FOOD IMAGE RECOGNITION USING DEEP CONVOLUTIONAL NETWORK WITH PRE-TRAINING AND FINE-TUNING

Keiji Yanai   *Yoshiyuki Kawano*

Department of Informatics, The University of Electro-Communications, Tokyo, Japan
{yanai,kawano-y}@mm.inf.uec.ac.jp

ABSTRACT

In this paper, we examined the effectiveness of deep convolutional neural network (DCNN) for food photo recognition task. Food recognition is a kind of fine-grained visual recognition which is relatively harder problem than conventional image recognition. To tackle this problem, we sought the best combination of DCNN-related techniques such as pre-training with the large-scale ImageNet data, fine-tuning and activation features extracted from the pre-trained DCNN. From the experiments, we concluded the fine-tuned DCNN which was pre-trained with 2000 categories in the ImageNet including 1000 food-related categories was the best method, which achieved 78.77% as the top-1 accuracy for UEC-FOOD100 and 67.57% for UEC-FOOD256, both of which were the best results so far.

In addition, we applied the food classifier employing the best combination of the DCNN techniques to Twitter photo data. We have achieved the great improvements on food photo mining in terms of both the number of food photos and accuracy. In addition to its high classification accuracy, we found that DCNN was very suitable for large-scale image data, since it takes only 0.03 seconds to classify one food photo with GPU.

***Index Terms*—** deep convolutional neural network food recognition Twitter photo mining

1. INTRODUCTION

Food image recognition is one of the promising applications of visual object recognition, since it will help estimate food calories and analyze people's eating habits for health-care. Therefore, many works have been published so far [1, 2, 3, 4, 5, 6, 7]. To make food recognition more practical, increase of the number of recognizable food is crucial. In [4, 3], we created 100-class food dataset, UEC-FOOD100, and made experiments with 100-class food classification. The classification accuracy reported so far was 72.26% [8], which still needs to be improved for practical use. Moreover, recently we proposed a framework to extend an existing dataset automatically [9], and with it we extended a 100-class food dataset into a 256-class food dataset called as UEC-FOOD256. To make food classification more practical, we need more sophisticated food image classifiers.

Meanwhile, recently the effectiveness of Deep Convolutional Neural Network (DCNN) have been proved for large-scale object recognition at ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012. Krizhevsky et al. [10] won ILSVRC2012 with a large margin to all the other teams who employed a conventional hand-crafted feature approach. In the DCNN approach, an input data of DCNN is a resized image, and the output is a class-label probability. That is, DCNN includes all the object recognition steps such as local feature extraction, feature coding, and learning. In general, the advantage of DCNN is that it can estimate optimal feature representations for datasets adaptively [10], the characteristics of which the conventional hand-crafted feature approach do not have. In the conventional approach, we extract local features such as SIFT and SURF first, and then code them into bag-of-feature or Fisher Vector representations.

To train a DCNN directly, we need a large-scale image data such as the ILSVRC dataset which contains more than one million images. If a large-scale training data is always needed, applicable problems of a DCNN is very limited. To avoid such situation and to make a DCNN effective even for small-scale data, two important techniques have been proposed so far.

The first one is using a pre-trained DCNN with a large-scale dataset such as the ILSVRC dataset as a feature vector extractor for a small-scale data. By extracting activation signals from the intermediate layer of the DCNN after an image is provided into the first layer of the pre-trained DCNN and its signals are propagated into the upper layers, the extracted signal can be regarded as image features. This DCNN features are commonly extracted from the output signals of the previous layer of the last one in the pre-trained DCNN. Donahue et al. [11] confirmed the effectiveness of DCNN features with Caltech-101 [12] and SUN-397 database [13]. Chatfield et al. made comprehensive experiments employing both DCNN features and conventional features such as SIFT and Fisher Vectors on PASCAL VOC 2007 and Caltech-101/256 which can be regarded as small-scale datasets where they had only about one hundred or less images per class [14]. DCNN features have been proved to be effective not only for image classification but also image retrieval [15] and specific object recognition tasks [16]. In this case, a DCNN can be used for only a feature extractor, and a linear SVM is commonly used as a classifier. By using a SVM as a classifier, it is easy to fuse other kinds of image features such as bag-of-features representation and Fisher vector representation.

