10-2016

Deep-based ingredient recognition for cooking recipe retrieval

Jingjing CHEN

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Deep-based Ingredient Recognition for Cooking Recipe Retrieval

Jingjing Chen
City University of HongKong
Kowloon, HongKong
jingjchen9-c@my.city.edu.hk

Chong-Wah Ngo
City University of HongKong
Kowloon, HongKong
cscwngo@cityu.edu.hk

ABSTRACT

Retrieving recipes corresponding to given dish pictures facilitates the estimation of nutrition facts, which is crucial to various health relevant applications. The current approaches mostly focus on recognition of food category based on global dish appearance without explicit analysis of ingredient composition. Such approaches are incapable for retrieval of recipes with unknown food categories, a problem referred to as zero-shot retrieval. On the other hand, content-based retrieval without knowledge of food categories is also difficult to attain satisfactory performance due to large visual variations in food appearance and ingredient composition. As the number of ingredients is far less than food categories, understanding ingredients underlying dishes in principle is more scalable than recognizing every food category and thus is suitable for zero-shot retrieval. Nevertheless, ingredient recognition is a task far harder than food categorization, and this seriously challenges the feasibility of relying on them for retrieval. This paper proposes deep architectures for simultaneous learning of ingredient recognition and food categorization, by exploiting the mutual but also fuzzy relationship between them. The learnt deep features and semantic labels of ingredients are then innovatively applied for zero-shot retrieval of recipes. By experimenting on a large Chinese food dataset with images of highly complex dish appearance, this paper demonstrates the feasibility of ingredient recognition and sheds light on this zero-shot problem peculiar to cooking recipe retrieval.

Keywords

Food categorization; ingredient recognition; zero-shot retrieval; multi-task deep learning

1. INTRODUCTION

While there is a large number of cooking recipes posted on the Internet, finding a right recipe given a picture of dish remains a challenge yet to be fully explored. The major problem underlying this challenge is the recognition of food categories as well as their ingredients. Indeed, the problem is commonly shared among health-related applications. For example, food-log management [1], which records daily food intake for dietary habit monitoring, often requires manual input of food intake. In addition to time-consuming, the process is error-prone. As investigated in [11], self-reporting data obtained from unfriendly acquired process tends to underestimate the actual food intake. These concerns motivate the use of mobile devices as a convenient means in capturing pictures of food intake for automatic recognition [24] [14] [25] [3] [16].

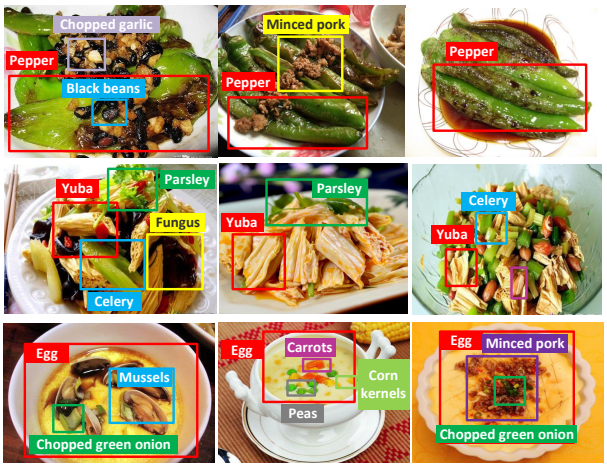


Figure 1: Variations in visual appearance and composition of ingredients show the challenges of predicting ingredients even for dishes within the same food category. The first row shows three examples of dishes for the category "fried green peppers", followed by "yuba salad" ad "steam egg custard" in second and third rows respectively.

This paper studies the recognition of ingredients for recipe retrieval in the domain of Chinese dishes. Different from food categorization, which is to identify the name of a dish (e.g., fried green pepper shown in Figure 1), ingredient recognition is to uncover the ingredients inside a dish (e.g., green pepper, black bean, chopped garlic). In the literature, associating food categories to their respective recipes is regarded as a general pipeline that facilitates the estimation of calories and nutrition facts [35] [14]. The pipeline is effective for recognizing restaurant dishes and the food categories with standardized cooking method (e.g., fast food) that often have similar visual appearance with the same ingredients. However, most dishes in Chinese food have no standardized cooking method, food presentation and ingredient composition. Direct mapping between dishes and recipes, by using the names of food categories, is not likely to attain satisfactory retrieval rate, not mentioning the imperfect performance in food recognition. The difficulty of this task is probably alleviated, nevertheless, with the presence of GPS and restaurant menus as utilized by Im2Calories [24] and Menu-

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2964315

Match [2]. However, restaurant information are difficult to acquire as stated in [24] and such context-aware recognition is only limited to restaurant food. Therefore, this paper argues the need of ingredient recognition beyond food categorization for general recipe retrieval.

In the domain of Chinese food, two major obstacles in recognition are diverse appearances of dishes and wild composition of ingredients. Figure 1 shows some examples of Chinese dishes. Automatic recognition is challenged by the wildly different ways of mixing and placing ingredients even for the same food category. For the food category "steamed egg custard" (last row of Figure 1), there is even no overlap in ingredients except egg. Retrieving recipes without explicitly naming the underneath ingredients is expected to include false positives. Basically, ingredients can be treated as attributes of food categories. As the number of food categories is generally far larger than the number of ingredients, recognizing attributes is more feasible than food categories in terms of scale. Furthermore, ingredient recognition also gives light to the retrieval of recipes for unknown food categories during model training, a problem generally referred to as zero-shot recognition or retrieval [29].

Generally speaking, ingredient recognition is more difficult than food categorization. As observed in Figure 1, the size, shape and color of ingredients can exhibit large visual differences due to diverse ways of cutting and cooking, in addition to changes in viewpoints and lighting conditions. Recognizing ingredients alone without food category in mind is likely to result in unsatisfactory performance. This paper considers simultaneous recognition of food and ingredients, aiming to exploit the mutual relationship between them for enhancing the robustness of recognition. The key ingredients of a category remain similar despite composing with different auxiliary ingredients. Knowing food category basically eases the recognition of ingredients. On the other hand, the prediction of ingredients also helps food categorization, for example, the ingredient "fungus" has a higher chance than "pork" to appear in the food "yuba salad". Hence, learning food categories with the composition of ingredients in mind, and vice versa, in principle shall lead to better performance.

Figure 2 gives an overview of the proposed framework, which is composed of two modules: ingredient recognition and zero-shot recipe retrieval. The first module formulates the recognition of ingredients as a problem of multi-task learning using deep convolution neural network (DCNN). Given a picture of dish, the module outputs the name of dish along with a histogram of ingredients. The developed DCNN can recognize 172 Chinese food categories and 353 ingredients. To the best of our knowledge, there is no result published yet for ingredient recognition on such a large scale. The second module performs zero-shot retrieval, by matching the predicted ingredients against a large corpus containing more than 60,000 recipes. The corpus includes some food categories as well as ingredients unknown to the multi-task DCNN. To boost retrieval performance, a graph encoding the contextual relationship among ingredients is learnt from the recipe corpus. Using this graph, conditional random field (CRF) is employed to probabilistically tune the probability distribution of ingredients to reduce potential recognition error due to unseen food category.

