RECIPE RECOGNITION WITH LARGE MULTIMODAL FOOD DATASET

Xin Wang (1) *Devinder Kumar* (2) *Nicolas Thome* (1) *Matthieu Cord* (1) *Frédéric Precioso* (3)

(1) Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France

(2) University of Waterloo, Vision and Image Processing (VIP) Lab, Ontario, Canada

(3) Universités Nice Sophia Antipolis, UMR 7271, I3S, F-06900, Sophia Antipolis, France

ABSTRACT

This paper deals with automatic systems for image recipe recognition. For this purpose, we compare and evaluate leading vision-based and text-based technologies on a new very large multimodal dataset (UPMC Food-101) containing about 100,000 recipes for a total of 101 food categories. Each item in this dataset is represented by one image plus textual information. We present deep experiments of recipe recognition on our dataset using visual, textual information and fusion. Additionally, we present experiments with text-based embedding technology to represent any food word in a semantical continuous space. We also compare our dataset features with a twin dataset provided by ETHZ university: we revisit their data collection protocols and carry out transfer learning schemes to highlight similarities and differences between both datasets.

Finally, we propose a real application for daily users to identify recipes. This application is a web search engine that allows any mobile device to send a query image and retrieve the most relevant recipes in our dataset.

1. INTRODUCTION

Food category classification is a key technology for many food-related applications such as monitoring healthy diet, computational cooking, food recommendation system, etc. In [1], a novel smart phone application to record daily meal activities by image retrieval technique is developed. Based on this personal dietary data log system, they were able to conduct further usage preference experiments [2] and food nutrition balance estimation [3].

Open Food System [2] aims at inventing new smart cooking appliances, with the ability to monitor cooking settings automatically for optimal results and preserve the nutritional value and organoleptic qualities of cooked foods. The Technology Assisted Dietary Assessment (TADA) project of Purdue University [4] aims at developing a mobile food recorder which can translate dietary information to an accurate account of daily food and nutrient intake. Food category classification is an indispensable ingredient in all these applications.

[2] http://www.futur-en-seine.fr/fens2014/en/projet/open-food-system-2/

In this paper, we focus on building automatic systems for image recipe recognition. For this purpose, we propose a new very large multimodal dataset (UPMC Food-101) containing about 100,000 recipes for a total of 101 food categories collected from the web. Each item in this dataset is represented by one image and the HTML information including metadata, content etc. of the seed page from which the image originated. We detail our initiative to build our dataset in sections 2 and 3 explaining the specificities and the originality of our dataset. We perform experiments at a large scale to evaluate visual and textual features along with their fusion in section 4. We propose in section 5, further statistics to highlight dataset characteristics and comparison with another recent large scale dataset (ETHZ Food-101 [5]). Finally, in section 6, we demonstrate the interest of these recognition technologies coupled with web-based dataset in a mobile search application, which can receive food image as a query and return the most relevant classes and corresponding recipes.

2. RELATED WORKS ON FOOD DATASETS

There is an increasing demand of food category data in various food-related applications like dietary assessment, computational cooking, recipe retrieval, etc. However, specific public massive dataset for food research community is still insufficient. One of them is the Pittsburgh Food Image Dataset (PFID) [6] dataset of 4556 fast food images. Another one is UNICT-FD889 dataset [7] that has 889 distinct plates of food. The authors use this database for Near Duplicate Image retrieval (NDIR) by using three different state-of-the-art image descriptors. There are a couple of databases from the Max-Planck Institute for Informatics that contain images of cooking activities which focus on detecting fine grained activities while cooking [8]. UEC-Food100 [9] contains 100 categories of food images, each category contains about 100 images, mainly Japanese food categories. [10] performs a late fusion of deep convolutional features and conventional handcrafted image features upon dataset UEC-Food100 [9], which outperforms the best classification accuracy on this dataset. Most of the datasets are either collected in a controlled environment or contain too few samples for each food category to build a generic food recognizer or classifier.