The second technique is fine-tuning of the pre-trained DCNN. “Fine-tuning” is tuning the parameters pre-trained with a large-scale data using another small-scale data. With fine-tuning, the DCNN originally for a large-scale data is modified and adapted to other tasks. In this case, a DCNN is used as both a feature extractor and a classifier. That is, a classification result is obtained directly from the output layer of a DCNN. In general, the data for pre-training is not always needed to be related to the training data for fine-tuning. However, Oquab et al. [17] showed that it is important that the pre-training data is strongly related to the fine-tuning data for better performance. For fine-tuning for PASCAL VOC categories, they selected 512 additional categories related to twenty VOC categories such as bus bicycle and bird from the whole ImageNet database, and pre-trained a DCNN with 1512 ImageNet categories which consisted of the ILSVRC 1000 categories and the selected 512 categories.

Both techniques can be regarded as a kinds of “transfer learning” which trains a classifier with source domain data and adapts it with target domain data.

So far, Kawano et al. [8], Kagaya et al. [2] and Bossard et al. [1] have applied DCNN to food image classification. Kawano et al.[8] used DCNN features pre-trained with the ILSVRC data for food classification. However, they failed to confirm that the single DCNN-based method outperformed the conventional methods which was based on Fisher Vector [18]. They confirmed only that DCNN features improved the performance by integrating them with Fisher vectors.

Kagaya et al. [2] trained a DCNN from scratch without employing pre-training and fine-tuning. They prepared 170,000 images of 10 food categories. They proposed several kinds of DCNN architectures which are smaller than Alexnet [10] and compared them with conventional methods such as SPM-BoF except for Fisher vector. As results, DCNN outperformed conventional methods greatly.

Bossard et al. [1] trained the same DCNN as Alexnet except for the size of the output layer by Caffe [19] with the Food-101 dataset which contained one million food photos of 101 categories. They also trained DCNN from scratch. DCNN outperformed all the methods including the proposed method in [1].

Regarding food datasets, the effectiveness of DCNN has not been explored enough, because in the above-mentioned works only training DCNNs from scratch with a large-scale food photo data and usage of DCNN features pre-trained with the ILSVRC 1000 categories were examined, and no fine-tuning and pre-trained with augmented dataset have not been explored yet. Then, in this paper, we apply DCNN features and fine-tuning with extended training data for 100/256-class food dataset and examine the effectiveness of DCNN features and fine-tuning for food photos.

In addition, as an application of the obtained high-performance DCNN food classifier, we applied the food classifier employing the best combination of the DCNN techniques to Twitter photo data [20].

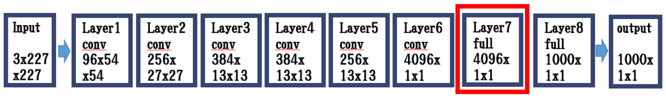

Fig. 1. DCNN structure for pre-training ImageNet Challenge Dataset in Caffe which is based on Krizhevsky’s network.

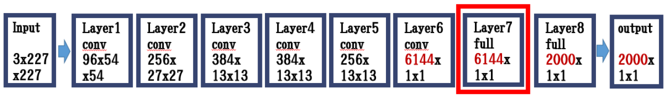

Fig. 2. DCNN structure for pre-training ImageNet 2000 categories. The number of cells in the full connection layers is modified from 4096 to 6144.

2. METHODS

2.1. DCNN Features with ILSVRC2012

Recently, it has been proved that Deep Convolutional Neural Network (DCNN) is very effective for large-scale object recognition. However, it needs a lot of training images. In fact, one of the reasons why DCNN won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012 is that the ILSVRC dataset contains one thousand training images per category [10]. This situation does not fit food datasets most of which have only about one hundred images a food category. Then, to make the best use of DCNN for food recognition, we need to use the pre-trained DCNN with the ILSVRC 1000-class dataset as a feature extractor, or fine-tune the pre-trained DCNN.

Following Donahue et al. [11], we extract the network signals from the previous layer (layer 7) of the last one of the pre-trained DCNN as a DCNN feature vector. We use the pre-trained deep convolutional neural network in Caffe [19] as shown in Figure 1. This is slight modification of the network structure (AlexNet) proposed by Krizhevsky et al. [10] where the number of elements in the last layer is the same as the number of the classes, 1000, and the number of elements in the full connection layer before the last one is 4096. Therefore, we extract a 4096-dim DCNN vector from the layer 7, and L2-normalize them to use it as a 4096-dim DCNN feature vector.

