented in blue, while the second evaluation protocol is in red. The results demonstrate the benefit of training our binary classifier on a large collection of images in comparison to the state of the art one-class SVM approach proposed by Farinella et al. [14] on such dataset, and reported as diamond shaped points in the Figure.

We report the comparison numbers for true positive, true negative and overall accuracy rates on the different portions of the dataset in Table 2. For both our binary models we set the threshold to 0.55. This value was chosen based on performance on a held-out part of our training set. Although the performance is superior to the one-class SVM approach, we noticed that a threshold of 0.6 would have further improved the rates (overall accuracy reaching 0.9641 and 0.9829, respectively). Perhaps some adaptation technique should be investigated in order to guarantee an optimal threshold choice on unseen data.

5.2 Food in Context

5.2.1 Experimental Setup

We evaluated three different models on six different datasets, each containing menu items from one of six popular restaurant chains among the top causal dining in the US. We collected and manually annotated images for each menu item were web sources in the same fashion as what described for Food500 in Section 4. The only exception was *Panera Bread*, for which we also collected pictures snapped by volunteers with smart phones of different menu items. As shown in Table 3, categories in the restaurant chains datasets contain very few images on average, since they are very specific. Food items in this datasets present more standard food compositions and less intra-class variance, compared to standard "wild" datasets like Food 101 [7] or our Food500.

For all the experiments, we randomly selected 75% of the images for training and the remaining 25% for testing.

We trained a separate classifier for each restaurant independently. We tested four different classifiers: 1) K-nn based on AlexNet deep features, 2) AlexNet pre-trained on ImageNet and fine-tuned on the restaurant chain, 3) GoogleNet pre-trained on ImageNet and fine-tuned on the restaurant chain, and 4) GoogleNet pre-trained on ImageNet, fine-tuned first on the positive images from our positive portion of food vs. non-food dataset (which contains 191 general classes), and finally fine-tuned again on the restaurant chain data. We call this last approach *GoogleNet_Food*.

5.2.2 Results and Discussion

From the results reported in Figures 6 and 7 and Table 4 we can derive a few observations:

Restaurant	Classes	Images	Images per class
Applebee's	50	405	8
Au Bon Pain	43	146	3
Dennys	56	325	6
Olive Garden	55	457	8
Panera Bread	79	2,267	28
TGI Fridays	54	432	8

Table 3: Distribution of the restaurant chains datasets.

Model	K-NN	AlexNet	GoogleNet	GoogleNet_Food
Top 1	0.604	0.646	0.700	**0.744**
Top 3	0.821	0.862	0.914	**0.921**

Table 4: Average recognition accuracy across restaurant chains datasets.

- there is a direct correlation between the number of images per category and the accuracy of the models. The more images, the better the performance. In fact, the Panera Bread chain, which has by far the largest number of images, achieves well over 90% Top 1 accuracy.
- generally fine-tuning CNNs is better than using a K-nn model on top of deep features, except when the number of images per class is too small, such in the case of Au Bon Pain, where the K-nn model performs best.
- performing intermediate rounds of fine-tuning, on a dataset different from the target one but still related (general, less specific food categories), helps to boost the performance of the model.

5.3 Food "in the wild"

5.3.1 Experimental Setup

For foods "in the wild", we wanted to test the state of the art in food recognition on our Food 500 dataset. As reported by Myers et al. [30], fine-tuning of a pre-trained GoogleNet on Food 101 achieves a state of the art performance of 79% accuracy. In the same manner as the Food in Context experiments and the Food 101 experimental setup, we randomly split the Food 500 into 75% training and 25% test, and evaluated fine-tuning a GoogleNet CNN on it.

5.3.2 Results and Discussion

Table 5 summarizes the experimental results. It must be noted that we were unable to reproduce the accuracy levels of [30], but nonetheless the accuracy level on Food 500 is considerably lower, given the same model. On one hand

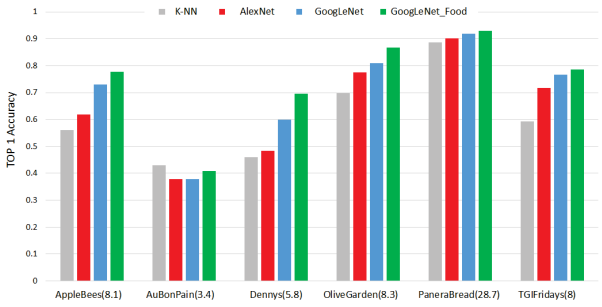

Figure 6: Top 1 classification accuracy for the 6 Restaurant chains datasets. The average number of images per category are reported next to each restaurant chain name.

