Zhang XJ, Lu YF, Zhang SH. Multi-task learning for food identification and analysis with deep convolutional neural networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(3): 489–500 May 2016. DOI 10.1007/s11390-016-1642-6

Multi-Task Learning for Food Identification and Analysis with Deep Convolutional Neural Networks

Xi-Jin Zhang, *Student Member, CCF*, Yi-Fan Lu, *Student Member, CCF*, and
Song-Hai Zhang*, *Member, CCF, ACM*

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

E-mail: zhangxijin91@gmail.com; luyifanfrank@foxmail.com; shz@tsinghua.edu.cn

Received December 1, 2015; revised March 9, 2016.

Abstract In this paper, we proposed a multi-task system that can identify dish types, food ingredients, and cooking methods from food images with deep convolutional neural networks. We built up a dataset of 360 classes of different foods with at least 500 images for each class. To reduce the noises of the data, which was collected from the Internet, outlier images were detected and eliminated through a one-class SVM trained with deep convolutional features. We simultaneously trained a dish identifier, a cooking method recognizer, and a multi-label ingredient detector. They share a few low-level layers in the deep network architecture. The proposed framework shows higher accuracy than traditional method with handcrafted features, and the cooking method recognizer and ingredient detector can be applied to dishes which are not included in the training dataset to provide reference information for users.

Keywords multi-task learning, convolutional neural network, food recognition, machine learning

1 Introduction

There is an old Chinese saying as "Food is the first necessity of the people". Diet plays an important role in people's daily life with its strong correlation to health and chronic diseases, such as obesity, diabetes, heart disease, and cancer. Nowadays, lots of people use portable devices such as smartphones to take photos of what they eat every day. These photos, as a kind of visual media, provide a way of diet logging and contain valuable information that can help people in improving their health. However, most people just take the food photos and leave them behind, and manually mining information from them is tedious and time-consuming. To overcome this difficulty, lots of attention has been paid to automatic food recognition. A practicable automatic food recognition system can help people better understand the nutritional content of their diet, and provide medical advice. Meanwhile, different dishes are composed of different combinations of ingredients and can be cooked in different ways. Ingredients and cooking methods are closely related to food's nutrition and health effects. One can learn the ingredients and cooking method of a dish through the recipe, given a known dish. But Asian foods (particularly Chinese foods) usually have many more ingredients and cooking methods than western foods. Different combinations of ingredients and cooking methods will result in a myriad of dishes, which cannot be exhaustively included in a dataset. Therefore it is important to build a system which can not only identify a dish but also recognize its cooking method and ingredients. Yang *et al.*[1] used food ingredient pixel labels to form a feature for food recognition. Nine food ingredient categories can be recognized and pixel-wise labels are needed in the training progress. To the best of our knowledge, there is

Regular Paper

Special Section of CVM 2016

This work was supported by the National High Technology Research and Development 863 Program of China under Grant No. 2013AA013903, the National Natural Science Foundation of China under Grant No. 61373069, the Research Grant of Beijing Higher Institution Engineering Research Center, and the Tsinghua University Initiative Scientific Research Program.

*Corresponding Author

no existing work that provides a dataset with cooking method labels or trains a cooking method recognizer.

Automatic food identification and analysis is a challenging task because dishes are highly variable in appearance, and the spatial relationships among ingredients are complex. Some ingredients may be occluded or placed at a totally different location in a given food image due to its cooking and assembly method, especially in Chinese foods, which severely decreases the reliability of the descriptive local features of ingredients and the performance of the conventional local-feature based methods. The complex spatial relationships also decrease the reliability of optimal feature extraction methods with image components like in [2], which are often used in face recognition. In [3], it is found that many general objects (like giraffes or boats) can be cross-depiction classified robustly with models that explicitly encode spatial relations between parts. Unlike in these general objects, the spatial structure features in foods are weak. This indicates that food identification is more challenging than common object identification.

Deep convolutional neural network (DCNN) is now a state-of-the-art technique for image recognition challenges such as the large scale visual recognition challenge[4]. DCNN is also successfully applied to multi-label cross-domain detection problem[5] and multi-task learning[6]. In general, the advantage of DCNN over conventional hand-crafted feature approaches is that it can estimate optimal feature representations for datasets adaptively. This feature allows DCNN to take advantage of large amounts of data to obtain better results. However, the previous studies are mainly concentrated on obtaining better results on small datasets[7-9].

To exploit the potential of DCNN on large datasets, we built up a dataset of 360 classes of different foods including Chinese foods, Japanese foods, Korean foods, desserts and a few kinds of western foods. There are at least 500 images for each class, which were collected from the Internet along with their class labels. Outlier images were detected and eliminated through a one-class SVM trained with deep convolutional features. We labeled each class with its ingredients and cooking method. A multi-task deep network was proposed and the three tasks were trained simultaneously on our dataset. These tasks share the convolutional layers and the first full-connected layer. We evaluated the performance of our system and analyzed the effect of multi-task training. Our method outperforms the traditional local-feature-based methods by more than 30% in accuracy on our dataset. The experimental results and comparisons are described in detail in this paper.

The contributions of our work include:

1) A larger and more complex dataset close to practical situations. We labeled the food categories with their ingredients and cooking methods. The dataset will be open to researchers interested in this topic and for a fair comparison. We also showed that the deep convolutional features are useful for eliminating outliers in food images.

2) A multi-task food identification and analysis system. For the food identification task, the overall accuracy for 360 categories of foods achieves 57.25%. The success rate at top-3 and top-5 candidates can reach 76.00% and 82.29%. The performance is significantly better compared with the baseline method. Our experiments on a large and complex dataset show that DCNN offers several advantages over the state of the art.

3) A food ingredient detector and a cooking method recognizer. The recall for 93 categories of food ingredients achieves 69.41% and the precision is 60.74% with our system. The accuracy of 11 cooking methods achieves 69.50%. To our knowledge, this is the first work on food ingredient detection and cooking method recognition. The experiment shows that the cooking method recognizer and the ingredient detector can also be applied to dishes which are not included in the training dataset to provide reference information for users.