To summary, this paper contributes by developing multi-task learning technique for ingredient recognition and demonstrates its application for zero-shot recipe retrieval. Our work differs from the existing works, which mostly focus on recognition of food categories and operate in domains such as western and Japanese food [4] [22]. To our knowledge, zero-shot recipe retrieval, which requires knowledge of ingredients, has not yet been considered in the literature. Along with this paper, we will release the collected Chinese food dataset, VIREO Food-172, which contains 172 food and 353 ingredient labels. The dataset is larger than the publicly available datasets such as Food-101 [4], UEC Food-100 [22] and PFID [6], each with around 100 western or Japanese food categories but without ingredient labels.

2. RELATED WORK

Variants of recognition-centric approaches have been investigated for different food-related applications. These efforts include food quantity estimation based on depth images [7], image segmentation for volume estimation [25], context-based recognition by GPS and restaurant menus [3], taste estimation [23], multi-food recognition [22], multi-modal fusion [13] and real-time recognition [14]. This section mainly reviews previous works on recognition of food and ingredients using deep and hand-crafted features.

The challenge of food recognition comes from visual variations in shape, color and texture layout. These variations are hard to be tackled by hand-crafted features such as SIFT [21], HOG [8] and color [30]. Instead, deep features extracted from DCNN [17], which is trained on ImageNet [9] and fine-tuned on food images, often exhibit impressive recognition performance [24] [36] [34]. Combination of multi-modal features sometimes also leads to better recognition performance, as reported in [15] [34]. One of best the performances on UEC Food-100 dataset is achieved by fusion of DCNN features with RootHOG and color moment [15], and similarly for UPMC Food-101 dataset by fusion of textual and deep features [34]. Different from these works which directly adopt DCNN for food recognition, this paper proposes new architectures based upon DCNN for simultaneous recognition of food categories and ingredients.

Compared to food categorization, recognition of ingredients receives fewer attentions. One early model is PFD (pairwise local feature distribution) [37], which leverages the result of ingredient recognition for food categorization. In PFD, based upon the appearance of image patches, pixels are softly labeled with ingredient categories. The spatial relationship between pixels is then modeled as a multi-dimensional histogram, characterized by label co-occurrence and their geometric properties such as distance and orientation. With this histogram representation, PFD shows impressive food recognition performance. PFD, nevertheless, is hardly scalable to the number of ingredients. Using only eight categories of ingredients as demonstrated in [37], the histogram already grows up to tens of thousands of dimensions, not mentioning 353 categories as in our paper where the number of dimensions could be as high as ten millions. Our work is more scalable and based on multi-task learning, in contrast to the two-step recognition in PFD without the feedback loop.

Few recent works explore spatial layout [12], feature mining [4] and image segmentation [24] for ingredient or food item recognition. In [12], ingredient regions are detected by shape and texture models, where the shape is based on DPM (deformable part-based model) while the texture is based on STF (semantic texton forest). Similar to PFD [37], the regions are encoded into a histogram modeling spatial relationship between them for food recognition. The spatial relationship is not statistically encoded as in [37], but rather explicit relationships such as "above", "below" and "overlapping" are modeled. Such relationships are helpful for recognizing food such as dessert and fast food, but difficult to be generalized such as for Chinese dishes. In [4], an interesting work which mines the composition of ingredients as discriminative patterns is proposed for food classification. A drawback of this approach is the requirement of image segmentation, which is sensitive to parameter setting

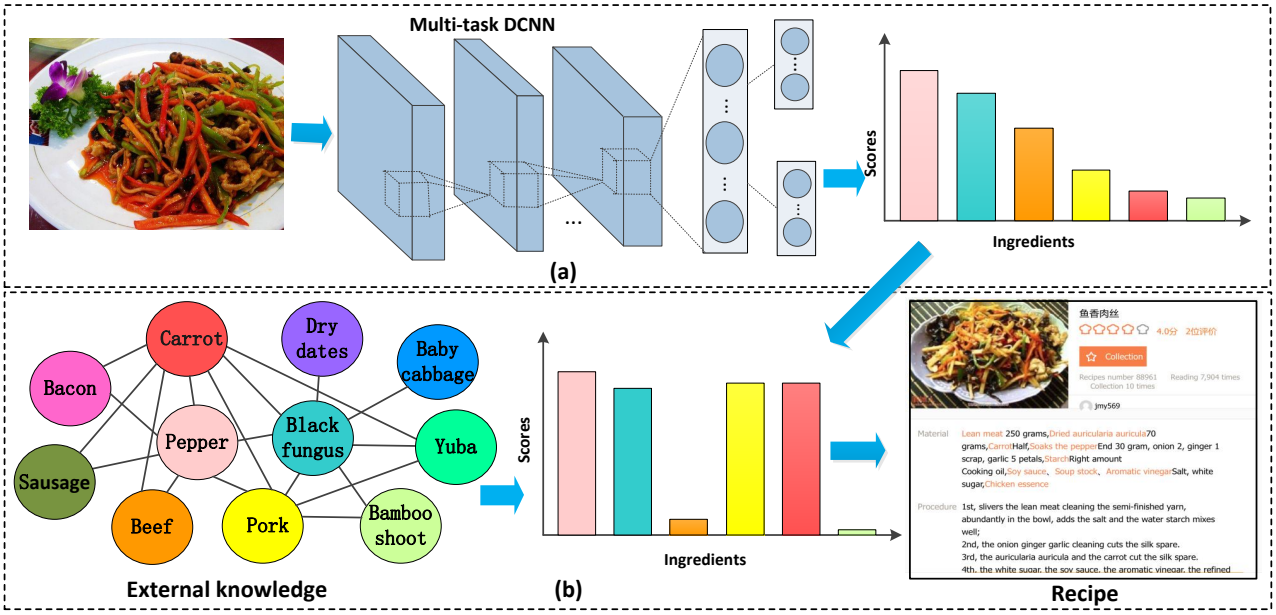


Figure 2: Framework overview: (a) ingredient recognition, (b) zero-shot recipe retrieval. Given a picture of dish with unknown food category, the framework retrieves a recipe for the dish. The recipe is originally in Chinese and Google translated to English.

and can impact recognition performance. As reported in [4], the performance is not better than of DCNN without image segmentation on Food-101 dataset. Similar to [25], image segmentation is employed in [24], but using a more advanced technique based on conditional random field (CRF) with unary potentials provided by DCNN [5]. The promising performance in segmentation for western food, nevertheless, comes from the price for requiring training labels that need manual segmentation of food items for model learning. For Chinese food, collecting such training labels is extremely difficult, given the fuzzy composition and placement of ingredients as shown in Figure 1.