In this paper, we propose a new very large multimodal dataset henceforth named as UPMC Food-101, which is collected in uncontrolled environment with a huge diversity among the instances. UPMC Food-101 contains about 100,000 images and textual descriptions for 101 food categories. This dataset aims at stimulating the research of food category recognition in the domain of multimedia and it will be released freely to the research community. In addition to the large number of images, an extra property of our dataset is that it shares the same food categories with one of the largest public food image dataset ETHZ Food-101 [5]. Such a "twins dataset pair" can thus enable many interesting research hot spots such as transfer learning. To the best of our knowledge, UPMC Food-101 and ETHZ Food-101 are the first "twins dataset pair" in food area. We explain the similarities and differences between both datasets in detail in the following sections.

3. UPMC FOOD-101 DATASET

3.1. Data Collection Protocol

To create a real world and challenging dataset (with both visual and textual data) which can truly represent the existing large intra-class variations among food categories, we decide to use Google Image search. Unlike controlled sources, using long ranking (one thousand results) of Web search engine allows to explore recipes that are potentially deeply buried in the world wide web. Similarly, Y. Kawano and K. Yanai explore the Web resources in [11], to extend their initial UEC Food-100 dataset [9]. It is also interesting to note that the past approaches [12] using Google search engine to obtain images for classification tasks have reported around 30 percent of precision level on some of collected images (in 2006). We observe that the results returned by Google Image search in 2014 for textual queries related to food images are more relevant with very low level of noise. This is explained by the large improvement in the field of searching and page ranking algorithms since 2006. Based on these preliminary findings, we decide to create our database by querying Google image search with 101 labels taken from the ETHZ Food-101 dataset [5] along with an added word "recipes". We added the word "recipes" to each label before passing the query to Google for two reasons:

- As we are interested in recipe recognition, adding "recipes" word after the labels, for example, "hamburger recipe", returns more focused information about "how to make hamburgers" rather than other topics like "where to eat hamburgers" or "Hamburger is junk food" in the textual form.
- We observed that adding "recipes" to our queries helps decreasing the noise level a little further in the returned images. For example, a simple "hamburger" in search engine could return some thing like "hamburger menu icon" or "hamburger-like evening dress" which are far from our expectations.

We then collect the first $1,000$ images returned for each query and remove any image with a size smaller than $120\ pixels$. In total, UPMC Food-101 contains 101 food categories and $90,840$ images, with a size range between 790 and 956 images for different classes. Figure 1 shows representative instances of all 100 categories. Due to no human intervention in grasping these data, we estimate that each category may contain about 5% irrelevant images for each category. 3 examples of "hamburger" class are shown in Figure 2. We notice that adding the keyword "recipes" results in taking into account ingredient or intermediate food images. Determining whether these images should be considered as noise or not, directly depends on the specific application. Additionally, we save $93,533$ raw HTML source pages which embed images. The reason that we don't have $101,000$ HTML pages is that some pages are not available. The number of the images that have text is $86,574$.

Fig. 1: Category examples of our UPMC Food-101 dataset.

Dataset	class num	image num per class	source	Data type
UPMC	101	790 - 956	various	text&image
ETHZ	101	1000	specific	image

Table 1: UPMC Food-101 and ETHZ Food-101 statistics

(a) Correct (b) Ingredient (c) Noise

Fig. 2: Example images within class "hamburger" of UPMC Food-101. Note that we have images completely irrelevant with hamburger like Figure 2c, as well as hamburger ingredient like Figure 2b, which depends on the specific application to judge whether it is noise or not.

Fig. 3: Example images within class "hamburger" of ETHZ Food-101. All these images have strong selfie style as they are uploaded by consumers. Although some background noise (human faces, hands) are introduced in images, it ensures images out of food categories are excluded from this dataset.

3.2. Comparison with ETHZ Food-101

The food dataset ETHZ Food-101 [5] has been recently introduced. 101,000 images for 101 food categories have been collected from a specific website (*e.g.* www.foodspotting.com). The labels of food categories were chosen from the top 101 most popular dishes on the mentioned website.