2 Related Work

Several studies have been proposed for food recognition and identification. The Pittsburgh Fast-Food Image Dataset[10] is a dataset of American fast-food images, which was used to evaluate food recognition method in [1]. This dataset has 61 categories of western fast food, which may not have enough food diversity. [11-12] use color, texture, gradient, and SIFT features to train separate classifiers, and the weighted combination of them with the multiple-kernel learning method achieves 61.3% and 62.5% in accuracy on 50 and 85 categories of Japanese foods respectively. [7] also trains a separate classifier for each feature and uses multi-class AdaBoost algorithm to fuse these classifiers. Chen *et al.*[7] built a dataset with 50 categories of major Chinese foods and achieved 68.3% in average accuracy. In [13], the proposed method achieves 55.8% and 68.9% in accuracy on multiple food-item images and single food-item images with the help of candidate region detecting respectively. Their dataset has 100 food categories and

9 060 images in total. A real-time mobile food recognition system was developed in [14] which uses a linear SVM with bag-of-SURF and color histogram features. The system achieves 53.5% accuracy on a 50-category food dataset. The local and global image features were tested in [15]. The authors of [15] reported that the color feature worked best for food recognition.

For the food ingredient recognition problem, a real-time recognition system was proposed in [16]. Bag-of-features and RBF kernel SVM were utilized to recognize 30 kinds of food ingredients. In [1], each image pixel was soft-labeled into food ingredient categories employing the semantic texton forest (STF). Pairwise features were then extracted using these labels, dish category was classified, and nine categories of ingredients were used in Yang *et al.*'s work[1].

DCNN has been applied to food recognition in recent years. In [8], multiple network structures were evaluated, and a two-layer network was reported to achieve the best accuracy, 73.70%. But less than 30 000 images with 10 categories of foods were contained. DCNN on food detection task was tested. Kawano and Yanai[9] reported that they failed to confirm that the DCNN-based method outperformed the conventional method because the size of the UEC-FOOD100 dataset was not large enough.

Multi-task learning with DCNN is widely employed in the human part detection and pose regression[6]. In [17], the visual feedback model was designed with feedback between "what" and "where" tasks, improving adaptiveness and closeness for object recognition simultaneously. In [18], the authors found that joint-training tends to find the most useful features in the input for both tasks. In [19], a more robust network was proposed by integrating extra tasks into surface normal estimation task. These studies inspire us to add extra tasks to the dish identification task.

3 Dataset Build-Up

We built a dataset that is larger and more complex than the existing ones. The images and food category labels were collected from the Internet. We cleaned the data and labeled the ingredients and cooking method of each category.

3.1 Data Collecting and Merging

All food images and food category labels are collected from a Chinese cooking website called "Xiachufang"[①]. The food images are uploaded by the website users and most foods in the images are cooked by themselves. The original image size is 280x280 and the label names are in Chinese. We wrote a web crawler in Python to download images automatically, and only those food categories that have more than 500 images were downloaded. We collected about 270 000 images of 556 kinds of foods in the beginning, but we found that some foods were labeled with more than one name. These differently named foods should be merged before being used in the dataset.

Our merging strategy is as follows. 1) Synonym replacement for the food names was conducted with a manually defined dictionary which contains 20 pairs of words. 2) Chinese word segmentation for the food names was done based on the Viterbi algorithm, using Jieba, a word segmentation library in Python. 3) The Levenshtein distance between food names was calculated, and Chinese words were used as the basic units. 4) The food categories whose names' Levenshtein distance is less than 4 were merged. 5) A few food categories which had not been successfully merged were merged manually.

After the merging, there are 360 food categories left. Some examples of images in the dataset are shown in Fig.1. Most of the food categories are Chinese foods or desserts (the first and the second rows in the figure), the others are Korean foods (e.g., Bibimbap), Japanese foods (e.g., Sushi) (the 3rd row in the figure), and western foods (e.g., Sandwich, Spaghetti,

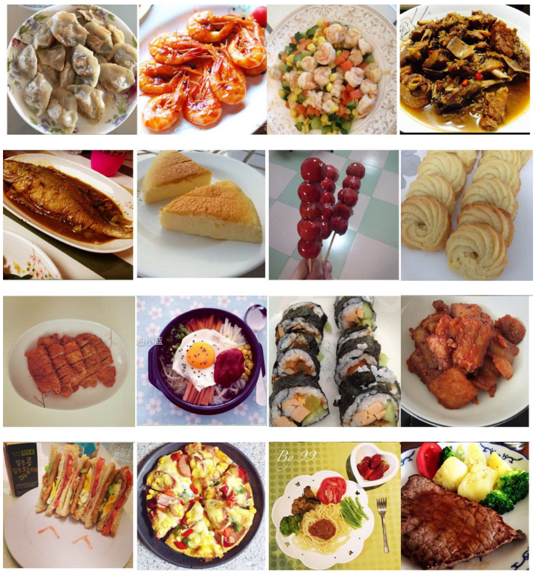


Fig.1. Examples of images in the dataset.

[①] http://www.xiachufang.com/, Apr. 2016.

and Pizza) (the last row in the figure). As the data is collected from a Chinese website, the numbers of dishes are not balanced between oriental and western cuisines. We believe that the system that can handle Chinese foods well should be easily transformed to deal with many other kinds of foods because of the great diversity of Chinese foods. Our dataset and the proposed system can be extended to foods from all over the world in future.

3.2 Data Cleaning

Since the images are user-generated, among all the images in one class, there are always some outliers that look unreasonably different from other images in the same class. The outliers will affect the model learning and thus should be eliminated. A one-class SVM[20] was trained with deep convolutional features to do the elimination. We took a pre-trained DCNN model on the ILSVRC 1 000-class dataset[4], changing only the output number of the last fully-connected layer to 360. The pre-trained model was used for efficiency. The model was fine-tuned with base learning rate 0.000 1, which was decreased by half every 20 000 iterations. The fine-tuning used all the collected images including all the outliers. After 100 000 iterations (about 25 epochs), the accuracy reached 44.99%. We extracted the network signals just before the last layer as a feature vector of 4 096 dimensions. A one-class SVM for each class was trained with the extracted feature using RBF kernel and setting $\mu = 0.5 + 0.95 \times t, \gamma = 1.0/4\,096$. Images whose SVM decision function values were in the smallest $t\%$ of all the values in their categories were regarded as outliers and eliminated. Through the observation of our collected images, we set $t = 6$. Most of the outliers were eliminated and few false outliers were also eliminated. Some examples of typical outliers detected by our data cleaning method are shown in Fig.2.

