Multimedia Tools and Applications (2019) 78:11155–11172
https://doi.org/10.1007/s11042-018-6695-9

Recognition of Chinese food using convolutional neural network

Jianing Teng[1,2] · Dong Zhang[1,2] · Dah-Jye Lee[2,3] · Yao Chou[2,3]

Received: 9 November 2017 / Revised: 7 September 2018 / Accepted: 14 September 2018 /
Published online: 21 September 2018

Abstract
Food recognition is the first step for dietary assessment. Computer vision technology is being viewed as an effective tool for automatic food recognition for monitoring nutrition intake. Of the many food recognition algorithms in the literature, Bag-of-Features model is a proven approach that has shown impressive recognition accuracy. In this paper, we propose a small and efficient convolutional neural network architecture for Chinese food recognition, which is more applicable for resources limited platforms. Our network architecture is designed to model and perform a pipeline of processing similar to the Bag-of-Features approach. The main advantage of the proposed architecture, like other convolutional neural networks, is its ability to unifiedly optimize the entire network through back propagation, which is critical to recognition accuracy. We further compare and correlate our architecture with the traditional Bag-of-Features model in an attempt to investigate the similarities between them and identify factors that influence the recognition accuracy. The proposed architecture with a 5-layer deep convolutional neural network achieves the top-1 accuracy of 97.12% and the top-5 accuracy of 99.86% on a newly created Chinese food image dataset that is composed of 8734 images of 25 food categories. Our experimental result demonstrates the feasibility of applying the proposed compact CNN architecture to a challenging problem and achieve real-time performance.

Keywords Chinese food · Recognition · Convolutional neural network · Bag-of-features model

1 Introduction

Statistics show that approximately 600 million adults and 100 million children worldwide were obese in 2015 [4]. Obesity increases the possibility of contracting diseases or affecting

✉ Dong Zhang
zhangd@mail.sysu.edu.cn

[1] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China

[2] SYSU-CMU Shunde International Joint Research Institute, Shunde, Foshan, China

[3] Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT, USA

health conditions such as cardiovascular problems, type 2 diabetes, certain types of cancer, osteoarthritis and depression [13, 19]. It is regarded as one of many serious health problems in most developing and developed countries. The American Medical Association even classified obesity as a disease in 2013 [2, 22]. Obesity may be a result of any combinations of excessive food intake, lack of physical activities, endocrine or mental disorders [5], and heredity. It is considered preventable by some experts through social changes and personal choices. Among treatments for obesity, an accurate measurement of the daily nutrition amount or portion of food intake is an effective way to control or monitor obesity. It provides valuable insights into the occurrence of disease and subsequent approaches for mounting intervention programs [34].

Traditional methods to assess the amount of daily food intake are based on human visual recognition and self-reporting through the use of questionnaires and structured interviews [6, 34] (e.g., 24-Hour Dietary Recall and Food Record), which require the help from the nutritionist and cooperation from the obese patient. Investigations found that the accuracy of food intake data obtained by self-reporting is much dependent on experience and is prone to being underestimated [6, 31]. The accuracy of self-reporting methods is often affected by human errors.

Recent advancements in hardware, image processing, and pattern recognition make computer vision a viable technique for personal health assistance. Many promising works based on computer vision have been proposed to address the food recognition and assessment problem. With the widespread use of smart phones which are all equipped with Internet connection, high resolution cameras, large memory capacity, and a powerful processor, computer vision based food recognition and amount estimation algorithms can be readily implemented and run on a smart phone as a portable device to automatically acquire accurate diet assessment data [29].

A computer vision based diet assessment scheme usually pre-processes the acquired food image and segments the food regions from the background. An elaborate algorithm is employed to recognize the type of the food in the image. The nutrition content of the food can then be retrieved by accessing public nutrition datasets, e.g., Food and Nutrient Database for Dietary Studies (FNDDS) [25], through Internet connection, or looking up a nutrition database stored locally on the phone. The amount or the weight of food can either be estimated by stereo vision techniques or directly weighted by a scale mounted beneath the food container. The energy or the calories of the food can be calculated using the type, the nutrition content, and the amount of the food.