We have used the same class labels as ETHZ Food-101 for our dataset. In Table 1, general statistics on both sets are reported. The main difference comes from the data collection protocols. Since our data is collected directly from a search engine with automatic annotations, whereas ETHZ Food-101 dataset images were collected from a specific website which contains manual annotated images uploaded by humans, leading to less number of false positive/noise in ETHZ Food-101 than in UPMC Food-101. As the three examples of "hamburger" class show in Figure 3, ETHZ Food-101 ensures images irrelevant with food categories are mostly excluded from this dataset. Moreover, there was no textual data provided with images in ETHZ Food-101. However, to classify between two variants of the same food categories, text can help a lot. We explore visual and text classification in the next section.

4. CLASSIFICATION RESULTS

In the following subsections we run several classification algorithms by using visual information, textual information and the fusion, to make quantitative descriptions of our dataset. The results are shown in Table 2. A unified training and test protocol is applied for both visual and textual tests, in order to evaluate and compare the performances with minimal extra factors. The protocol is as follows: we split out the examples which have both image and text, then randomly select 600 training examples for each category to train a one-vs-rest linear SVM [13] with $C = 100$, the remaining examples are for test. We evaluate our results by averaging accuracy over 10 tests, where accuracy is defined as $\frac{\#(true\ positives)}{\#(test\ examples)}$.

4.1. Visual Features Experiments

4.1.1. Bag-of-Words Histogram (BoW) + SIFT

We represent images as Bag-of-Words histogram with a spatial pyramid as our first baseline. In detail, we first proportionally resize images which has a size larger than 300 $pixels$, then extract mono-scale SIFT with window size 4 and step size 8, 1024 word visual dictionary, soft coding and max pooling with 3 level spatial information. This baseline obtains an average accuracy 23.96%.

4.1.2. Bossanova Image Pooling Representation

Bossanova [14] reinforces the pooling stage of BoW by considering distance between a word and a given center of a cluster. As Bossanova only modifies the pooling stage, we can reuse the same coding setting as BoW. In our experiment, 2 bins are used in the quantization step to encode the distances from sifts to clusters, BoW is concatenated with vector histogram with no scaling factor, we set range of distances per cluster to $[0.4, 2.0]$, for each word we consider 10 neighbors. This method results in an average accuracy of 28.59%, which constitutes an improvement of 19.37% over the BoW model.

4.1.3. Convolutional Neural Networks (CNN) Deep Features

CNN deep feature is the state of the art in many image recognition challenges. Deep feature contains color information, nevertheless, its dimensionality is often much lower than the traditional SIFT descriptor but with much better performance. In our experiment, we first adopt the "fast network" pretrained model of OverFeat[3] as the feature extractor. The output of the $7th$ stage of this model, a 4096 dimension vector, is used as the feature description of a given image. We get an average accuracy of 33.91%, which gains an relative improvement of 18.6% with respect to the Bossanova.

This result is very interesting because the OverFeat CNN was trained on 1,000 class dataset ImageNet, which contains very few images of food categories (French fries, few images of waffles etc). Even after having been trained on few food images, the OverFeat CNN produces very good deep features

[3] http://cilvr.nyu.edu/doku.php?id=software:overfeat:start

Visual				Textual	Fusion
BoW	Bossanova	Deep	Very Deep	TF-IDF	TF-IDF + Very Deep
23.96%	28.59%	33.91%	40.21%	82.06%	85.10%

Table 2: Classification results (Ave. Precision %) on UPMC Food-101 for Visual, Textual and Combined features.

which outperform the standard Bossanova baseline in the context of classification.

[15] pushes CNN network to $16-19$ weight layers, which is about twice deeper than the previous work. In our experiment, we use the pre-trained model "imagenet-vgg-verydeep-19"[4] to extract features. This model is also trained on ImageNet so it is comparable with the result in the last paragraph. We take the output of the last layer before the classification layer as image features. Each feature is a 4096 dimensions vector. We finally achieve an accuracy of 40.21% over our dataset with very deep features.

4.2. Textual Features Experiment

Since our raw textual data is in html format, we need some preprocessing in order to remove numerous noisy elements such as html tags, code, punctuations. Our foremost preprocessing is parsing content out from HTML pages by Python package html2text[5].