3.3 Labeling

The food category labels were collected alone with the images and merged, while the food ingredient and the cooking method need to be manually labeled. Instead of labeling each image with all its ingredients, we just labeled each category with its typical ingredients. Some of the ingredients are certainly in or not in a certain dish, but for some others it is hard to decide. We treated the task as a multi-label detection problem. We labeled each ingredient as $\{-1, 0, 1\}$ with 0 meaning ignored, -1 negative and 1 positive. There are 93 kinds of main food ingredients in our dataset. Any ingredient was present in at least one food category and at most 79 categories. Table 1 shows the 10 most-frequent food ingredients in our dataset.

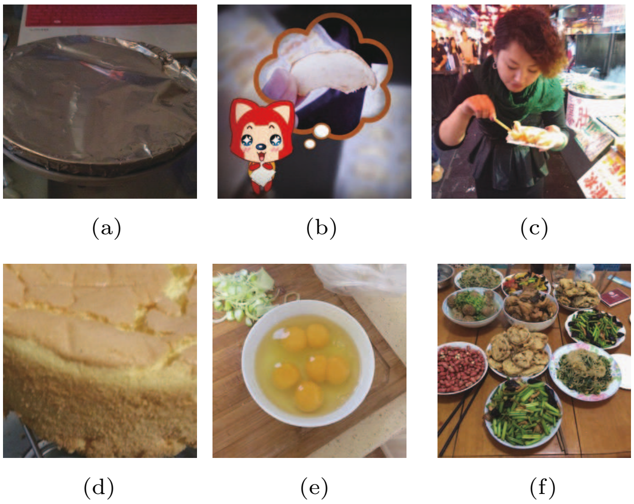


Fig.2. Examples of typical outliers detected. Food names and the reasons to be outliers are listed. (a) Lobster sauce ribs: covered. (b) Apple chips: too much editing or decoration. (c) Spicy boiled fish: non-food object in image. (d) Braised hairtail: wrong food category uploaded by user. (e) Leeks scrambled eggs: raw ingredients instead of dishes. (f) Fried meatballs: more than one dish in the image, not focused.

Table 1. 10 Most-Frequent Food Ingredients in Our Dataset

Ingredient	Quantity
Wheat flour	79
Chili or green pepper	48
Meat	32
Egg	29
Potato	28
Milk	22
Chicken	20
Glutinous rice	17
Garlic	13
Chinese cabbage	12
Fish	12
Sugar	12
Beef	11
Green vegetables	11
Tomato	11

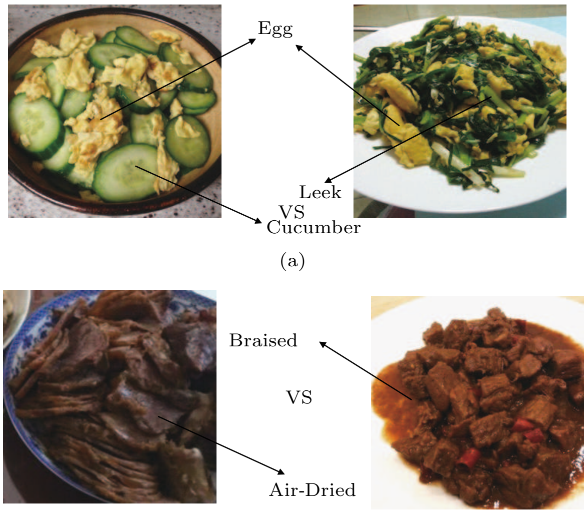
Compared with western foods, more subdivided cooking methods are used in the production of Chinese foods. There are 11 cooking methods in our dataset which are shown in Chinese Pinyin and explained in English in Table 2.

Table 2. Cooking Methods in Our Dataset

Chinese Name	English Explaination
Kao	Grilled or roasted or baked
Jian	Pan-fried
Chao	Multiple ingredients fried together, with little oil
Liang-ban	Salad with multiple seasonings
Zha	Deep-fried
Zheng	Steamed
Tang-cu	Sweet and sour
Shao	Braised or teriyaki
Tianpin	Dessert
Lu	Stewed with soy sauce and spices, then let cool or air-dried
Zhu	Stewed or boiled

4 Method

Our multi-task framework consists of three tasks: 1) the dish identification task, whose aim is to identify which food category the dish belongs to; 2) the cooking method recognition task, whose goal is to predict how the food is cooked; 3) the food ingredient detection task, whose target is to determine whether each food ingredient is in the dish. These tasks have a natural correlation. As shown in Fig.3, most dishes can be identified through two questions: what the ingredients are and which cooking method is used. The features learned to distinguish the ingredients and the cooking method are useful for the dish identification task. Therefore, in our proposed system, these three tasks share the lower layers of a DCNN structure while having their own fully-connected layers and loss functions. The loss functions are linearly combined to form the global loss function.



(a)

(b)

Fig.3. Dishes can be identified by the ingredients and the cooking method. (a) Different ingredients. (b) Different cooking methods.

4.1 Network Structure

DCNN is a multilayer neural network, whose neurons take small patches of the previous layer as input. A CNN system consists of convolution layers and pooling (or subsampling) layers. In the convolution layer, weights can be considered as $n \times n$ filters. Each filter convolves the input to output a kind of feature. This configuration simulates the local perception characteristics of the human visual system. Compared with general object recognition, food recognition relies more on local color and texture features than on edges and contours, thereby we use smaller filter size and stride in our network. Each layer has many filters that generate different outputs. According to Kagaya *et al.*'s report[8], there are more edge-specific kernels in the kernels for general images, but almost all the kernels are color-specific in their trained network for food images. In order to distinguish subtle color differences in our complex dataset, the number of filters in the first convolution layer in our network was set to 128, which is a little more than the number in the AlexNet network[4].

The pooling layer produces the outputs by calculating the maximum or average value over rectangular regions. This makes the convolution layers' outputs more position-invariant. The activation method of all the pooling layers in our network is max-pooling, and the region size is 3×3 with stride length 2.

A typical DCNN contains one or several fully-connected layers after multiple convolution layers and pooling layers. In some studies like [9], the fully-connected layer's outputs of the last but one layer are extracted as a feature vector. The feature vector is used to train an SVM classifier, which achieves good classification results. The three tasks in our framework share the first fully-connected layer with 4 096 outputs. Because most of the Chinese foods cannot be distinguished with the spatial distribution of visual patterns, we trained a common fully-connected layer to acquire high-level features for the three tasks instead of convolution features. The quantity and the output number of fully-connected layers for the three tasks are chosen according to their complexities.