Our work is also related to multi-task learning [10] [31] [32] and fine-grained classification [18] [20] [19]. In [10], multi-task DCNN models are proposed for simultaneous categorization and pose estimation of general objects (e.g., airplane, sofa, car). In [31], a deep network is designed for the tasks of face classification and verification. As concluded by [31], adding one more supervisory signals for feature learning greatly improves the performance of face verification. Similarly for cascade network [18] [20] and bilinear model [19] which couple multiple deep models for fine-grained classification. Nevertheless, these models mostly focus on localization, alignment and classification of object parts, which are not directly applicable to our problem. As ingredients can scatter around a dish and occlude each other (e.g., "yuba salad" in Figure 1), localization and alignment of ingredients are hardly applied for food domain. To the best of our knowledge, there is no multi-task learning model yet developed for food recognition.

3. MULTI-TASK DEEP LEARNING

The conventional DCNN is an end-to-end system with input as picture and output as the prediction scores of class labels. DCNN models such as AlexNet [17] and VGG [28] are trained under the single-label scenario, specifically, there is an assumption of exactly one label for each input picture. As ingredient recognition is a multi-label problem, i.e., more than one labels per image, a different loss function needs to be used for training DCNN. On the other hand, directly revising DCNN with appropriate loss function for ingredient recognition may not yield satisfactory performance, given the varying appearances of an ingredient in different dishes. To this end, we propose to couple food categorization problem, which is a single-label problem, together with ingredient recognition for simultaneous learning.

3.1 Architecture Design

We formulate food categorization and ingredient recognition as a multi-task deep learning problem and modify the architecture of DCNN for our purpose. The modification is not straightforward for involvement of two design issues. The first issue is about whether the prediction scores of both tasks should *directly* or *indirectly* influence each other. Direct influence means that the input of one task is connected as the output of another task. Indirect influence decouples the connection such that each task is on a different path of the network. Both tasks influence each other through updating the shared intermediate layers. The second issue is about the degree in which the intermediate layers should be shared. Ideally, each task should have its own private layer(s) given that the nature of both tasks, single versus multi-labeling, is different. In such a way, the updating of parameters can be done more freely for optimization of individual performance.

Based on the two design issues, we derive four different deep architectures as depicted in Figure 3, respectively name as Arch-A to Arch-D. The first design (Arch-A) considers stacked architecture by placing food categorization on top of ingredient recognition, and vice versa. As the composition of ingredients for different dishes under the same food category can be different, this architecture has the risk that model learning converges slowly as observed in the experiment. The second design (Arch-B) is similar except that indirect influence is adopted and both tasks are at different pathways. Both designs are relatively straightforward to implement by adding additional layers to DCNN. The next two architectures consider the decoupling of some intermediate layers. The third design (Arch-C) allows each task to privately own two intermediate layers on top of the convolutional layers for parameter learning. The last design

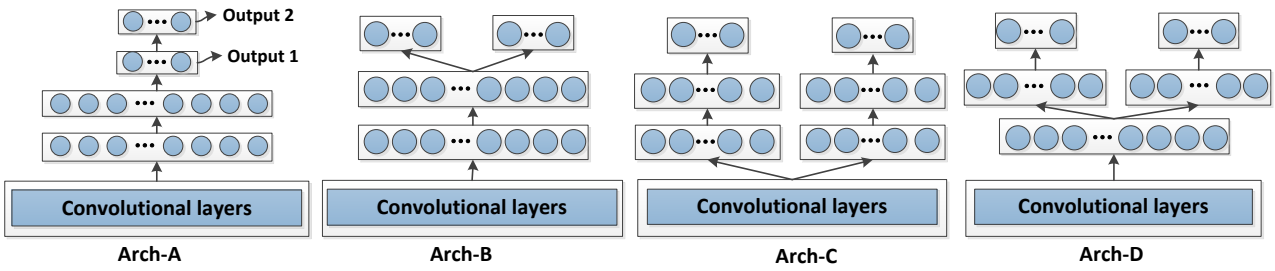


Figure 3: Four different deep architectures for multi-task learning of food category and ingredient recognition.

(Arch-D) is a compromise version between the second and third architectures, by having one shared and one private layer. Arch-D has the peculiarity that the shared layer can correspond to the high or mid-level features common between two tasks at the early stage of learning, while the private layer preserves the learning of specialized features useful for optimizing the performance of each task.

3.2 Implementation

The architectures are modified from VGG 16-layers network [28]. In terms of design, the major modification is made on the fully connected layers. For the private layers in Arch-D, there are 4,096 neurons for food categorization, and 1024 neurons for ingredient layers. Due to different natures of the tasks, we adopt multinomial logistic loss function L_1 for single-label food categorization, and cross-entropy as the loss function L_2 for multi-label ingredient recognition. Denote N as the total number of training images, the overall loss function L is as following:

$$L = -\frac{1}{N}\sum_{n=1}^{N}(L_1 + \lambda L_2) \quad (1)$$

where λ is a parameter trading off the loss terms. This loss function is also widely used in other works such as [31]. During training, the errors propagated from two branches are linearly combined and the weights of first 11 layers shared between two tasks will be updated accordingly. The updating will subsequently affect the last two layers simultaneously, adjustify the features separately owned by food and ingredient recognition. Let $\hat{q}_{n,y}$ as the predicted score of an image x_n for its ground-truth food label y, L_1 is defined as following:

$$L_1 = log(\hat{q}_{n,y}) \quad (2)$$

where $\hat{q}_{n,y}$ is obtained from softmax activation function. Furthermore, denote $p_n \in \{0,1\}^I$, represented as a vector in I dimensions, as the ground-truth ingredients for an image x_n. Basically p_n is a binary vector with entries of value 1 or 0 indicating the presence or absence of an ingredient. The loss function L_2 is defined as

$$L_2 = \sum_{c=1}^{I} p_{n,c} log(\hat{p}_{n,c}) + (1 - p_{n,c}) log(1 - \hat{p}_{n,c}) \quad (3)$$

where $\hat{p}_{n,c}$ denotes the probability of having ingredient category c for x_n, obtained through sigmoid activation function.