2.2. DCNN Features with ImageNet 2000 Categories

To improve DCNN features, Oquab et al. [17] indicated that it was effective to pre-train DCNN with the dataset augmented with additional categories related to the target recognition task. They added ImageNet categories related to the PASCAL VOC categories such as furniture and motor-vehicle, and obtained improved results.

Since in our case the targets are foods, we selected 1000 food-related categories from ImageNet 21,000 categories and added them with the ILSVRC 1000 ImageNet categories for pre-training of DCNN. To select additional 1000 food-related categories, first we examined the WordNet hierarchy and extracted all the subordinate concept nodes (synsets) under "food". Then, we obtained 2714 synsets. Among them, only 1526 synsets have been listed in the ImageNet database [1] and we selected the top 1000 synsets in descending order of the number of images assigned to each synset. By adding them to 1000 ImageNet categories in the ILSCRV2012 dataset, we prepared 2000 ImageNet categories enhanced with 1000 kinds of food-related categories.

We pre-trained a DCNN with this 2000 categories in ImageNet using Caffe [19]. The DCNN used for training 2000 categories is slightly modified from the DCNN for 1000 categories as shown in Figure 2. The number of the elements in the output layer is 2000, and the number of the elements in the full connection layers is 6144. We trained a DCNN with Caffe on a GPU workstation equipped with NVidia GeForce TITAN BLACK with 6GB memory, which took about one week.

In the same way as the DCNN pre-trained with 1000 categories, we extract a 6144-dim vector from the layer 7, and L2-normalize it for using it as a 6144-dim DCNN feature vector.

2.3. Fine-Tuning

We fine-tuned both the DCNN pre-trained with 1000 categories and the DCNN pre-trained with 2000 categories using Caffe [19]. We change the size of the last output layer to the same number as the number of the food categories. In the experiments, we set the number of food categories as 100 and 256. In addition, for Twitter food photo mining, we add a non-food category to 100 food categories and fine-tuned the DCNN as a 101-class classifier as well.

2.4. Baseline Features

As conventional baseline features, we extract RootHoG patches and color patches, and code them into Fisher Vector (FV) representation with Spatial Pyramid with three levels (1x1+3x1+2x2). Fisher Vector is known as a state-of-the-art coding method [18].

RootHoG is an element-wise square root of the L1 normalized HOG, which is inspired by "RootSIFT" [21]. The HOG we use consists of 2×2 blocks (totally four blocks). We extract gradient histogram regarding eight orientations from each block. The total dimension of a HOG Patch feature is 32. After extraction of HOG patches, we convert each of them into a "RootHOG". As color patches, we extract mean and variance values of RGB value of pixels from each of 2×2 blocks. Totally, we extract 24-dim Color Patch features. After extracting RootHoG patches and color patches, we apply PCA and code them into Fisher Vectors (FV) with the GMM consisting of 64 Gaussians. As results, we obtain a 32768-dim RootHOG FV and a 24576-dim Color FV for each image. This setting is almost the same as [3] except for the number of spatial pyramid levels.

2.5. Classifiers for DCNN features and Baseline features

We use one-vs-rest linear classifiers for 100/256-class food classification for DCNN activation features and baseline features. In addition to classification employing only single features, we integrate both DCNN and conventional baseline features as well. For integrating both features, we adopt late fusion with uniform weights. For lower-dimensional DCNN features, we use a standard linear SVM, while for higher-dimensional FV features, we use an online learning method, AROW [22]. As their implementations, we use LIBLINEAR [2] and AROWPP [3].

3. EXPERIMENTS

As a food dataset for the experiments, we use Japanese food image datasets, the UEC-FOOD100 dataset [4, 3] and the UEC-FOOD256 dataset [9] which are an open 100/256-class food image dataset [4]. Both include more than 100 images for each category and bounding box information which indicates food location within each food photo. We extract features from the regions inside the given bounding boxes following [3]. We evaluate the classification accuracy within the top N candidates employing 5-fold cross validation.