The output values of neurons in convolutional layers or fully-connected layers are calculated by an activation function f_{act}. Most of the activation functions in our network are Rectified Linear Units (ReLu)[21], i.e., $f_{\text{act}}(x) = \max(0, x)$. The computation cost for this function is low[21]. We used the softmax function in the last layer of the dish identification task and the cooking method recognition task, and the sigmoid function in the last layer of food ingredient detection task.

Our network structure is shown in Fig.4.

4.2 Loss Function of Dish Identification Task and Cooking Method Recognition Task

The dish identification and the cooking method recognition are one-out-of-many image classification problems. The outputs of each task's last fully-connected layer are mapped to each task's probability distribution over classes using the softmax function. We used the multinomial logistic loss as the loss functions of these two tasks,

$$\begin{aligned}
\hat{p}_{dnk} &= \exp(x_{nk}) / \left(\sum_{k'}^{K} \exp(x_{nk'}) \right), \\
E_d &= \frac{-1}{N} \sum_{n=1}^{N} \log(\hat{p}_{dnl_{dn}}), \\
\hat{p}_{cnj} &= \exp(x_{nj}) / \left(\sum_{j'}^{J} \exp(x_{nj'}) \right), \\
E_c &= \frac{-1}{N} \sum_{n=1}^{N} \log(\hat{p}_{cnl_{cn}}).
\end{aligned} \tag{1}$$

N is the number of training samples, K is the number of dish classes and J is the number of cooking method classes; x_{nk} is the k-th dish identification layer (i.e., last fully-connected layer of the dish identification task)

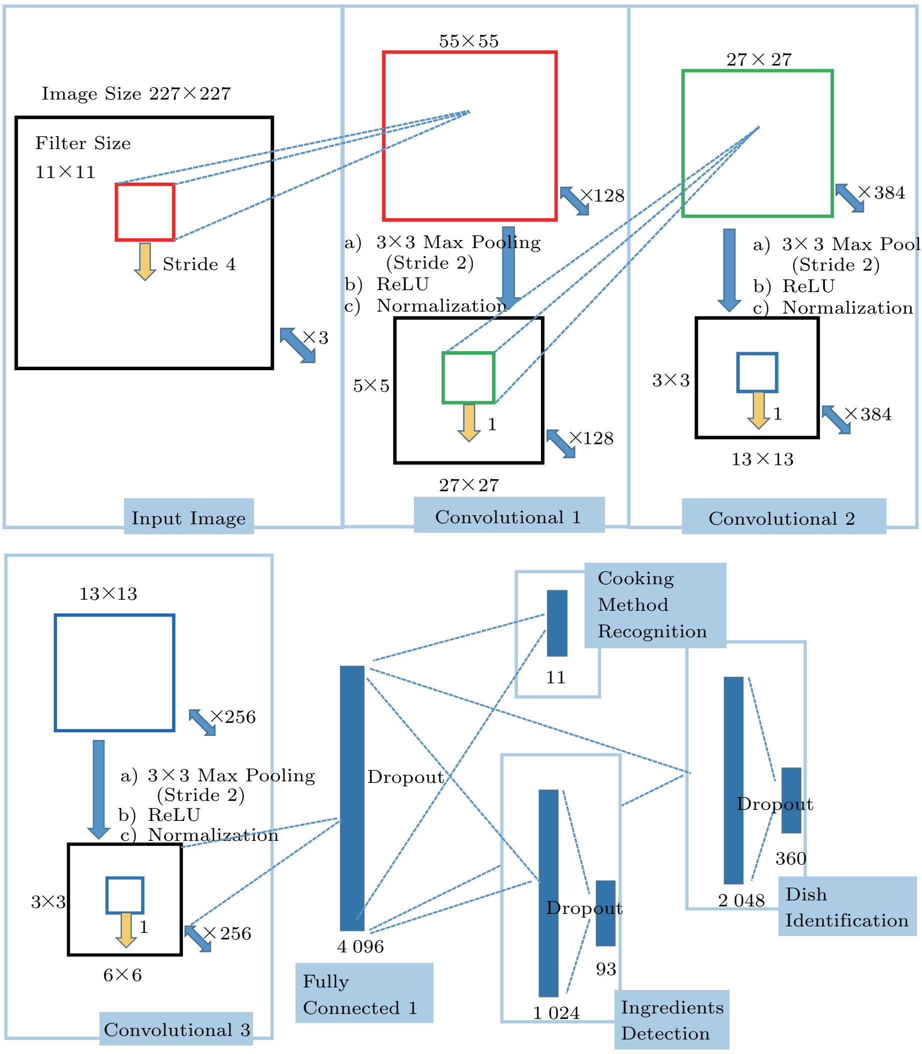


Fig.4. Our network architecture.

output of the n-th sample and x_{nj} is the j-th cooking method recognition layer output of the n-th sample; $\hat{p}_{dnk}$ is the softmax probability of the n-th sample and the k-th dish class, and $\hat{p}_{cnj}$ is the softmax probability of the n-th sample and the j-th cooking method; l_{dn} is the ground truth dish class label of the n-th sample, and l_{cn} is the ground truth cooking method label of the n-th sample; E_d is the loss function of the dish identification task and E_c is the loss function of the cooking method recognition task. E_d and E_c are linearly combined to form the global loss function as detailed in Subsection 4.4.

4.3 Loss Function of Multi-Label Food Ingredient Detection Task

The food ingredient detection is a multi-label classification task. We only cared about whether a certain food ingredient exists in a food image but did not care about the location. The outputs from the last fully-connected layer of this task were mapped to probability predictions $\hat{p}_{nk} = \sigma(x_{nk}) \in [0, 1]$ using the sigmoid function. n and k are indicators for the training samples and ingredients respectively. The label of the n-th sample for the k-th ingredient is $l_{nk} \in \{-1, 0, 1\}$, with 0 meaning ignored, -1 negative and 1 positive. A modified cross-entropy loss is used in this task.