4. ZERO-SHOT RETRIEVAL

Training a deep network for recognizing all available food categories is not feasible. In addition to the reality that there exist more than tens of thousands of categories, collecting training examples for each of the categories can be a daunting task. Hence, a practical problem is how to leverage the limited knowledge learnt in a network for recognizing dishes of previously unseen category. As the proposed architectures are capable of predicted ingredients, in principle the problem can be addressed by retrieving recipes through matching of ingredients. We refer this problem to as zero-shot retrieval, which is to find recipes for test pictures of unseen food categories. Two scenarios are considered here. Suppose each recipe is associated with a picture of the dish. The first scenario is to use the FC7 features, specifically the features extracted from the private layer(s) of Arch-C or Arch-D, to represent images for retrieval. In other words, the search of recipe is equivalent to image retrieval. The second scenario assumes absence of pictures in recipes, and uses the predicted scores of ingredients as the semantic labels for text-based retrieval of recipes. As the approach for the first scenario can be straightforwardly implemented, this section focuses on the presentation of the second scenario. The idea is to incorporate external knowledge to refine the predicted ingredient scores for more realistic way of zero-shot retrieval.

4.1 Ingredient Refinement with CRF

While the composition of ingredients is fuzzy in Chinese food, the mixing is not purely random. Intuitively, certain groups of ingredients co-occur more often (e.g., corn and carrot), while some ingredients are likely exclusive of each other (e.g., fish and beef). Such statistics can be mined from training data and utilized for adjusting the predicting scores of ingredients. Nevertheless, considering the zero-shot problem and potentially the limited knowledge in deep network, we mine the statistics from a large corpus composed of more than 60,000 Chinese cooking recipes. The major advantage of doing so is to learn a graph modeling ingredient relationships, where their correlations are more generalizable and not restricted by training data, and hence enhance the success rate of zero-shot retrieval.

We extract ingredients from recipes and construct a graph modeling their co-occurrences based on conditional random field (CRF). Denote $\mathcal{N} = \{c_1, ..., c_I\}$ as the set of available ingredients and I as its set cardinality. The graph G is composed of the elements of $\mathcal{N}$ as vertices and their pairwise relationships, denoted as $\phi_i(\cdot)$, as edges. Further let l_i as an indication function that signals the presence or absence of an ingredient c_i. The joint probability of ingredients given the graph is

$$p(l_1, ...l_I) = \frac{1}{Z(\phi)} exp(\sum_{i,j \in \mathcal{N}} l_i l_j \phi(i,j)) \quad (4)$$

where $Z(\cdot)$ is a partitioning function. To learn the graph, we employ Monte Carlo integration to approximate $Z(\cdot)$ and the gradient descent to estimate $\phi(\cdot)$ to optimize the data likelihood [26]. Given a test image, CRF inferences a label sequence $\mathbf{y}$ based on the graph G. The energy function for inference is composed of unary and

pairwise potentials, defined as

$$E(\mathbf{y}) = \sum_{c \in \mathcal{N}} \psi_u(y_c) + \sum_{(c,v) \in \varepsilon} \psi_p(y_c, y_v) \quad (5)$$

where ε denotes the set of pairwise cliques. The unary term is set as $\psi_u(y_c) = -log(x_c)$, where x_c is the predicted score by the deep network for ingredient c. The pairwise potential is defined as

$$\psi(y_u, y_v) = \begin{cases} 0 & \text{if } y_u = y_v \\ \phi(y_u, y_v) & \text{if } y_c \neq y_v \end{cases} \quad (6)$$

where the value of $\phi(\cdot)$ is obtained from the graph G. Through inferencing, CRF searchs for the optimal label sequence of $\mathbf{y}$ that agrees with the predicted scores and the contextual relationship captured in the graph G. We employ off-the-shelf algorithm, loopy belief propagation [33], for minimize Eqn-5. The output label sequence $\mathbf{y}$ will indicate the presences or absences of ingredients and their probabilities.

4.2 Recipe Search

With the output sequence $\mathbf{y}$ by CRF, a query image is represented as a vector Q^i. Every element in Q^i corresponds to an ingredient and its value indicates the probability output by CRF. On the other hand, the ingredients extracted from a recipe is represented as a binary vector O. The matching score, s_i, between them is defined as

$$s_i = \sum_{c \in O \cap c \in Q^i} x_c \quad (7)$$

Note that the score is not normalized in order not to bias recipes with a small number of ingredients. As a result, Eqn-7 tends to give a higher score for the recipes with excessive number of ingredients. To prevent such cases, the matching between Q^i and O is performed only for the top-k predicted ingredients with higher probability scores. The value of k is empirically set to 10 as there are few recipes with more than 10 ingredients in our dataset.

5. DATASET COLLECTION

We construct a large food dataset specifically for Chinese dishes, namely VIREO Food-172[1], which is made publicly available. Different from other publicly available datasets [4] [22] [6], both food category and ingredient labels are included. In addition, a large corpus of recipes along with dish pictures is also collected.

5.1 VIREO Food-172

The food categories were compiled from "Go Cooking" [2] and "Meishi" [3], which are two websites for popular Chinese dishes. We combine the categories from both websites by removing duplication. All the images in the dataset were crawled from Baidu and Google image search. For each category, the name was issued as keywords in Chinese to search engines. Categories with no more than 100 images returned were removed from the list. For the remaining categories, we manually checked each crawled images up to the depth of 1,300, for excluding images with the resolution lower than 256 $\times$ 256 or suffer from blurring, images with more than one dishes, and false positives. This process ends up with 172 food categories in the dataset.

The 172 categories cover eight major groups of food, as shown in Figure 5. The group *meat* contains the most number of categories, with examples include "braised pork" and "sautéed shredded pork in sweet bean sauce". On the other hand, there are only eight categories under the group *bean product*, and examples include "Mapo tofu" and "braised tofu". All the images in the dataset were crawled from Baidu and Google image search. The names of food categories, were issued as keywords in Chinese to search engines, and 1,300 images are crawled per food category. Figure 4 shows some examples of food categories in VIREO Food-172.

5.2 Ingredient labeling

We compiled a list of more than 300 ingredients based on the recipes of 172 food categories. The ingredients range from popular items such as "shredded pork" and "shredded pepper" to rare items such as "codonopsis pilosula" and "radix astragali". Labeling over hundreds of ingredients for over hundred thousands of images could be extremely tedious, not mentioning the challenge of ingredient annotation. First, some ingredients are difficult to be recognized, for example, ingredients under soup or sauce. Second, some ingredients are invisible in flour-made food categories such as dumpling and noodle. Third, certain ingredients such as egg exhibit large visual variations (see Figure 6) due to different ways of cutting and cooking. Hence, the labeling considers only the annotation of visible ingredients. In addition, we create additional labels for ingredients with large visual appearance, for example, we have 13 different labels for "egg", such as "preserved egg slices" and "boiled egg".

We recruited 10 homemakers who have cooking experience for ingredient labeling. The homemakers were instructed to label only visible and recognizable ingredients. They were also allowed to annotate new ingredients not in the list, which would be explicitly checked by us. To guarantee the accuracy of labeling, we purposely awarded homemakers with cash bonus as incentives to provide quality annotation, in addition to regular payment. For this purpose, we checked a small subset of labels and provided immediate feedback to homemakers such that they were aware of their performance. The whole labeling process ended in two weeks. By excluding images with no ingredient labels, VIREO Food-172 contains a total of 353 ingredient labels and 110,241 images, with the average of 3 ingredients per image. Figure 7 shows the distribution of positive examples in food and ingredient categories. On average, there are 640 positive samples per food category, and 745 per ingredient.