4.2.1. TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) value measures the importance of a word w in a document D with respect to the whole corpus, where TF evaluates the importance of word in a document, and IDF evaluates the importance of a word in the corpus.

To represent a document with TF-IDF, we generate the dictionary by preprocessing words as follows: 1/ Stemming all words. For example, words like "dogs" and "sleeping" are respectively stemmed to "dog" and "work", 2/ Removing words with high frequency of occurrence (stop words) such as "the","is","in", 3/ Removing words occurred less than in 11 docs, 4/ Keeping stems with length between 6 and 18. After the pre-processing, 46972 words are left. We then form a dictionary $Dict_t$ using these words.

We calculate TF-IDF value for every word in document by formula $tfidf_{w,D} = tf_{w,D} \times idf_w$, with $tf_{w,D} = \frac{n_{w,D}}{\sum_k n_{k,D}}$, where $n_{i,j}$ is the frequency of word i appearing in document j, and $idf_w = \log \frac{|N|}{|\{j:w \in D_j\}|}$, where N is the total number of documents in the corpus, and $|\{j : w \in D_j\}|$ is the number of documents where the term w appears. TF-IDF value favors the words less occurred in corpus and more occurred in a given document D, and suppress the word in reverse case. A document can be represented by the TF-IDF value of all its words belonging to the dictionary $Dict_t$. We obtain 82.06% classification average accuracy on our dataset. Such a high score is partly due to the bias introduced by our data crawling protocol.

[4] http://www.vlfeat.org/matconvnet/pretrained/

[5] https://pypi.python.org/pypi/html2text

4.3. Late Fusion of Image+Text

We merge very deep features and TF-IDF classification scores by late fusion. The fusion score s_f is a linear combination of the scores provided by both image and text classification systems, as $s_f = \alpha s_i + (1-\alpha) s_t$, where α is the fusion parameter in the range $[0, 1]$, s_i is the score from the image classifier and s_t is the score from the text. We select α by cross-validation over different splits of data and the final classification score is 85.1%, which improves 3.6% with respect to textual information alone and 109.8% with respect to visual information alone. Note that the classification scores were not calibrated prior to late fusion so that α does not depend on the relative accuracy of each source of scores.

5. QUANTITATIVE ANALYSIS OF UPMC FOOD-101

In this section, we report further analysis of UPMC Food-101. We investigate the word vector representations [16] for its strong semantic expressiveness. Transfer learning between UPMC Food-101 and ETHZ Food-101 is also analyzed.

5.1. Word Vector Representation

We first introduce how to extract word vectors, then explore some interesting features of this representation.

After parsing out the content of web pages, we concatenate all of them together to build a corpus for training a dictionary $Dict_v$ with word2vec [16], which is a tool to efficiently compute vector representations of words. Words with an occurrence frequency less than 5 in the corpus are removed from $Dict_v$. This condition results in 137092 words, in which each word is described by a 200 dimensional feature vector. $Dict_v$ contains stop words and other noisy words, so we intersect $Dict_t$ and $Dict_v$, which creates a new dictionary $Dict$ containing 46773 words.

On the other hand, each document is first preprocessed by the tool html2text, then represented by the element-wise average of its valid word vectors, where "valid" means that the word is in $Dict$. A linear SVM is trained and we obtain an average accuracy of 67.21% on our dataset. Although this classification result is worse than TF-IDF (82.06%), it can be enhanced by more advanced pooling strategies, rather than

TF-IDF	word2vec	TF-IDF+word2vec
82.06%	67.21%	84.19%

Table 3: Late fusion of TF-IDF and average word2vec representations.

a simple average vector over all words, as reported in [17]. Additionally, recall that our data source is the Google search results according to a category name: this step can also reinforce the superiority for word frequency based methods like TF-IDF. On the other hand, since the word vector tries to learn a semantic representation of words with much less dimension, the simple word frequency statistical information will surely lose a lot. However, by late fusion with TF-IDF, we get the score of 84.19%, improving by 2% the single TF-IDF performance, as shown in Table 3. TF-IDF and word2vect encode complementary information in textual data.