$$\begin{aligned} E_i &= \frac{-1}{N} \sum_{n=1}^{N} \frac{1}{K} \sum_{k=1}^{K} f(L_{nk}, l_{nk}), \\ L_{nk} &= \hat{p}_{nk}(h(l_{nk}) - h(\hat{p}_{nk})) - \\ &\quad \log(1 + e^{\hat{p}_{nk} - 2\hat{p}_{nk} h(\hat{p}_{nk})}), \\ h(x) &= \begin{cases} 1, \text{ if } x > 0, \\ 0, \text{ otherwise}, \end{cases} \\ f(x, l) &= \begin{cases} 0, \text{ if } l = 0, \\ x, \text{ if } l \neq 0 \text{ and } x \geqslant 0, \\ \lambda_n x, \text{ otherwise}, \end{cases} \end{aligned} \tag{2}$$

where N is the number of training samples, $\hat{p}_{nk}$ is sigmoid output probability of the k-th ingredient in the n-th sample, and l_{nk} is the ground truth label value for each ingredient in each training sample. L_{nk} is the unbiased loss value of the k-th ingredient in the n-th sample; $h(x)$ and $f(x, l)$ are auxiliary functions used to simplify the representation of the loss function. This loss function can handle the negative or ignored labels. λ_n is a bias weight, which was used to balance the recall and the precision in the detector training (see Section 5 for details). The loss function E_i is linearly combined into the global loss function as detailed in Subsection 4.4.

4.4 Global Loss Function

Our global loss function is the linear combination of the dish identification loss, the cooking method classification loss and the multi-label ingredient detection loss, averaging over all N training images,

$$\Phi = \lambda_d E_d + \lambda_c E_c + \lambda_i E_i. \tag{3}$$

λ_d, λ_c and λ_i are the weights for dish identification, cooking method recognition and food ingredients detection respectively; E_d, E_c and E_i are the loss functions for the three individual tasks (see (1) and (2)). In our experiments, we found that small changes of these weights did not influence the final result much while values that were too large or too small led to numerical instability during the training process or significant under-fitting in some task. Therefore we set these values around 1.0. Setting λ_i smaller than the other two weights led to a slightly better result because E_i was generally larger than the loss of the other two tasks. In our final system, λ_d is 1.0, λ_c is 1.0 and λ_i is 0.5.

4.5 Training

We resized the images for training to 256 × 256, whose original size is 280 × 280. Then, the mean values of three channels were subtracted from training images. We also augmented the training data by randomly cropping the images to 227 × 227 and randomly mirror flipping the images.

We jointly trained the three networks with global loss function in (3). Mini-batched SGD[22] is used to do back-propagation weight updating. For the layers with more than one subsequent layer (e.g., fully-connected layer 1 in Fig.4), the gradients from its subsequent layers are summed together to update the weights. The mini-batch size for SGD was 50, the momentum was 0.9, and the weight decay was 0.0005. The starting learning rate for SGD was 0.001 and decreased by half every 40000 iterations for the first 200000 iterations. Then the learning rate was changed to 0.0001 from 0.0000625 and decreased to one tenth of it every 80000 iterations. We trained the network for 400000 iterations in total, i.e., about 100 epochs.

"Dropout"[23] is used between every pair of fully-connected layers to prevent over-fitting. The dropout probability was 0.5 in our experiments. In each iteration, the neurons before a dropout layer are randomly selected with probability 0.5 to do forward and back-propagation. In the testing stage, all the neurons are used for prediction with their output values multiplied by 0.5. No significant over-fitting was observed during the training with the help of the "Dropout".

5 Experiments and Results

We present experiments using our method on our dataset. We also use DCNN only for dish identification task and a multi-class SVM with traditional image features as a comparison.

5.1 Experimental Setup

Before all the experiments, the dataset was randomly split: 80% for training, 15% for testing, and 5% for validation. We trained the network with Caffe, a C++ framework for deep learning. A NVIDIA GTX970 GPU was used to do the training. It took about half an hour to complete 10 thousand iterations. We also trained a network only for dish identification by deleting the cooking method related and the ingredients related fully-connected layers and their losses from our proposed multi-task network.

From previous work, we observe that the SIFT and color features are the best for food recognition and the sparse-coded features are better than naive bag-of-SIFT or color histograms. Thus we compared our method with a multi-class SVM trained with Fisher Vector (FV) coded SIFT and color features. Specifically, we extracted SIFT features (at 2 scales, cell width = 4, and 8) and CIE color features, with a Fisher-codebook. The size of the codebook was 100, and there were two levels in the spatial pyramid. We extracted features and trained the SVM with Steve's Object Detection Toolbox[24].

5.2 Evaluation on Dataset

We tested the proposed network, the dish identification network without additional tasks and the SVM method on our dataset.

For the food identification task, the classification accuracy within the top N candidates of our method is shown in Fig.5.

Our network achieves 57.25% in the top-1 accuracy, 76.00% in the top-3 accuracy, and 82.29% in the top-5 accuracy. The comparison among different methods is shown in Table 3.

This indicates that traditional SVM method with hand-crafted features cannot handle complex data with a large number of categories well. Those features cannot distinguish between a variety of Asia dishes to such a fine degree. Our method outperforms the traditional method by more than 30% in accuracy. In the previous literatures, the outperformance is less than 20%. It can be inferred that the DCNN method shows more advantages over the traditional method on a large and complex dataset and the DCNN approach has the promise to become practical. The comparison also shows that the food identification task can slightly benefit from the other two tasks.

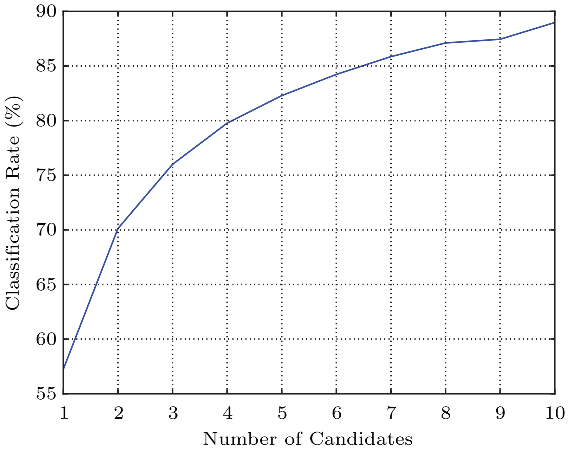


Fig.5. Classification accuracy within the top N candidates.