Figure 3 and Figure 4 show the classification accuracy of UEC-FOOD100 and UEC-FOOD256 within the top-N candidates with each of single features, RootHOG FV, Color FV, DCNN and DCNN-FOOD (DCNN pre-trained with 2000 categories), the combination of RootHoG and Color FV (written as 'FV'), the combination of FV and DCNN and FV and DCNN-FOOD, DCNN(ft) (fine-tuned DCNN) and DCNN-FOOD(ft) (fine-tuned DCNN-FOOD). The numeric values in classification accuracy for top-1 and top-5 are shown in Table 1. Table 1 contains DCNN(ft2) and DCNN-FOOD(ft2) which are not shown is the figures. Both of them represent the results by the DCNN fine-tuned with the augmented UEC-FOOD100 we created by adding at most 1000 food photos mined from Twitter to each of the 100 categories.

Regarding UEC-FOOD100 classification, among the three single features, DCNN, RootHoG-FV, and Color-FV, the DCNN feature achieved the best performance, 57.87%, in the top-1 accuracy, while RootHoG-FV and Color-FV achieved 50.14% and 53.04%, respectively. Although the combination of both FVs achieved 65.32% which was better than single DCNN features, the total dimension of the FV combination was 57,344, which 14 times as larger as the dimension of DCNN features. In addition, DCNN-FOOD outperformed FV combination and DCNN greatly. By adding 1000 food-related categories for pre-training of DCNN, the top-1 classification rate for UEC-FOOD100 was improved by 14.39 points.

The combination of FV and DCNN achieved 72.26% in the top-1 accuracy which was almost comparable to the accuracy by the single DCNN-FOOD, while the combination of

[1] ImageNet 2011 Fall release which has 21841 synsets.

[2] http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

[3] https://code.google.com/p/arowpp/

[4] http://foodcam.mobi/dataset/

Table 1. Food classification rates on UEC-FOOD 100 and 256 with baseline features and DCNN-based features.

	UEC-FOOD100		UEC-FOOD256	
features	top-1 rate	top-5 rate	top-1 rate	top-5 rate
Color FV	53.04	77.32	41.60	64.00
RootHOG FV	50.14	75.63	36.46	58.83
FV (Color+HOG)	65.32	86.70	52.85	75.51
DCNN	57.87	83.73	43.98	71.29
DCNN-FOOD	71.80	92.75	58.81	83.24
FV + DCNN	72.26	92.00	59.06	82.30
FV + DCNN-FOOD	77.35	94.85	63.77	85.82
DCNN(ft)	75.25	93.19	63.64	86.01
DCNN-FOOD(ft)	**78.48**	**94.85**	**67.57**	**88.97**
DCNN(ft2)	76.68	94.40	——	——
DCNN-FOOD(ft2)	**78.77**	**95.15**	——	——

Table 2. Classification rate on ETH Food-101 dataset [1].

RF-based [1]	DCNN [1]	DCNN(ft)	FOOD-DCNN(ft)
50.76	56.40	68.44	70.41

FV and DCNN-FOOD achieved 77.35%.

Regarding the results with fine-tuned DCNNs, although DCNN(ft) was beaten by FV+DCNN-FOOD, DCNN-FOOD(ft), 78.48%, outperformed all the combinations and single features, which was the best performance for UEC-FOOD100 so far. The difference, 3.23, between DCNN(ft) and DCNN-FOOD(ft) is not as large as the difference, 13.97, between DCNN and DCNN-FOOD, because both are fine-tuned with the food data set.

In case of using the augmented UEC-FOOD100 which contains at most 1000 images for each of the food categories for fine-tuning, the obtained improvement was very limited for both DCNN(ft2) and DCNN-FOOD(ft2). This shows the effectiveness of fine-tuning for a small-scale data.

Regarding UEC-FOOD256 classification, we observed the similar tendency. Especially, in case of larger number of food categories, the accuracy by a single DCNN was poor which are comparable to a single Color FV. In the same way as UEC-FOOD100, DCNN-FOOD outperformed DCNN greatly, and fine-tuned DCNNs outperformed others. By adding 1000 food-related categories for pre-training of DCNN, the top-1 classification rate for FOOD256 was improved by 14.83 points, while the difference between DCNN(ft) and DCNN-FOOD(ft) was only 3.93 points. Finally we achieved 67.57% top-1 accuracy for UEC-FOOD256 by DCNN-FOOD(ft).