As a unique application of object recognition, food recognition based on efficient image features plays a critical pole in the process of diet assessment. Many human designed descriptors are proposed to characterize the specifics of the food image, e.g., size, color, texture, shape, and context-based features. The characteristics or the features of food are input to classifiers such as artificial neural networks (ANN), support vector machine (SVM), or Adaboost for classification. In recent years, the Bag-of-Features (BoF) model have been employed to address the challenges of food recognition. Methods based on the BoF model learn a dictionary of visual words from the training sets of food images to provide an accurate local description of food image patches. They represent the food image with a histogram of visual words which are designed to describe particular visual patterns. BoF methods ignore the order of local descriptors corresponding to a learned word dictionary and consider only the frequency they appear to form a global representation of the food image. The concept of BoF model adequately fits for solving the problem of food recognition and has obtained promising results.

In recent years, Convolutional Neural Networks (CNNs) and its improvements [18, 27, 33], as a type of highly parallelized method, has achieved great success in many computer vision applications such as classification, segmentation, object detection, and edge detection. In this paper, we propose a compact and efficient CNN architecture which is suitable for resources limited platforms for Chinese food recognition, and investigate the relationship between the BoF model based pipeline and CNNs. The proposed CNN architecture does not learn the dictionaries explicitly for modeling the patch space. Instead, it implicitly accomplishes the learning task similar to the BoF with hidden layers. The filters or kernels formulated as the convolutional layers in CNNs serve for the purposes similar to the atoms in the dictionary of BoF. Similarly, the process of generating a histogram to form global features in the BoF is equivalent to the pooling layers in CNNs. As opposed to optimizing feature extraction, feature representation, and classification separately and independently in the BoF methods, the proposed CNN architecture performs better than the BoF methods by taking the advantage of the intrinsic unified optimization of the entire network.

The contributions of this work include three aspects: (1) A small and efficient convolutional neural networks architecture for Chinese food recognition; (2) Comparison and correlation between the proposed CNN architecture and the BoF model; (3) Demonstration of the effectiveness and accuracy of our CNN architecture.

2 Related works

Traditional human visual recognition and self-reporting methods are not able to provide accurate and instantaneous evaluation on dietary intake, Shroff et al. propose a mobile phone-based fast-food recognition system for calorie monitoring [24]. Food image is captured by a cell phone, while the recognition of the food image is performed on a remote server. Because of the varying and implicit nature of certain foods, contextual information from the user and system are required as auxiliary information for traditional image recognition techniques. Zhu et al. develop a volume estimation method which utilizes camera parameter estimation and model reconstruction to estimate the volume of the food items [34]. To provide standard baselines for research work in the field of food recognition, Chen et al. create a visual dataset of 101 foods from 11 fast food chains [6]. A larger visual dataset with nearly 5000 food images that are organized into 11 classes is created by Anthimopoulos et al. [3]. Up to now, more than 10 datasets have been created for the evaluation of food recognition algorithms [9]. A group of researchers claim to have created a dataset of Chinese food but its link is not yet available [8]. An accurate and efficient method that is able to recognize the special characteristics of Chinese food is needed as Chinese food becomes a worldwide favorite. A typical reason for this need is that Chinese food prefers to retain the ingredient's original color, smell, and taste. Color and texture features of the food ingredients play a more important role in the recognition process of Chinese food than foods from other cultures.

Many research papers in the literature address food classification as a unique pattern recognition problem while others focus on constructing applicable solutions for dietary evaluation. Extracting efficient features and identifying suitable classifiers are the two main challenges for pattern recognition and have attracted great attention from researchers. Martin et al. describe a selected food image region only with the color features, and use Mahalanobis distance for classification [20]. Zhu et al. extract the average value of the pixel intensity along with two color components and Gabor filtering based texture features to characterize the local specifics of the segmented food image region [34]. A feature vector composed of 48 texture features and 3 color features is sent to a support vector machine (SVM) to classify the food image region. Anthimopoulos et al., after a comprehensive comparison, use an optimized system to compute dense local features using the scale-invariant feature transform (SIFT) in the HSV color space. A visual dictionary of 10,000 visual words is built with the hierarchical K-means clustering [3]. A linear SVM is used as the classifier and obtains the classification accuracy of approximately 78%. Yang et al. investigates the spatial relationships between different ingredients and describe the food item by calculating pairwise statistics between local features computed over a soft pixel-level segmentation of the image [30]. The statistics accumulated in a multi-dimensional histogram is used as a feature vector for an SVM.