Table 3. Top-1 Dish Identification Accuracy of Different Methods

Method	Top-1 Accuracy (%)
Ours	**57.25**
Our single task	54.95
Color FV + SVM	8.93
SIFT FV + SVM	20.23
Color + SIFT FV + SVM	26.39

For the cooking method recognition task, our network achieves 69.50% in the top-1 accuracy and 92.13% in the top-3 accuracy. Table 4 shows the confusion matrix of cooking method recognition. "Zha" and "Jian" are mainly confused with "Kao", because they both use much oil making the food look very greasy. "Zhu" and "Shao" are easily confused. The reason may be that the only difference between them is how much water is used and whether to use soy sauce. Foods cooked in these two ways may look very similar. In fact, identifying the cooking method from food images is not an easy task for human. Table 5 shows the recognition rates of cooking methods by human for comparison.

Table 4. Confusion Matrix of Cooking Method Recognition

	Kao (%)	Jian (%)	Chao (%)	Liang-Ban (%)	Zha (%)	Zheng (%)	Tang-Cu (%)	Shao (%)	Tianpin (%)	Lu (%)	Zhu (%)
Kao	**67.2**	7.1	1.5	1.1	7.0	3.4	0.6	2.9	7.4	0.8	1.0
Jian	**12.9**	**60.4**	2.4	3.0	9.6	4.4	0.0	3.0	2.7	0.5	1.0
Chao	2.4	1.7	**75.9**	3.0	2.2	2.8	1.2	6.3	0.4	0.1	4.1
Liang-ban	3.2	2.8	4.8	**69.6**	3.4	5.4	1.8	1.8	3.8	0.0	3.2
Zha	**10.1**	8.1	2.9	0.9	**66.6**	3.7	0.4	3.0	3.3	0.1	1.0
Zheng	4.4	2.7	3.7	2.7	2.0	**74.1**	0.6	2.9	3.5	0.2	3.3
Tang-cu	3.6	0.0	2.9	2.6	2.9	0.3	**71.4**	**14.3**	0.3	0.3	1.3
Shao	6.4	1.8	8.2	1.6	2.5	2.5	7.4	**61.9**	1.1	1.2	5.6
Tianpin	7.2	2.8	1.2	2.4	2.3	4.6	0.2	0.5	**76.1**	0.7	1.9
Lu	3.3	0.0	0.0	1.1	1.1	1.1	0.0	7.7	3.3	**82.4**	0.0
Zhu	2.8	0.9	7.0	3.3	1.1	4.9	1.6	8.7	3.4	0.5	**65.9**

Note: the confusion rates which are higher than 10% are (highlighted) in bold.

The human performance was measured in the experiment with a subset of the test images. The subset has 1 650 images selected randomly from the test dataset. The three subjects of the experiment are Chinese people who have a general sense of cooking. As shown in the table, the human recognition rates of only five cooking methods are higher than the corresponding rates of our system. Identifying or labeling cooking methods has certain subjectivity, and thus the recognition performance by a person without supervised training with the dataset might be affected. But this result can still show that our system can provide helpful reference information about the cooking methods, which is comparable to human performance.

Table 5. Recognition Rates of Cooking Methods by Human

Name	Recognition Rate (%)
Kao	**69.9**
Jian	**68.2**
Chao	73.8
Liang-ban	65.1
Zha	**68.6**
Zheng	**75.8**
Tang-cu	42.3
Shao	47.8
Tianpin	73.5
Lu	76.5
Zhu	**68.5**

Note: the recognition rates which are higher than the rates of the proposed system are highlighted in bold.

Table 6 shows five different test scores under different loss function parameter settings for the food ingredient detection task. In the experiments, we found that λ_n in the loss function of ingredient detection task could control the balance between the recall and the precision. When enlarging λ_n, the recall of ingredients would rise and the precision would decline, but the F1-score would not be influenced too much. The loss was numerical unstable when λ_n was too large, but there was not much difference in the training result between using 5.0 or a larger λ_n value. From the viewpoint of users, higher recall will be preferred, because removing detected ingredients is easier than adding new ones when using this system for food logging purpose. The 69.41% recall and the 60.74% precision demonstrate the effectiveness of our system, which can provide reference information for users.

Table 6. Test Scores under Different Loss Function Parameter Setings

Score Name	λ_n Value	
	1.0	2.0
Recall (TP/P)	48.29	69.41
Specificity (TN/N)	99.78	99.13
Harmonic mean (2/(P/TP+N/TN))	64.96	81.58
Precision (TP/(TP + FP))	80.94	60.74
$F1$ score (2 TP/(2 TP + FP + FN))	60.39	64.76

Note: TP: true positive; FP: false positive; TN: true negative; FN: false negative; P: condition positive; N: condition negative.