5.3 Recipe Corpus

The corpus was compiled from a popular website "Xinshipu" [4]. The website offers ontology for 530 key ingredients in Chinese food. Using all of these ingredients as queries, a total of 65,284 Chinese cooking recipes were crawled from this website. Each recipe basically contains four sections, including brief introduction, ingredient list, cooking procedure, and a picture showing the appearance of the dish. The recipes were uploaded by Internet users, and thus there may be multiple recipes sharing the same name but with different ingredient lists. Conversely, there are also few recipes about the same dish but in different names.

6. EXPERIMENTS

We split the experiments into three parts, verifying the performances of multi-task learning (Section 6.1), the impact of CRF (Section 6.2) and the application for zero-shot retrieval (Section 6.3). The first part aims to evaluate different deep architectures for multi-task learning in comparison to single-task DCNN. The last

[1] http://vireo.cs.cityu.edu.hk/VireoFood172/
[2] https://www.xiachufang.com/category/
[3] http://www.meishij.net
[4] http://www.xinshipu.com

Figure 4: Examples of food categories in VIREO Food-172.

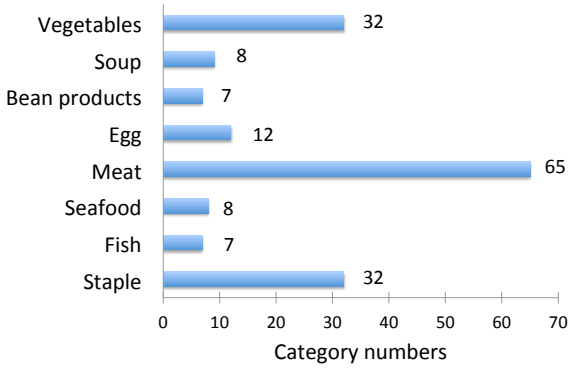


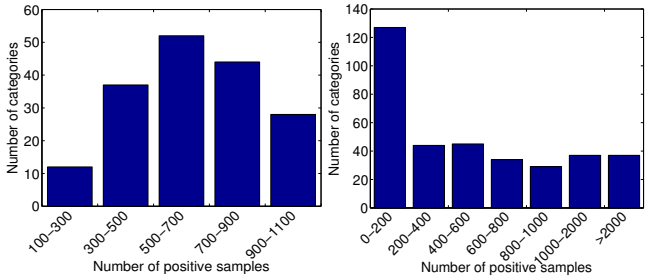
Figure 5: The distribution of food categories under eight major food groups in VIREO Food-172.

Figure 6: The ingredient "egg" shows large difference in visual appearance across different kinds of dishes.

part aims to demonstrate the merit of leveraging ingredient labels for novel recipe retrieval.

6.1 Deep Architectures

The experiments are conducted mainly on VIREO Food-172 dataset. In each food category, 60% of images are randomly picked for training, while 10% for validation and the remaining 30% of images for testing. For performance evaluation, the average top-1 and top-5 accuracies are adopted for food categorization, which are standard measures for the single-label task. For ingredient recognition which belongs to multi-label, micro-F1 and macro-F1 that take into account both precision and recall for each ingredient are employed.



(a) Food category (b) Ingredient

Figure 7: The distribution of food categories (a) and ingredients (b).

The evaluation compares baseline, single and multi-task learnings. The baseline includes SVM classifiers trained using hand-crafted (Gist [27] and color moment [30]) and deep (FC7 of DCNN [17]) features. The single-task learning includes the AlexNet and VGG networks fine-tuned on training and validation sets. Note that for baseline and single-task, different classifiers and networks need to be trained separately for food categorization and ingredient recognition. Specifically, multi-label SVM (MSVM) is trained for baseline, and cross entropy loss function (Eqn-3) is used for single-task DNN. The multi-task learning includes the four deep architectures illustrated in Figure 3. Note that we experiment two variants of Arch-A, with the layer of food categorization on top of ingredient recognition (Arch-A1) and vice versa (Arch-A2).

Grid search of parameters is performed to find the best possible model settings for all the compared approaches, based on the training and verification sets. As ingredient recognition involves multiple labels, a threshold is required to gate the selection of labels. The threshold is set to be the value of 0.5 following the standard setting when sigmoid is used as the activation function. For multi-task deep architectures, the learning rate is set to 0.001 and the batch size to 50. The learning rate decays after every 8,000 iterations. Using Arch-D as example, Figure 8 shows the impact of λ parameter in Eqn-1. Basically, the Top-1 and Mirco-F1 measures fluctuate within the range of 0.06, when the value of λ varies from 0.1 to 1.0. The best performances attained for food categorization (Top-1) is when $\lambda = 0.1$, and for ingredient recognition (Micro-F1) when $\lambda = 0.3$. To balance the performances, we use F1 of Top-1 and Micro-F1 measures to pick the optimal value, where $\lambda = 0.2$ as shown in Figure 8.

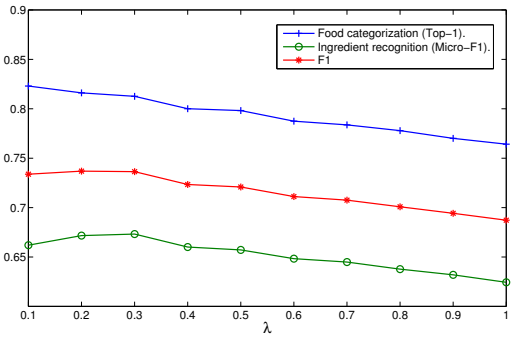


Figure 8: Sensitivity of λ parameter in Eqn-1 for multi-task deep architecture Arch-D.

Table 1 lists the performance for food categorization. The general trend is that deep architectures significantly outperform baselines with either deep or hand-crafted features, while large performance gap is also observed between the results of VGG network and AlexNet. Among the deep architectures for multi-task learning, the designs based on simple modification of DCNN, i.e., Arch-A and Arch-B, show slightly worse performance in Top-1 accuracy compared with single-task VGG. Since the recognition results for both food and ingredients are imperfect, layer stacking as in Arch-A actually could hurt each other's performance. Specifically, the inaccurate prediction in one task will directly affect the other task. On the other hand, while having separate paths as in Arch-B leads to better performance, the improvement is rather minor by the fact that both tasks share the same lower layers. Basically the performances of Arch-C and Arch-D show the merit of having separate paths and layers for both tasks. Arch-C, which only shares convolution layers, improves slightly over single-task VGG. We speculate that the design of Arch-C eventually trains two independent leaners and hence the advantage over single-task is not obvious. Arch-D, which shares one layer while also learning separate layers tailor-made for different tasks, attains the best performance among all the compared approach for both average Top-1 and Top-5 accuracies.