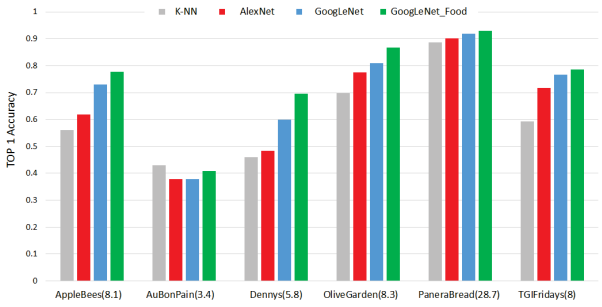

Figure 7: Top 3 classification accuracy for the 6 Restaurant chains datasets. The average number of images per category are reported next to each restaurant chain name.

this drop in performance could be expected given the significantly larger number of classes (508 versus 101) and thus increased potential for confusion. This result demonstrates the challenge presented by the Food 500 dataset, which is one step closer to represent the actual food recognition problem.

In Figure 8 the trend of direct correlation between number of images per class and accuracy is confirmed, similarly to what observed in the Food in Context experiments.

We show the best and worst performing classes in Figure 9 (c) and (a), respectively. It is also interesting to look into the most confused class pairs, as shown in Figure 9 (b). Intuitively they seem to make sense, as their visual appearance is quite similar.

In general this set of experiments has confirmed food recognition "in the wild" to be an extremely challenging problem, for which stronger models are required, able to perform finer-grained distinctions as the number of dishes to be recognized increases.

Dataset	Accuracy
Food 101 [30]	79
Food 101	69.64
Food 500	40.37

Table 5: Average recognition of fine-tuned GoogleNet across "wild" datasets.

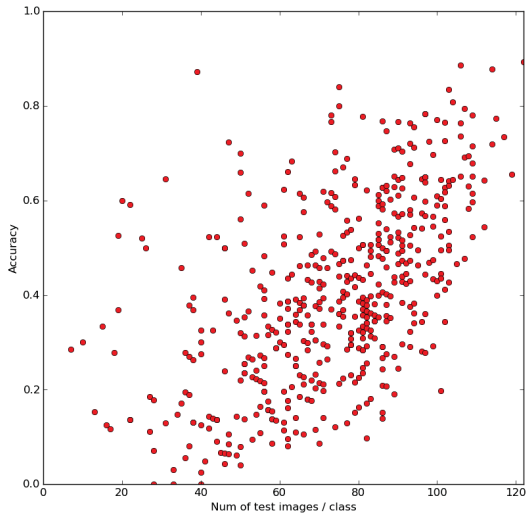

Figure 8: Correlation between accuracy and number of images per class on the Food500 dataset.

6. CONCLUSIONS

We presented an end-to-end system for food recognition and for dietary assistance, composed by an interactive web and mobile client interface and a back end server able to apply suitable visual recognition models according to the context of the request.

Toward the goal of creating a real working system, we have introduced the largest existing food recognition dataset, containing over 500 food categories and almost 150K images.

Extensive experiments for food recognition both "in context", that is, restricting the exploration domain to restaurant menu items, and "in the wild" showcased a need for a minimum number of images per class in order for state of the art fine-tuned CNN architectures to achieve a level of performance that could be acceptable in a commercial application. Multiple rounds of fine-tuning on domain related datasets have proven beneficial.

There are multiple directions which we look forward to explore in order to improve the current system. First, the visual recognition engine can benefit from better modeling, including a learning architecture that could exploit the semantic relationships among food classes. Second, from the data point of view, we aim at growing the database of restaurant menus and dishes at a much larger scale than the initial six we have investigated. Finally, a real working system "in the wild" will need food segmentation and portion estimation components in order to be useful in terms dietary assistance and calories intake logging. We plan to develop components that can serve that purpose within our framework.