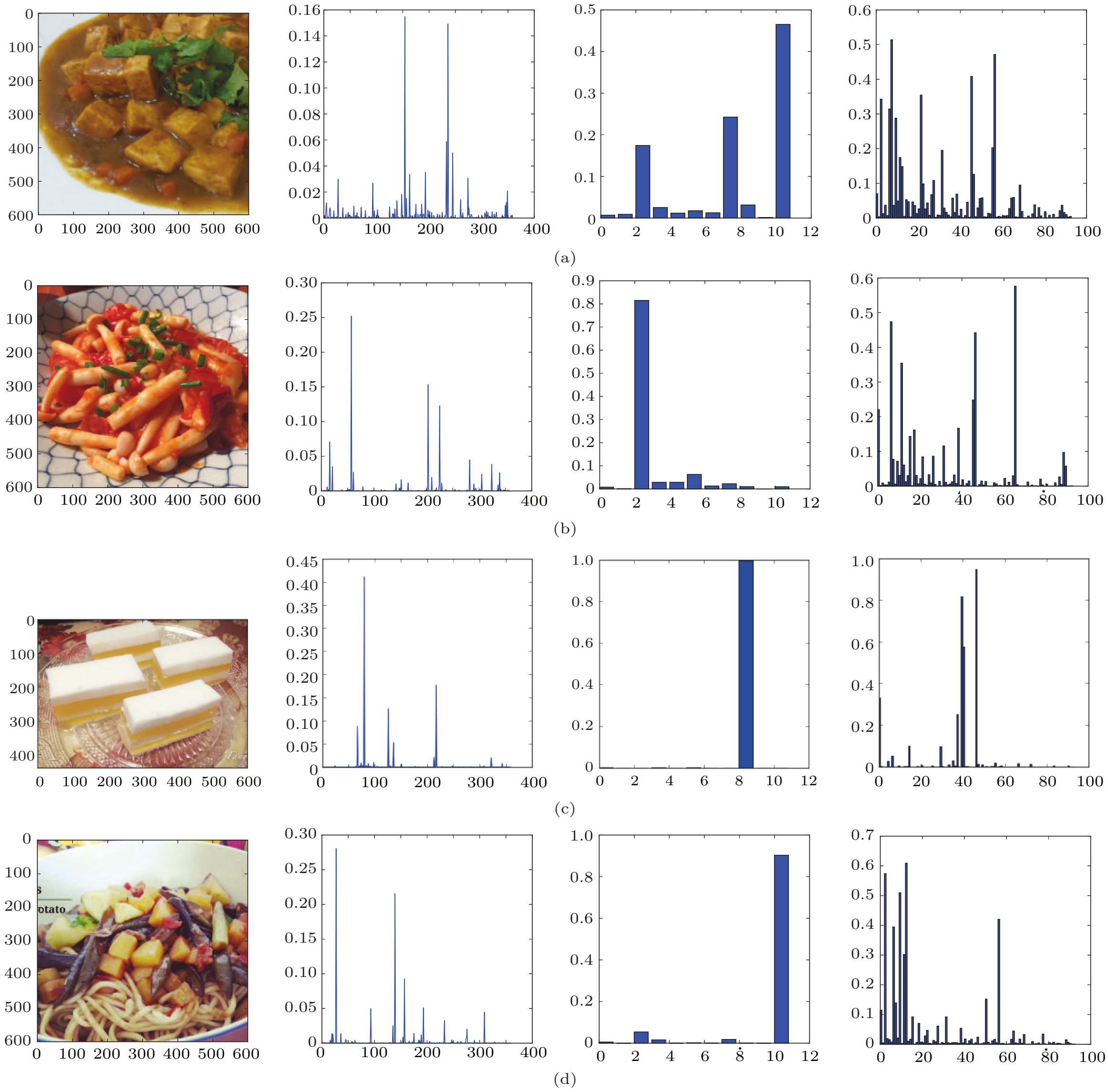
The ingredient recognition task can also be accomplished by first recognizing the food category and then get the ingredients according to the food category's information if the target food is contained in the dataset. We experimented with this method and calculated the test scores in the same way. For the predicted food category, if its label of an ingredient is 0 (ignored), this ingredient was also treated as non-detected. The scores are shown in Table 7. The specificity, the precision, and the $F1$ score (which is the harmonic mean of recall and precision) are higher than the scores of the multi-task method, but the recall and the harmonic mean of recall and specificity are lower. This result is similar to the result of the multi-task method under some loss function parameters set between 1.0 and 2.0. Users will prefer higher recall in practice as discussed in the last paragraph. Therefore, the proposed method with loss function parameter 2.0 is recommended. In addition, the advantage of the proposed method is that it can provide food ingredient information directly without a database mapping dish categories to ingredients. It can also be applied to images of dish categories which are not included in our training dataset.

Table 7. Test Scores of Category Recognition plus Category-Ingredients Mapping Method

Score Name	Percentage Value (%)
Recall (TP/P)	65.84
Specificity (TN/N)	99.38
Harmonic mean(2/(P/TP + N/TN))	79.20
Precision (TP/(TP + FP))	66.37
F1 score (2TP/(2TP + FP + FN))	66.17

5.3 Test for Application

We tested our system on some images that neither are in our dataset nor belong to any of the categories included in our dataset. The example images are shown in Fig.6.



(a)

(b)

(c)

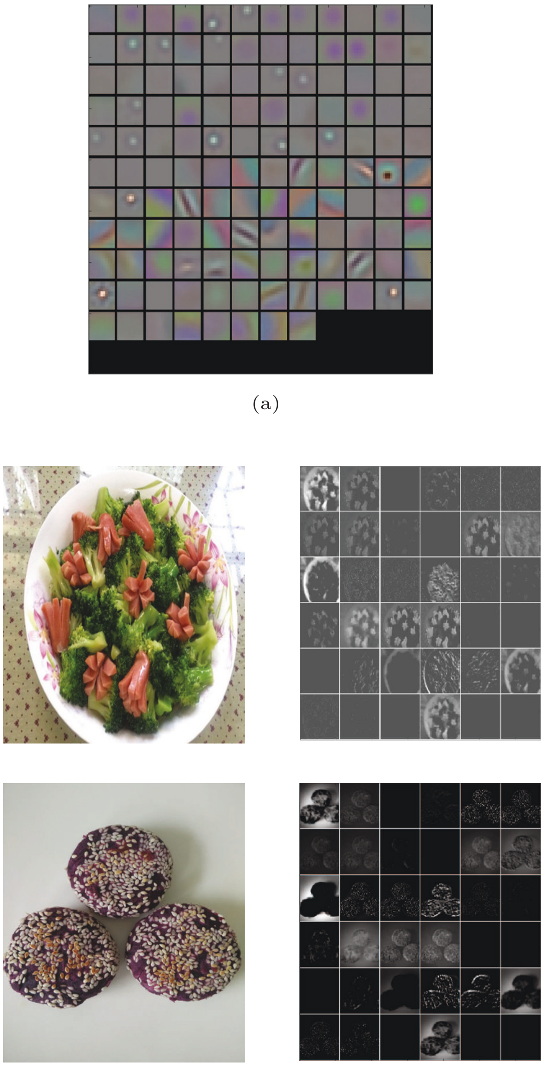
(d)

Fig. 6. New images tested. Original image, predictions from dish identification, the cooking method recognition, and ingredients detection are shown from the left to the right.

Our system detects curry and tofu from the first image, tomatoes and mushrooms from the second image, mangos and milk from the third image, and noodles, potatoes, pepper, and beef from the last image. The cooking methods of foods in these four images are precisely recognized as “Zhu” (boiled), “Chao” (multiple ingredients fried together, with little oil), “Tianpin” (dessert) and “Zhu” respectively.

5.4 Features Visualization

The convolution kernels from the first layer of our network and the convolution outputs examples are shown in Fig.7.


(a)

(b)

Fig.7. (a) Convolution kernels from the first layer of our network and (b) the convolution outputs examples.

We can observe from the figure that there are lots of color kernels which support the conclusion in [8]. The kernel visualization also shows that there are some small point shape kernels, which may detect the small ingredients in the food. As shown in the convolution outputs examples, the broccoli, sesame and scorched edges of purple sweet potato pies can stimulate particular response with the convolution kernels.