Table 2 shows the performance of ingredient recognition, and similar trends are observed as food categorization. For multi-task learning, all deep architectures but except Arch-A outperform single-task VGG, and with larger performance gaps compared with food categorization. The result basically verifies the merit of joint learning for both tasks. Different from food categorization, sharing layers appears to be a better design choice for ingredient recognition when comparing Arch-B and Arch-C. The best result is attained by Arch-D, which could be viewed as a compromised design between Arch-B and Arch-C. To verify that the improvement is not by chance, we conduct significance test to compare multi-task (Arch-D) and single-task (VGG) using the source code provided by TRECVID[5]. The test is performed by partial randomization with 100,000 numbers of iterations, with the null hypothesis that the improvement is due to chance. At a significance level of 0.05, Arch-D is significantly different from VGG in both food categorization and ingredient recognition by Top-1 accuracy and Macro-F1, respectively. The p-values are close to 0, which reject the null hypothesis.

To validate the proposed work on other food domain, we also conduct experiments on UEC Food-100 [22] dataset for Japanese dishes. The dataset contains 100 categories of food and totally 12,564 images. Each category has at least 100 positive examples. Nevertheless, ingredient labels are not provided. Similar to VIREO

[5]http://www-nlpir.nist.gov/projects/t01v/trecvid.tools/randomization.testing

	Method	Top-1 (%)	Top-5 (%)
Baseline	FC7	48.02	72.01
	Gist	15.39	31.85
	CM	16.54	39.76
Single-task	AlexNet	64.91	85.32
	VGG	80.41	94.59
Multi-task	Arch-A1	78.58	94.24
	Arch-A2	78.63	94.10
	Arch-B	79.05	94.70
	Arch-C	80.66	95.05
	Arch-D	**82.06**	**95.88**

Table 1: Average top-1 and top-5 accuracies for single-label food categorization on VIREO Food-172 dataset.

	Method	*Micro-F1 (%)*	*Macro-F1 (%)*
Baseline	FC7	42.94	32.22
	Gist	23.01	19.45
	CM	21.08	14.06
Single-task	AlexNet	47.63	34.81
	VGG	60.81	43.73
Multi-task	Arch-A1	55.17	43.75
	Arch-A2	59.69	43.48
	Arch-B	66.32	44.85
	Arch-C	63.44	44.26
	Arch-D	**67.17**	**47.18**

Table 2: Performance of multi-label ingredient recognition on VIREO Food-172 dataset.

Food-172, we compiled a list of 190 ingredients for Japanese food and conducted manual labeling. A total of 1,997 images are excluded from experiments for no ingredient labels. The experiment is conducted based on 5-fold cross-validation, using the same data split and settings as [36]. In [36], DCNN based on AlexNet is first pre-trained with 2,000 categories in ImageNet, including 1,000 food-related categories. The network is then fine-tuned with training examples in the dataset. Table 3 lists the detailed performance. Note that, although not using 1,000 food categories for pre-training, Arch-D still manages to outperform [36] by 3.5% in terms of average top-1 accuracy for food categorization. Overall, similar to the performance on VIREO Food-172, Arch-D attains the best performances for both tasks.

Method	Categorization		Ingredient recognition	
	Top-1 (%)	Top-5 (%)	*Micro-F1 (%)*	*Macro-F1 (%)*
FC7	58.03	83.71	52.80	32.51
Gist	30.53	58.80	23.93	11.84
CM	24.11	46.42	16.01	7.830
AlexNet	75.62	92.43	55.62	35.63
VGG	81.31	96.72	57.38	38.62
[36]	78.77	95.15	–	–
Arch-D	**82.12**	**97.29**	**70.72**	**43.94**

Table 3: Performance comparision on UEC Food-100 dataset.

6.2 Effect of CRF

This section verifies the use of CRF in refining the predicted ingredients. All the 65,284 recipes are used for the construction of CRF. A special note is that most recipes do not include the fine-grained description of ingredients. For example, a recipe will simply list "egg" as ingredient, instead of explicitly stating whether the ingredient as either "sliced egg" or "boiled egg". Rather such

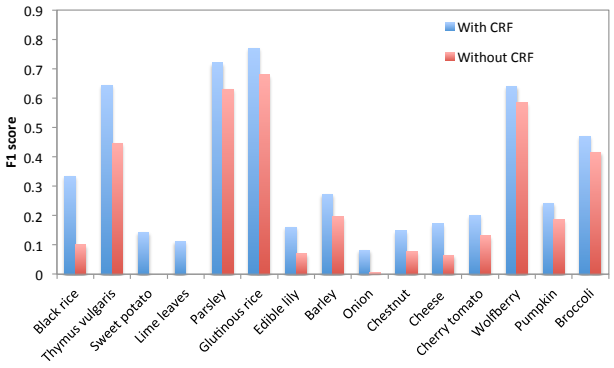


Figure 9: The F1 scores of 15 ingredients that achieve large margin of improvement after CRF.

information can only be inferred from cooking procedure, such as "leaving the eggs boil for 4 minutes". In this experiment, we do not perform natural language processing to obtain the fine-grained description of ingredients. As a consequence, some labels in VIREO Food-172 are merged and this ends up to 257 ingredient labels for experimentation. For the deep architectures, max pooling is adopted to merge the results of fine-grained ingredients. Specifically, if a network predicts "boiled egg" with the probability of 0.5 and "sliced egg" with 0.1, the probability for "egg" is set to be 0.5 in the CRF.

	Method	*Micro-F1 (%)*	*Macro-F1 (%)*
Baseline (recipe)	Food category	40.75	37.47
	Food ingredient	37.39	33.69
Single-task	Without CRF	63.94	46.81
	With CRF	66.23	48.25
Multi-task	Without CRF	68.84	49.98
	With CRF	**71.25**	**51.18**

Table 4: Ingredient recognition with contextual modeling using CRF.

In addition to assessing the effect of CRF for single and multi-task learnings, we also compare the results against the baseline that directly infers the ingredients from a retrieved recipe. More specifically, given a predicted food category by VGG network, the corresponding recipe is retrieved based on name matching. The predicted labels are then based on the ingredients listed in the recipes. This strategy is often used by some approaches [14] for estimation of nutrition facts. We compare to two baselines, based on the predicted names of food categories or ingredients. Note that as a few recipes have the same name despite using different ingredients, and hence multiple recipes could be retrieved. In this case, we only show the result for the recipe which obtains the highest F1 score.

Table 4 lists the performance of different approaches. Note that the performance of multi-task is based on Arch-D. Basically, CRF improves the performance of both single and multi-task learnings. All variants of baseline perform poorly in this experiment, far lower than directly using the ingredients predicted by deep architectures. The result is not surprising due to the fact that, for Chinese food, the composition of ingredients for dishes under the same category can vary depending on factors such as geographical regions, weather and culture.