7. REFERENCES

[1] K. Aizawa, Y. Maruyama, H. Li, and C. Morikawa. Food balance estimation by using personal dietary tendencies in a multimedia food log. *IEEE Transactions on Multimedia*, 15(8):2176–2185, 2013.

[2] A. H. Andrew, G. Borriello, and J. Fogarty. Simplifying mobile phone food diaries: Design and

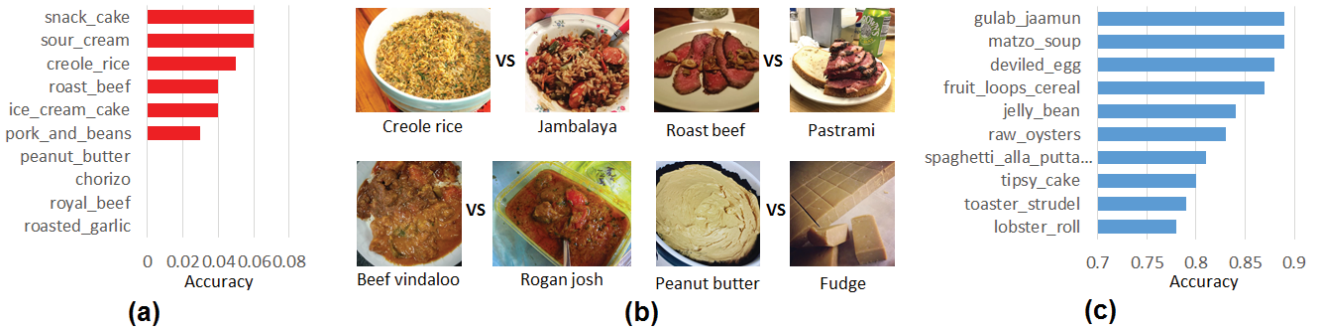

Figure 9: Food 500 classification results: (a) worst performing classes (b) most confused classes (c) best performing classes.

evaluation of a food index-based nutrition diary. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '13, pages 260–263, 2013.

[3] M. Anthimopoulos, J. Dehais, S. Shevchik, B. H. Ransford, D. Duke, P. Diem, and S. Mougiakakou. Computer vision-based carbohydrate estimation for type 1 patients with diabetes using smartphones. *J, Diabetes Science and Technology*, 9(3):507–515, 2015.

[4] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar. Menu-match: Restaurant-specific food logging from images. In *WACV*, pages 844–851, 2015.

[5] V. Bettadapura, E. Thomaz, A. Parnami, G. Abowd, and I. Essa. Leveraging context to support automated food recognition in restaurants. In *WACV*. IEEE, January 2015.

[6] J. Blechert, A. Meule, N. A. Busch, and K. Ohla. Food-pics: an image database for experimental research on eating and appetite. *Frontiers in Psychology*, 5(617).

[7] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[8] M. C. Carter, V. J. Burley, C. Nykjaer, and 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):e32, 2013.

[9] J. Chen and C.-W. Ngo. Deep-based ingredient recognition for cooking recipe retrieval. In *ACM Multimedia*, 2016.

[10] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. In *ICIP*, pages 289–292, Nov 2009.

[11] S. Christodoulidis, M. Anthimopoulos, and S. Mougiakakou. Food recognition for dietary assessment using deep convolutional neural networks. In *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, pages 458–465. Springer International Publishing, 2015.

[12] F. Cordeiro, D. A. Epstein, E. Thomaz, E. Bales, A. K. Jagannathan, G. D. Abowd, and J. Fogarty. Barriers and negative nudges: Exploring challenges in food journaling. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1159–1162, 2015.

[13] G. M. Farinella, D. Allegra, and F. Stanco. A benchmark dataset to study the representation of food images. In *European Conference on Computer Vision*, pages 584–599. Springer, 2014.

[14] G. M. Farinella, D. Allegra, F. Stanco, and S. Battiato. On the exploitation of one class classification to distinguish food vs non-food images. In *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*, pages 375–383. Springer International Publishing, 2015.

[15] Z. Ge, C. McCool, C. Sanderson, and P. I. Corke. Modelling local deep convolutional neural network features to improve fine-grained image classification. In *ICIP*, 2015.