6 Conclusions

In this work, we built a dataset with more than 250 000 images of 360 categories of foods. We developed an automatic outlier elimination method employing deep convolutional features. A multi-task DCNN system was proposed and achieved 57.25% in the top-1 accuracy and 82.29% in the top-5 accuracy for the dish identification task. The result outperforms the traditional SVM method significantly. The system achieved 69.50% classification rate in the cooking method recognition task and 69.41% recall, 60.74% precision in the food ingredient detection task. Our system can be used to provide reference information for users with a variety of foods no matter whether they are in our dataset or not.

In the future work, we will try to combine food image segmentation in our multi-task DCNN system to further improve the performance and collect more data to enlarge the dataset with a wider range of foods so as to further train and test our network. We also intend to implement the proposed framework on mobile devices.

Acknowledgement We would like to thank all the anonymous reviewer for their valuable comments. We would also like to express our thanks to the staffs and the users of the xiachufang.com website for sharing the food images.

References

[1] Yang S, Chen M, Pomerleau D, Sukthankar R. Food recognition using statistics of pairwise local features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), June 2010, pp.2249-2256.

[2] Retna Swami M S S K, Karuppiah M. Optimal feature extraction using greedy approach for random image components and subspace approach in face recognition. *Journal of Computer Science and Technology*, 2013, 28(2): 322-328.

[3] Hall P, Cai H, Wu Q, Corradi T. Crossdepiction problem: Recognition and synthesis of photographs and artwork. *Computational Visual Media*, 2015, 1(2): 91-103.

[4] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In *Proc. the 26th Conference on Neural Information Processing Systems* (*NIPS*), December 2012, pp.1106-1114.

[5] Ghosh S, Laksana E, Scherer S, Morency L P. A multi-label convolutional neural network approach to cross-domain action unit detection. In *Proc. IEEE Int. Conf. Affective Computing and Intelligent Interaction* (*ACII*), May 2015, pp.609-615.

[6] Li S, Liu Z Q, Chan A. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. *International Journal of Computer Vision*, 2015, 113(1): 19-36.

[7] Chen M Y, Yang Y H, Ho C J, Wang S H, Liu S M, Chang E, Yeh C H, Ouhyoung M. Automatic Chinese food identification and quantity estimation. In *Proc. SIGGRAPH Asia 2012 Technical Briefs*, November 2012, pp.29:1-29:4.

[8] Kagaya H, Aizawa K, Ogawa M. Food detection and recognition using convolutional neural network. In *Proc. the 22nd ACM International Conference on Multimedia*, November 2014, pp.1085-1088.

[9] Kawano Y, Yanai K. Food image recognition with deep convolutional features. In *Proc. the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, September 2014, pp.589-593.

[10] Chen M, Dhingra K, Wu W, Yang L, Sukthankar R, Yang J. PFID: Pittsburgh fastfood image dataset. In *Proc. the 16th IEEE International Conference on Image Processing* (*ICIP*), November 2009, pp.289-292.

[11] Hoashi H, Joutou T, Yanai K. Image recognition of 85 food categories by feature fusion. In *Proc. IEEE International Symposium on Multimedia* (*ISM*), December 2010, pp.296-301.

[12] Joutou T, Yanai K. A food image recognition system with multiple kernel learning. In *Proc. the 16th IEEE International Conference on Image Processing* (*ICIP*), November 2009, pp.285-288.

[13] Matsuda Y, Hoashi H, Yanai K. Recognition of multiple-food images by detecting candidate regions. In *Proc. IEEE International Conference on Multimedia and Expo* (*ICME*), July 2012, pp.25-30.

[14] Kawano Y, Yanai K. Real-time mobile food recognition system. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops* (*CVPRW*), June 2013.

[15] Bosch M, Zhu F, Khanna N, Boushey C J, Delp E J. Combining global and local features for food identification in dietary assessment. In *Proc. the 18th IEEE International Conference on Image Processing* (*ICIP*), September 2011, pp.1789-1792.

[16] Maruyama T, Kawano Y, Yanai K. Realtime mobile recipe recommendation system using food ingredient recognition. In *Proc. the 2nd ACM International Workshop on Interactive Multimedia on Mobile and Portable Devices*, Oct. 29-Nov. 2, 2012, pp.27-34.

[17] Wang C, Huang K Q. VFM: Visual feedback model for robust object recognition. *Journal of Computer Science and Technology*, 2015, 30(2): 325-339.

[18] Yang X, Kim S, Xing E P. Heterogeneous multitask learning with joint sparsity constraints. In *Proc. the 23rd Annual Conference on Neural Information Processing Systems* (*NIPS*), December 2009, pp.2151-2159.

[19] Wang X, Fouhey D F, Gupta A. Designing deep networks for surface normal estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), June 2015, pp.539-547.

[20] Amer M, Goldstein M, Abdennadher S. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proc. the ACM SIGKDD Workshop on Outlier Detection and Description*, August 2013, pp.8-15.

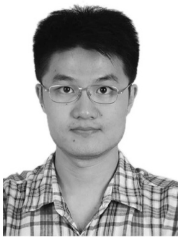
[21] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines. In *Proc. the 27th International Conference on Machine Learning* (*ICML*), June 2010, pp.807-814.

[22] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors. *Nature*, 1986, 323(6088): 533-536.

[23] Hinton G E, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R R. Improving neural networks by preventing co-adaptation of feature detectors. arXiv: 1207.0580, 2012. http://arxiv.org/abs/1207.0580, Mar. 2016.

[24] Branson S, Beijbom O, Belongie S. Efficient large-scale structured learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), June 2013, pp.1806-1813.

Xi-Jin Zhang is currently a Ph.D. student at Tsinghua University, Beijing. He received his B.S. degree in electronic and information engineering from Xidian University, Xi'an, in 2014. His research interests include image and video processing, computer vision, and machine learning.



Yi-Fan Lu is currently a Master student in the Department of Computer Science and Technology at Tsinghua University, Beijing. He received his B.S. degree in biology from Wuhan University, Wuhan, in 2013. His main research interests include computer vision and machine learning.

Song-Hai Zhang received his Ph.D. degree in computer science in 2007 from Tsinghua University, Beijing. He is currently an associate professor in the Department of Computer Science and Technology of Tsinghua University, Beijing. His research interests include image and video processing, and geometric computing.