A few ingredients record large improvement as shown in Figure 9. The examples include "black rice" (F1 score = 0.1 to 0.33), "sweet potatoes" (F1 score = 0 to 0.14), and "cherry tomato" (F1 score = 0.13 to 0.2). CRF successfully captures the knowledge that "black rice" often co-occurs with "rice" and "soybeans" for food categories involving "cereal porridge". Similarly for the co-occurrence among "cherry tomato", "corn" and "lettuce" for food categories related to "vegetable salad". Figure 10 shows a few of success and fail examples refined by CRF. While CRF successfully improves the F1 score and particularly the precision of detection, the recall for few labels is also dropped as noticed in Figure 10(e) and Figure 10(f). Overall, the average precision (micro) is boosted from 0.795 to 0.833, with 193 out of 257 ingredients show improvement. The average recall (micro) is also boosted from 0.607 to 0.623, with 91 ingredients show improvement and 88 ingredients dropped.

6.3 Zero-shot Recipe Retrieval

This section assesses the use of predicted ingredients for retrieving recipes for food categories unknown in VIREO Food-172 dataset. We compile a list of 20 food categories as shown in Table 6 for the experiment. Each category is associated with 1 to 20 recipes. For each category, we make sure that at least its key ingredients are known to VIREO Food-172. On average, there are 3 key ingredients per category. Among the 20 categories, 4 out of them include ingredients that are not seen in VIREO Food-172. For each category, a total of 50 images are crawled from Baidu for testing. The experiment is conducted by, given a test image, the system searches against 65,284 recipes in the corpus and returns top-10 recipes. The performance is measured by top-10 hit rate, which counts the percentage of test images where the ground-truth recipes is found in the top-10 rank list.

We compare three major groups of approaches: image retrieval, ingredient matching (Eqn-7) and their combination. For image retrieval, only the pictures associated with recipes are involved. We compare the effectiveness of different features for retrieval. For VGG, FC7 feature is extracted from the model trained for ingredient recognition. Similarly for Arch-D, where the deep feature is extracted from the private layer specialized for ingredient labels. For ingredient matching, we compare the performances of single (VGG) and multi-task (Arch-D) learnings, where the ingredient prediction scores are both adjusted by CRF. Finally, late fusion is performed for Arch-D and VGG by combining the scores obtained from image retrieval and ingredient matching. Min-max normalization is employed to convert the scores into the range of [0,1]. The fusion is based on joint probability, specifically $1-(1-p_i)\times(1-p_j)$, where p_i and p_j are scores from different approaches.

	Method	*Top-10 hit rate*
Image Retrival	Gist	0.039
	Color moment	0.035
	VGG	0.439
	Arch-D	0.523
Ingredient matching	VGG	0.447
	Arch-D (without CRF)	0.462
	Arch-D	0.554
Fusion	VGG	0.464
	Arch-D	**0.570**

Table 5: Performance of zero-shot recipe retrieval.

Table 5 lists the performance of different approaches. For image retrieval, deep features perform significantly better than hand-crafted features. Our proposed model Arch-D outperforms VGG, showing the superiority of multi-task learning not only in recognition but also feature learning. For text-based ingredient matching, Arch-D also shows better performance than VGG, attributed mainly to the lower recognition error made in ingredient predic-

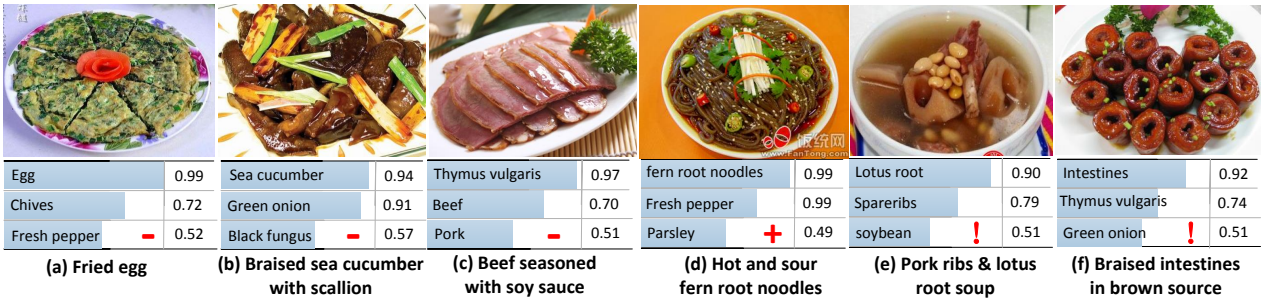


Figure 10: Example of test images showing effect of CRF in refining ingredient labels. The "-" sign means the false positives that are successfully excluded after CRF, while the "+" sign means the false negatives that are recalled by CRF. The "!" sign indicates true positives that are erroneously removed by CRF.

tion, especially after CRF refinement. Further fusion of both results from Arch-D achieves the overall best performance among all the compared approaches.

Category	Image retrival	Ingredient matching	Fusion
Assorted corn (12)	**0.92**	0.84	0.86
Braised noodles with lentil (16)	0.44	0.42	**0.46**
Braised chicken & potato (8)	**0.42**	0.34	0.34
Cucum. & fungus with eggs (2)	**0.68**	0.34	0.56
Carrot & kelp (4)	**0.78**	0.64	0.72
Cabbage & vermicelli (5)	0.30	**0.54**	0.44
Corn, carrot & ribs soup (19)	0.62	0.64	**0.70**
Dried tofu & pepper(8)	0.48	**0.80**	0.68
Griddle cooked chicken*(7)	0.36	0.40	**0.44**
Loofah egg soup (15)	0.76	**0.98**	0.92
Mustard pork noodles (10)	0.34	**0.74**	0.70
Noodles with peas & meat*(4)	**0.30**	0.22	**0.30**
Pepper & bitter gourd (7)	**0.68**	0.60	**0.68**
Ribs claypot (5)	0.16	**0.50**	0.12
Sichuan cold noodles (12)	**0.86**	0.82	0.84
Soybeans & pork leg soup (12)	0.48	0.46	**0.54**
Sausage claypot (19)	0.30	**0.66**	0.60
Spicy crab*(20)	**0.82**	0.38	0.56
Shredded chicken & pea sprouts*(1)	**0.20**	0.08	0.08
Tomato & egg noodles (18)	0.56	**0.94**	0.86

Table 6: Recipe retrieval performance on 20 unknown food category. The parenthesis indicates the number of recipes for a category. The categories containing unseen ingredients in VIREO Food-172 are indicated by "*".