[16] A. Hashimoto, J. Harashima, Y. Yamakata, and S. Mori. *Design in Everyday Cooking: Challenges for Assisting with Menu Planning and Food Preparation*, pages 182–192. 2016.

[17] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp. Food image analysis: Segmentation, identification and weight estimation. In *ICME*, pages 1–6, 2013.

[18] L. Herranz, R. Xu, and S. Jiang. A probabilistic model for food image recognition in restaurants. In *ICME*, pages 1–6, June 2015.

[19] J. Hessel, N. Savva, and M. J. Wilber. Image representations and new domains in neural image captioning. *EMNLP Vision + Learning workshop*, 2015.

[20] H. Kagaya, K. Aizawa, and M. Ogawa. Food detection and recognition using convolutional neural network. In *ACM Multimedia*, pages 1085–1088, 2014.

[21] Y. Kawano and K. Yanai. Real-time mobile food recognition system. In *CVPR Workshops*, pages 1–7, June 2013.

[22] Y. Kawano and K. Yanai. Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In *ECCV 2014 Workshops*, pages 3–17, 2014.

[23] Y. Kawano and K. Yanai. Food image recognition with deep convolutional features. In *UbiComp*, pages 589–593, 2014.

[24] Y. Kawano and K. Yanai. Foodcam: A real-time mobile food recognition system employing fisher

vector. In *International Conference on MultiMedia Modeling, MMM 2014*, pages 369–373, 2014.

[25] Y. Kawano and K. Yanai. Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools Appl.*, 74(14):5263–5287, 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[27] P. Kuhad, A. Yassine, and S. Shimohammadi. Using distance estimation and deep learning to simplify calibration in food calorie measurement. In *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), IEEE International Conference on*, pages 1–6, 2015.

[28] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

[29] N. Martinel, C. Piciarelli, C. Micheloni, and G. L. Foresti. On filter banks of texture features for mobile food classification. In *Proceedings of the 9th International Conference on Distributed Smart Cameras*, ICDSC '15, pages 14–19, 2015.

[30] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy. Im2calories: towards an automated mobile vision food diary. In *ICCV*, 2015.

[31] C. M. Niki Martinel, Claudio Piciarelli and G. L. Foresti. A structured committee for food recognition. In *ICCV*, 2015.

[32] W. H. Organization. Obesity and overweight - fact sheet. June 2016.

[33] P. Pouladzadeh, P. Kuhad, S. Peddi, A. Yassine, and S. Shirmohammadi. Mobile cloud based food calorie measurement. In *Multimedia and Expo Workshops (ICMEW)*, pages 1–6, 2014.

[34] R. Rivest. The md5 message-digest algorithm, 1992.

[35] J. Scharcanski. Bringing vision based measurements into our daily life : A grand challenge for computer vision systems. *Frontiers in ICT*, 3(3), 2016.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[37] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso. Recipe recognition with large multimodal food dataset. In *Multimedia Expo Workshops (ICMEW)*, pages 1–6, June 2015.

[38] Y. Wang, Y. He, F. Zhu, C. Boushey, and E. Delp. The use of temporal information in food image analysis. In *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, pages 317–325. Springer International Publishing, 2015.

[39] M. J. Wilber, I. S. Kwak, D. J. Kriegman, and S. J. Belongie. Learning concept embeddings with combined human-machine expertise. In *ICCV*, 2015.

[40] R. Xu, L. Herranz, S. Jiang, S. Wang, X. Song, and R. Jain. Geolocalized modeling for dish recognition. *IEEE Transactions on Multimedia*, 17(8):1187–1199, Aug 2015.

[41] K. Yanai and Y. Kawano. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *ICME Workshops*, pages 1–6, 2015.

[42] W. Zhang, Q. Yua, B. Siddiquie, A. Divakaran, and H. Sawhney. âĂŹsnap-n-eatâĂŹ: Food recognition and nutrition estimation on a smartphone. *J, Diabetes Science and Technology*, 9(3):525âĂŞ533, 2015.

[43] F. Zhou and Y. Lin. Fine-grained image classification by exploring bipartite-graph labels. In *CVPR*, 2016.