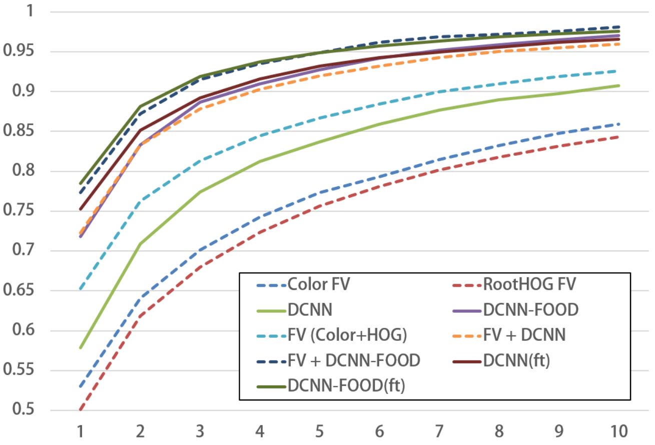

Fig. 3. Classification accuracy within the top N candidate on UEC-FOOD100 with DCNN, RootHoG-FV, Color-FV, their combinations and fine-tuned DCNN.

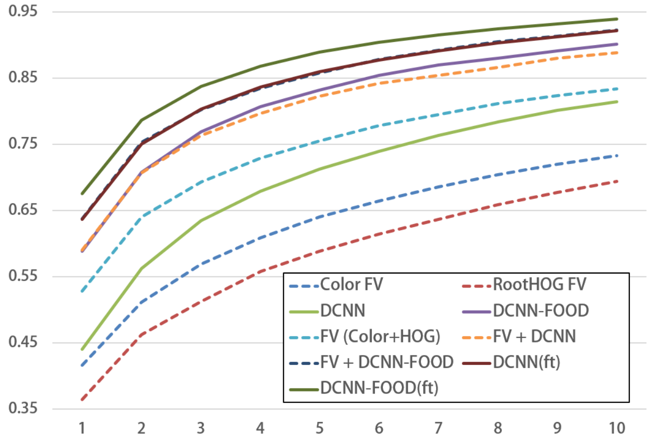

Fig. 4. Classification accuracy within the top N candidate on UEC-FOOD256 with DCNN, RootHoG-FV, Color-FV, their combinations and fine-tuned DCNN.

In addition, we fine-tuned DCNN and DCNN-FOOD with ETH Food-101 food image dataset[5] consisting of 101 kinds of food images where each of 101 categories has 1000 images, and examined the food recognition performance on ETH Food-101. We followed the official train/test splits for the evaluation of 101 food category classification. Table 3 shows the results obtained by random forest based discriminative component mining [1], DCNN trained from scratch [1], fine-tuned DCNN pre-trained with ImageNet1000 (DCNN(ft)), and fine-tuned DCNN pre-trained with 2000 ImageNet categories including food-related categories (FOOD-DCNN(ft)). Compared between DCNN trained from scratch and the fine-tuned ImageNet-1000 DCNN(ft), DCNN(ft) was much improved by 12.04 points. Regarding DCNN(ft) and FOOD-DCNN(ft), FOOD-DCNN(ft) outperformed DCNN(ft) by 1.97 points.

From these results, it has been strongly proved that it was effective and essential to use 1000 food-related categories for pre-training of DCNN for better performance on food classification.

[5] https //www.vision.ee.ethz.ch/datasets_extra/food-101/

Table 3. The number of selected photos and their precision(%) with four different combinations.

food category	raw	FC	FC+100	DCNN	DCNN (May 2011-March 2015)
ramen noodle	275652 (72.0%)	200173 (92.7%)	80021 (99.7%)	132091 (99.5%)	272375
beef ramen noodle	861 (94.3%)	811 (99.0%)	555 (99.7%)	590 (100%)	1876
curry	224685 (75.0%)	163047 (95.0%)	59264(99.3%)	68091 (100%)	156397
cutlet curry	10443 (92.7%)	9073 (98.0%)	6339 (99.3%)	7024 (99.9%)	18196
sushi	86509 (69.0%)	43536 (86.0%)	25898 (92.7%)	22490 (99.8%)	83289
dipping noodle	33165 (88.7%)	24896 (96.3%)	22158 (99.0%)	22004 (100%)	69632
omelet with fried rice	34125 (90.0%)	28887 (96.3%)	17520 (99.0%)	20039 (99.9%)	78378