The embedded word vector space allows to explore semantic relationships. To investigate this aspect, we report in Table 4 the closest words by using the cosine distance metric for -ravioli, -sushi, -pho in the embedded vector space (using the $Dict_v$ dataset). The five most closest words are strongly semantically related to the given query. Additionally, calculating a simple average of the words in a phrase also results in a reasonable semantic. In Table 5, we show the closest words of -rice, -japan and -rice japan. As we can see, -koshihikari, which is a popular variety of rice cultivated in Japan, is closest to -rice japan, meanwhile for either -rice or -japan, -koshihikari- is out of their first five candidates, which means word vector has well expressed the semantic of the short phrase -rice+japan. Moreover, -koshihikari is not among the 101 food category, its meaning and relation with other words are all learned from the corpus in a purely unsupervised manner. Such a powerful semantic understanding property could help search engine understand user-level needs with natural language as input. It is a promising tool for filling the semantic gap.

ravioli	sushi	pho
gnocchi 0.67	nigiri 0.69	souppho 0.68
tortelli 0.58	maki 0.65	vietnames 0.59
cappellacci 0.55	uramaki 0.65	phos 0.57
delallocom 0.52	sashimi 0.64	beefnoodl 0.58
itemtitlea 0.52	norimaki 0.64	bo 0.56

Table 4: 5 most similar words of -ravioli, -sushi and -pho. Each group is indeed semantically relevant, except for some words with low scores like -delallocom and -itemtitlea.

5.2. Transfer Learning

As another set of experiments, we perform knowledge transferring experiments over both datasets (ETHZ Food-101 and UPMC Food-101), namely learning the classifier model on one dataset and testing it on the other one. This experiment aims at showing the different performances of UPMC Food-101 and ETHZ Food-101 when performing visual classification. In this experiment, we use very deep features. The results of the transfer learning experiments are shown in Table 6. The first two rows show the results of classification when training with the same number of examples (*e.g.* 600 examples for each class) of one dataset and testing on the rest of this dataset or on the whole of the other dataset, while the last two rows show the results of classification when training with all examples on one dataset and testing on the other dataset.

rice	japan	rice japan
calros 0.59	osaka 0.70	**koshihikari 0.64**
basmati 0.59	tokyo 0.62	awabi 0.61
vermicelli 0.58	kyoto 0.62	japanes 0.61
stirfri 0.58	chugoku 0.61	nishiki 0.59
veget 0.58	gunma 0.60	chahan 0.57

Table 5: Short phrase -rice japan, represented as the average of -rice and -japan, is closest to -koshihikari.

There are some interesting points that can be inferred from the results. The first one is that even though both datasets contain images for same food categories, they are very different from each other. This can be derived from the fact that there is a considerable difference of around 50% average accuracy when training on one dataset and testing on both datasets (first 2 rows in Table 6).

Second point that can be observed from the Table 6 is that training on part of UPMC Food-101 outperforms training on the whole UPMC Food-101 when testing on ETHZ Food-101 by a margin of 1.57%, while on the contrary, only a negligible difference (0.36%) for training on ETHZ Food-101 and testing on UPMC Food-101 is observed. This perhaps can be an indication of comparative noise levels in both datasets, UPMC Food-101 being the noisier dataset.

train / test	UPMC	ETHZ
UPMC (600 examples)	40.56	25.63
ETHZ (600 examples)	25.28	42.54
UPMC (all examples)	-	24.06
ETHZ (all examples)	24.92	-

Table 6: Results of transfer learning between UPMC Food-101 and ETHZ Food-101.

Note that our ETHZ deep results are not comparable with the CNN results in [5] because they train deep features as we use a pre-trained CNN on ImageNet.