Among the reported methods, BoF based algorithms that construct global features from the local patches of the food image are employed more often than others in recent years. Kawano et al. extract the color histogram and SURF-based bag-of-features to describe the segmented food image region and use a linear SVM for classification [14]. Giovany et al. describe the image patch with SIFT-feature and formed statistical features for classification. K-Dimensional tree and backpropagation neural network are used as the classifier to recognize Indonesian food [12]. Farinell et al. treat food recognition as a problem of texture classification and employ a bag of visual words model to represent the food image [11]. They process the food image with a bank of rotation and scale invariant filters. The responses of the filters are clustered by k-means to build a code book of textons. The learned class-based textons are collected in a single visual dictionary and the food image is represented as distributions of visual words. SVM is used for classification.

Convolutional Neural Networks have shown great success in many pattern recognition applications. Kawano et al. combine features generated from Deep Convolutional Neural Network and conventional hand-crafted image features to obtain high food recognition accuracy [15]. Zhang et al. use a five-layer convolutional neural network for food recognition and achieve an accuracy of 80.8% on the fruit dataset and 60.9% on the multi-food dataset [32]. Phat et al. compare CNNs with traditional methods with hand-crafted features for Vietnamese food recognition and conclude that CNNs outperform hand-crafted techniques by a significant margin [26]. Ciocca et al. perform comprehensive comparisons among different visual descriptors, including high dimensional color feature vectors, Gabor features, dual tree complex wavelet transform features, and CNNs features [9]. Test results show that the CNNs-based visual descriptors have the best recognition accuracy.

Although CNNs are generated by an iterative optimization process through backpropagation, most steps of CNNs are closely related to traditional pattern recognition procedures [10, 23]. For example, the convolutional layers of CNNs serve for the purpose similar to image filtering by which the features of edge or texture are extracted from a selected patch. The fully-connected layers of CNNs can be regarded as performing a linear transformation on its input information. The concept of pooling layers can also be considered as the counterpart of feature construction in the BoF method [1]. Considering the promising results obtained by the BoF model and the success of CNNs, this paper proposes a CNN architecture for Chinese food recognition and compares and correlates the two approaches in an attempt to identify factors that influence recognition accuracy.

3 A CNN structure for Chinese food recognition

A typical BoF method describes each patch of the food image with designed descriptors which characterize the local specifics of the input image. Local descriptors of the food image are used to generate a visual dictionary by means of learning algorithms. By using clustering or sparse coding, each local descriptor of the food image is represented as a combination of several visual words from the learned dictionary. The food image is represented by a histogram of visual words, which ignores the order of the visual words and considers only how frequently they appear. The histogram of visual words works as the global features of the food image. BoF methods use the global features of the training images and their corresponding labels to train a classifier. BoF methods obtain promising results in food recognition. However, the BoF pipeline has rarely been considered in a unified optimization framework [15]. In this paper, we propose to use a small and efficient Convolutional Neural Network to recognize Chinese foods and investigate the relationship between the sparse coding based BoF model and the Convolutional Neural Network to show the BoF pipeline serves for the purpose similar to the structure of CNNs. CNNs are able to achieve superior performance in food recognition because all steps in CNNs that involve the filters are optimized in a unified framework during training.

As shown in Fig. 1, the proposed CNN architecture consists of two phases of operations, i.e., feature extraction and feature combination for classification. The feature extraction phases contains k convolutional layers, each of which is followed by a Rectified Linear Unit (ReLU). Suppose the input to the feature extraction phase is an image resized to $N_1 \times N_1$. For the i-th layer of convolution, filters with the size of $n_i \times n_i$ are employed to extract the features of its corresponding layer, thus forms m_i feature maps with the size of $(N_i - n_i + 1) \times (N_i - n_i + 1)$.The feature combination and classification phase is formed with fully-connected layers, which combine the features extracted by the feature extraction phase and output a distribution over different class labels through a layer of Softmax.