Table 6 shows the detailed performance of Arch-D on 20 unknown food categories. The performance of image retrieval is influenced by the quality of pictures associated with recipes, particularly for the pictures in low resolution, having different appearances or lighting conditions than the queries. Such examples include "mustard pork noodle" and "tomato & egg noodles". On the other hand, solely matching ingredient lists is limited by the fact that the same set of ingredients can be used for different food categories. One such example is "cucumber & fungus with eggs", where the ingredients are also found in several other food categories, despite different visual appearance due to different ways of cooking and cutting. Image retrieval using the deep features, which are trained to deal with these visual variations, generally shows better performance. Fusion basically compromises both performances and produces the overall best performance. There are four categories where fusion successfully boosts the performances of both approaches. In these cases, image retrieval helps by "disambiguating" the rank lists generated by ingredient matching.

The retrieval performance is also affected by occlusion of ingredients. For example, the "chicken" in "shredded chicken & pea sprouts" is hardly visible under "pea sprouts", which is an ingredient unseen in VIREO Food-172. In this case, ingredient matching performs poorly as seen in Table 6. Image retrieval also performs unsatisfactorily due to diverse dish appearances for test images under this category. Another example is "spicy crab", where crab is hidden under other ingredients. Image retrieval, however, performs surprisingly well for this category because of the unique color and texture of the dishes. Finally, there are four categories that have unseen ingredients. Except "spicy crab", the performance of these categories is below average, showing the challenges of retrieval for recipes with unknown ingredients.

7. CONCLUSIONS

We have presented two main pieces of our work: ingredient recognition and zero-shot recipe retrieval. The former is grounded on a deep architecture (Arch-D) that exploits the joint relationship between food and ingredient labels through multi-task learning. The latter extends the knowledge of Arch-D for the out-of-vocabulary scenario, by learning contextual relationships of ingredients from a large textual corpus of recipes. Experimental results on a challenging Chinese food dataset (VIREO Food-172) show that, while the performance of food categorization is enhanced slightly, the improvement in ingredient recognition is statistically significant compared to the best single-task VGG model. The superiority in performance is not only noticed in VIREO Food-172 but also UEC Food-100, a large-scale Japanese food dataset. More importantly, when extracting the deep features (FC7) from the specialized or private layer learnt for ingredient recognition, the features show highly favorable performance for zero-shot recipe retrieval, in comparison to hand-crafted features and single-task model. The performance of ingredient recognition is also successfully enhanced with the contextual relationship modeling of ingredients and CRF. The experiment also indicates that using our proposed architecture and CRF for ingredient prediction can produce better performance than directly inferring ingredients from recipes searched by VGG. When further using the predicted ingredients for matching recipes of unknown food categories, our model also demonstrates impressive performance, including when fusing with the deep features.

While encouraging, the current work is worth further investigation in two directions. First, cooking method (e.g., frying, steaming, grilling) is not explicitly considered in the developed deep architecture. In the experiment, we notice that some dishes have the same ingredients but appear visually different mainly due to different cooking methods. Our current approach basically cannot distinguish recipes for this kind of dishes. Similarly for ways of cutting ingredients (e.g., chop, slice, mince) which may demand hierarchical way of ingredient recognition in deep network. In addition, our multi-task model could not deal with ingredients (e.g., honey, soy-

bean oil) that are not observable or visible from dishes. Secondly, while this paper considers the zero-shot problem of unknown food categories, how to couple this problem together with unseen ingredients remains unclear. Future work may include learning of embedded space that can capture the inherent "translation" between dish pictures and textual recipes, for dealing with the problem of unknown food and ingredient labels.

8. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (No. 61272290), and the National Hi-Tech Research and Development Program (863 Program) of China under Grant 2014AA015102.

9. REFERENCES

[1] K. Aizawa and M. Ogawa. Foodlog: Multimedia tool for healthcare applications. *IEEE MultiMedia*, 22(2):4–8, 2015.

[2] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar. Menu-match: Restaurant-specific food logging from images. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 844–851. IEEE, 2015.

[3] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, and I. Essa. Leveraging context to support automated food recognition in restaurants. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 580–587. IEEE, 2015.

[4] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014*, pages 446–461. Springer, 2014.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[6] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 289–292. IEEE, 2009.

[7] M.-Y. Chen, Y.-H. Yang, C.-J. Ho, S.-H. Wang, S.-M. Liu, E. Chang, C.-H. Yeh, and M. Ouhyoung. Automatic chinese food identification and quantity estimation. In *SIGGRAPH Asia 2012 Technical Briefs*, page 29. ACM, 2012.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[10] M. Elhoseiny, T. El-Gaaly, A. Bakry, and A. Elgammal. Convolutional models for joint object categorization and pose estimation. *arXiv preprint arXiv:1511.05175*, 2015.

[11] A. H. Goris, M. S. Westerterp-Plantenga, and K. R. Westerterp. Undereating and underrecording of habitual food intake in obese men: selective underreporting of fat intake. *The American journal of clinical nutrition*, 71(1):130–134, 2000.

[12] H. He, F. Kong, and J. Tan. Dietcam: Multi-view food recognition using a multi-kernel svm. *Journal of Biomedical and Health Informatics*, 2015.

[13] H. Hoashi, T. Joutou, and K. Yanai. Image recognition of 85 food categories by feature fusion. In *Multimedia (ISM), 2010 IEEE International Symposium on*, pages 296–301. IEEE, 2010.

[14] Y. Kawano and K. Yanai. Real-time mobile food recognition system. In *Computer Vision and Pattern Recognition Workshop*, 2013.

[15] Y. Kawano and K. Yanai. Food image recognition with deep convolutional features. *Proc. of ACM UbiComp Workshop on Cooking and Eating Activities (CEA)*, 2014.

[16] K. Kitamura, T. Yamasaki, and K. Aizawa. Food log by analyzing food images. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 999–1000. ACM, 2008.

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

[18] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1666–1674, 2015.

[19] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[20] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[21] D. G. Lowe. Object recognition from local scale-invariant features. *The proceedings of the seventh IEEE international conference on Computer Vision,*, pages 1150–1157, 1999.

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

[23] H. Matsunaga, K. Doman, T. Hirayama, I. Ide, D. Deguchi, and H. Murase. Tastes and textures estimation of foods based on the analysis of its ingredients list and image. In *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*, pages 326–333. Springer, 2015.

[24] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy. Im2calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1233–1241, 2015.

[25] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney. Recognition and volume estimation of food intake using a mobile device, 2009.

[26] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[27] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):300–312, 2007.

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[30] M. A. Stricker and M. Orengo. Similarity of color images. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*, pages 381–392. International Society for Optics and Photonics, 1995.

[31] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[32] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[33] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

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

[35] W. Wu and J. Yang. Fast food recognition from videos of eating for calorie estimation. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1210–1213. IEEE, 2009.

[36] K. Yanai and Y. Kawano. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015.

[37] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. *Computer Vision and Pattern Recognition (CVPR)*, pages 2249–2256, 2010.