6. MOBILE RECIPE RETRIEVAL

Providing an efficient way to automatically recognize the food/dish or its recipes on our plates will not only satisfy our curiosity but can have a wider impact on daily life in both the real and virtual worlds. "What is the name of this dish?", "How to cook this?". We all have asked these questions to a chef or friends. As a proof of concept, we have created a web search engine [5] that allows any mobile device to send a query image and to get answers to our questions. For any query image, the result is a ranking of the 3 best categories automatically found with a matching score that may at least indicate if the match is correct (positive) or not (negative score). For each selected category, images related to the query are displayed with the hyperlink to the recipe webpage available. Figure 4 presents the answer to a query image (representing a pizza) displayed at the top of the page. The categories predicted with the highest scores are returned on the next three lines, followed by seven clickable images (by category) linked to the original recipe webpages. In this case, only the correct result -pizza gets a positive score (top ranking).

Fig. 4: Results for a pizza image

7. CONCLUSION

In this paper, we introduce a large multimedia dataset with 101 food categories. We present an extended evaluation of BoVW, Bossanova and deep features for food image recognition, as well as TF-IDF for document classification. Our experiments suggest that for visual recognition, CNN deep feature is the best step forward. Due to the manner of collecting data, a strong bias makes bag-of-textual-words perform better than any other single method. Nevertheless, the fusion of visual and textual information achieves better average precision 85.1%. Additionally, we find that word vector shows powerful ability in representing any word in a semantical food continuous space. We also run complementary experiments to highlight differences and complementarity of our UPMC Food-101 dataset with the recently published ETHZ Food-101 dataset. Based on our dataset, we have proposed a retrieval system that we plan to improve using machine learning techniques [18, 19, 20] for user interaction.

[5] Available at http://visiir.lip6.fr/.

8. REFERENCES

[1] K. Aizawa, "Multimedia foodlog: Diverse applications from self-monitoring to social contributions," *ITE Transactions on Media Technology and Applications*, 2013.

[2] K. Aizawa and et al., "Comparative Study of the Routine Daily Usability of FoodLog: A Smartphone-based Food Recording Tool Assisted by Image Retrieval," *Journal of diabetes science and technology*, 2014.

[3] 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*, 2013.

[4] N. Khanna and et al., "An overview of the technology assisted dietary assessment project at purdue university.," in *ISM*, 2010.

[5] L. Bossard and et al., "Food-101 Mining Discriminative Components with Random Forests," in *ECCV*, 2014.

[6] M Chen and et al., "PFID: Pittsburgh fast-food image dataset," in *ICIP*, 2009.

[7] GM Farinella and et all, "A Benchmark Dataset to Study Representation of Food Images," in *ACVR*, 2014.

[8] Marcus Rohrbach and S Amin, "A database for fine grained activity detection of cooking activities," in *CVPR*, 2012.

[9] Y. Kawano and K. Yanai, "FoodCam: A Real-Time Mobile Food Recognition System Employing Fisher Vector," in *MMM*, 2014.

[10] Yoshiyuki Kawano and Keiji Yanai, "Food image recognition with deep convolutional features," in *ACM UbiComp*, 2014.

[11] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. of ECCV Workshop on TASK-CV*, 2014.

[12] A. Schroff, F. and Criminisi, A. and Zisserman, "Harvesting image databases from the web," *PAMI*, 2011.

[13] R. Fan and et al., "LIBLINEAR: A library for large linear classification," *JMLR*, 2008.

[14] S. Avila and et al., "Pooling in image representation: The visual codeword point of view," *CVIU*, 2013.

[15] K Simonyan and A Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014.

[16] T. Mikolov and et all, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[17] Quoc Le and Tomas Mikolov, "Distributed Representations of Sentences and Documents," in *ICML*, 2014.

[18] D. Gorisse, M. Cord, and F. Precioso, "Salsas: Sub-linear active learning strategy with approximate k-nn search," *Pattern Recognition*, vol. 44, no. 10, pp. 2343–2357, 2011.

[19] PH Gosselin and M Cord, "Active learning methods for interactive image retrieval," *Image Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1200–1211, 2008.

[20] D. Picard, M. Cord, and A. Revel, "Image retrieval over networks: Active learning using ant algorithm," *Multimedia, IEEE Transactions on*, vol. 10, no. 7, pp. 1356–1365, 2008.