4. APPLYING THE BEST DCNN FOR TWITTER FOOD MINING

In this section, we describe an example application of the best DCNN for UEC-FOOD100, DCNN-FOOD(ft2). Note that in the previous section, we used part of the dataset as training data for fine-tuning because of five-fold cross validation, while in this section we used whole the augmented UEC-FOOD100 data for fine-tuning the DCNN pre-trained with 2000 ImageNet categories including 1000 food-related categories. For Twitter food mining, it is required to exclude non-food photos. To do that, we added a non-food category to 100 categories of UEC-FOOD100. We used 10000 non-food photos collected from Twitter as training data for a non-food category. We fine-tuned the pre-trained DCNN-FOOD as a 101-class classifier which can recognize non-food photos as well as 100-class food photos. Before applying the fine-tuned DCNN for Twitter data, we evaluated food-nonfood classification performance by five-fold cross validation. As a results, it achieved 98.96%.

Following [20], we used 122,328,337 photo tweets with Japanese messages out of 988,884,946 photo tweets over all the world collected from May 2011 to August 2013 for two years and four months from the Twitter Stream. From these photo tweets, we selected 1,730,441 photo tweets the messages of which include any of the name words of the 100 target foods as the first step.

In [20], in the second step, they applied a "foodness" classifier (FC) to all the selected images. After applying FC, they applied 100-class one-vs-rest individual food classifiers. As a result, they obtained 470,335 photos which are judged as food photos corresponding to any of the 100 target food categories by the processing pipeline proposed in [20]. They adopted Fisher Vector and linear classifiers for FC and 100-class classifiers.

Instead of FC and FV-based 100-class food classifiers, we applied the 101-class DCNN classifier, which can achieve non-food photo detection and food photo classification simultaneously, to 1,730,441 Twitter photos selected by keyword search of the food names. In this large-scale food classification experiment, we found that DCNN was very suitable for large-scale image data, since it takes only 0.03 seconds to classify one food photo with GPU and totally it needed about four hours to classify 1,730,441 photos by four GPU machines. Finally, we obtained 581,271 food photos, which was 1.24 times as many as the result in [20].

Due to the page limitation, we show only seven results of the top five categories and two additional categories out of 100 food categories on Table 3, and show 40 automatically detected photos of each of "ramen noodle", "dipping noodle (tsukemen)“, "sushi" and "omelet" in Figure 5. Note that the precision rates shown in the table were estimated by subjective evaluation of random sampled 1000 photos for each categories, and the rightmost column of Table 3 shows the number of the food photos detected by DCNN from the Twitter stream from May 2011 till March 2015 for about four years.

Compared DCNN with FC+100 which corresponds to the final results of [20], the number of obtained food photos and precision are improved. Especially the number of ramen photos were increased greatly, while the number of sushi photos were decreased. Although the precision of sushi in [20] was low, it was improved much and became almost perfect. This is because non-food photos representing inside sushi restaurants and people face photos were completely excluded by food-nonfood classification of the DCNN. Regarding other foods than sushi, the precision rates were almost perfect. Only several photos are found in the 1000 random sampled photos in the evaluation time. We show some irrelevantly recognized photos in Figure 6.

5. CONCLUSIONS

In this paper, we examined the effectiveness of pre-training and fine-tuning of deep convolutional neural network (DCNN) with a small-scale food dataset which has around 100 training images for each of the food categories. In the experiments, we have achieved the best classification accuracy, 78.77% and 67.57%, for the UEC-FOOD100/256 dataset, which proved that that fine-tuning of the DCNN pre-trained with a large number of food-related categories (DCNN-FOOD) can boosted the classification performance the most greatly.

In addition, we applied the food classifier employing the best combination of the DCNN techniques to Twitter photo data. We have achieved the great improvements on food photo mining in terms of both the number of food photos and accuracy. In addition to its high classification accuracy, we found that DCNN was very suitable for large-scale image data, since it takes only 0.03 seconds to classify one food photo with GPU.

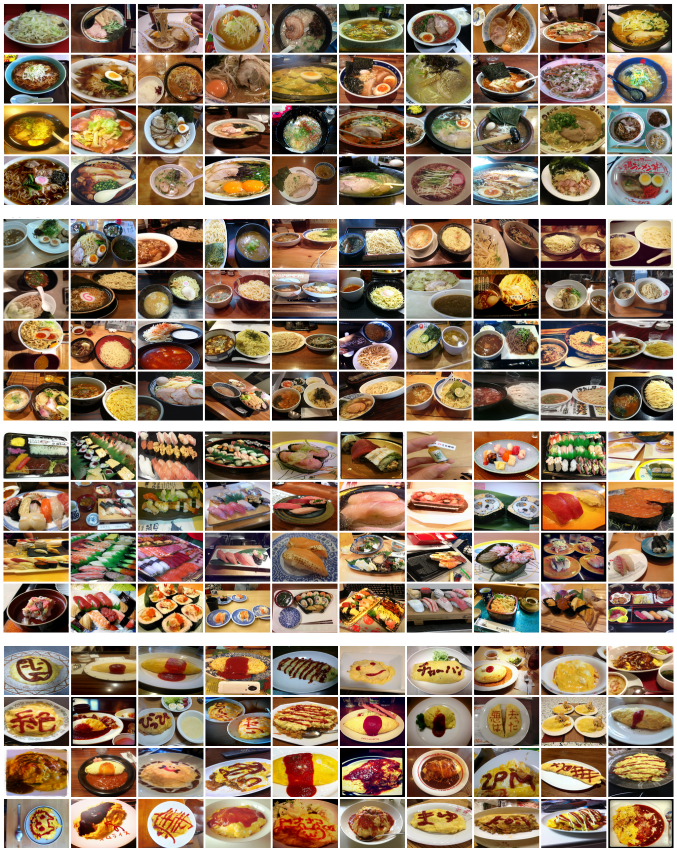

Fig. 5. Examples of automatically detected food photos with the proposed DCNN from the Twitter stream. (From the top) ramen noodles, dipping noodles (tsukemen), sushi and omelet.

For future work, we will implement the proposed framework on mobile devices. To do that, it is needed to reduce the amount of the pre-trained DCNN parameters which consist of about 60 million floating values. Regarding Twitter food photo mining, we plan to extend the framework to analysis food distribution and preference from the geo-spatial and temporal aspects.

6. REFERENCES

[1] L. Bossard, M. Guillaumin, and L. V. Gool, "Food-101 - mining discriminative components with random forests," in *Proc. of European Conference on Computer Vision*, 2014.

[2] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *Proc. of ACM International Conference Multimedia*, pp. 1085–1088, 2014.

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

[4] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Proc. of IEEE International Conference on Multimedia and Expo*, pp. 1554–1564, 2012.

[5] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung, "Automatic chinese food identification and quantity estimation," in *SIGGRAPH Asia*, 2012.

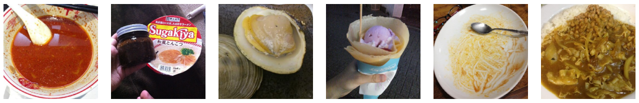

Fig. 6. Examples of misclassified Twitter food photos. Eaten ramen bowl (recognized as "ramen"), unopened instant ramen (ramen), a clam (sushi), ice cream (sushi), an eaten plate (omelet) and curry without cutlet (cutlet curry).

[6] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Combining global and local features for food identification in dietary assessment," in *Proc. of IEEE International Conference on Image Processing*, 2011.

[7] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Proc. of IEEE Computer Vision and Pattern Recognition*, 2010.

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

[9] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2014.

[10] A Krizhevsky, I Sutskever, and G E Hinton, "Imagenet classification with deep convolutional neural networks.," in *Advances in Neural Information Processing Systems*, 2012.

[11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. of International Conference on Machine Learning*, 2014.

[12] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, 2007.

[13] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.

[14] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. of British Machine Vision Conference*, 2014.

[15] J. Wan, D. Wang, S. Hoi, C. Hong, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. of ACM International Conference Multimedia*, pp. 157–166, 2014.

[16] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. of European Conference on Computer Vision*, 2014.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. of IEEE Computer Vision and Pattern Recognition*, 2014.

[18] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. of European Conference on Computer Vision*, 2010.

[19] Y. Jia, "Caffe: An open source convolutional architecture for fast feature embedding," `http://caffe.berkeleyvision.org/`, 2013.

[20] K. Yanai and Y. Kawano, "Twitter food image mining and analysis for one hundred kinds of foods," in *Proc. of Pacifit-Rim Conference on Multimedia (PCM)*, 2014.

[21] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 2911–2918, 2012.

[22] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *Advances in Neural Information Processing Systems*, pp. 414–422, 2009.