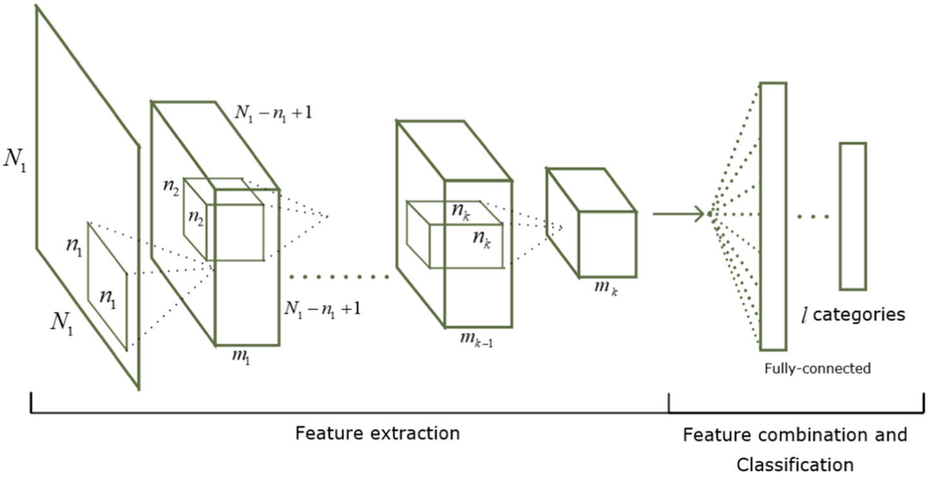


Fig. 1 The proposed CNN architecture for food recognition

3.1 Feature extraction

The feature extraction phase computes the responses of the learned filters to the input image. For the convenience of description, we suppose the input image is with the size of$N_1 \times N_1$. The first layer of convolution employs a set of m_1 learned filters of the size of $n_1 \times n_1$ to generate m_1 feature maps. Suppose the convolution operation is performed within the range of the input image and the stride of the convolution is set to one, the output will be m_1 size-reduced (depending on the size of the filters) feature maps, each of which is with the size of $(N_1 - n_1 + 1) \times (N_1 - n_1 + 1)$. The convolution operation is shown in Fig. 2a.

If the input image $\mathbf{x}$ is rearranged from an $N_1 \times N_1$ matrix into a $N_1^2 \times 1$ vector, the convolution of a filter with the input image to generate a feature map can be expressed as the production of a matrix, $\mathbf{M}^T$, and $\mathbf{x}$, as shown in Fig. 2b. Each row of $\mathbf{M}^T$is composed of a shifted version of the filter kernel (gray cells in Fig. 2b) and some filled zeros (white cells in Fig. 2b). Thus, $\mathbf{M}^T\mathbf{x}$ represents the production of $\mathbf{x}$ and a series of shifted versions of the filter kernel. The result of production has $(N_1 - n_1 + 1)^2$ elements, which represents the filter response to the input signal $\mathbf{x}$.

The generation of a feature map (one filter convolves with the input image) of the first convolutional layer is expressed as the production of a matrix, $\mathbf{M}^T$, and $\mathbf{x}$. The generation of m_1 feature maps can also be expressed by the production of a matrix $\mathbf{W}_1^T$ and the input $\mathbf{x}$ arranged as a 1D signal. The convolution matrix $\mathbf{W}_1^T$ is the concatenation of the matrices that are used to generate individual feature maps. As shown in Fig. 2c, the number of columns of $\mathbf{W}_1^T$ equals to the length of input signal N_1^2 for the multiplication between $\mathbf{W}_1^T$ and $\mathbf{x}$. The number of rows of $\mathbf{W}_1^T$ is $m_1(N_1 - n_1 + 1)^2$ since m_1 filter kernels are employed and the convolution is performed within the range of the input image. We construct the rows of $\mathbf{W}_1^T$ according to the shift of the filter kernels in the same manner as convolution. In Fig. 2c, as an example, the number of filter kernels m_1 is 5, the size of each filter kernel is 3 × 3 ($n_0 = 3$), and the size of input image is 6 × 6 ($N_1 = 6$). The top 5 rows of $\mathbf{W}_1^T$ show the filters without shift, while the next 5 rows show all the filters right shifted one pixel, and so on. Thus, the convolution operation of the m_1 filters with the input image is expressed as the measurement result of $\mathbf{x}$ by $\mathbf{W}_1$, which is equivalent to the inner product between the input image and the visual dictionary in the coding step of the BoF pipeline.

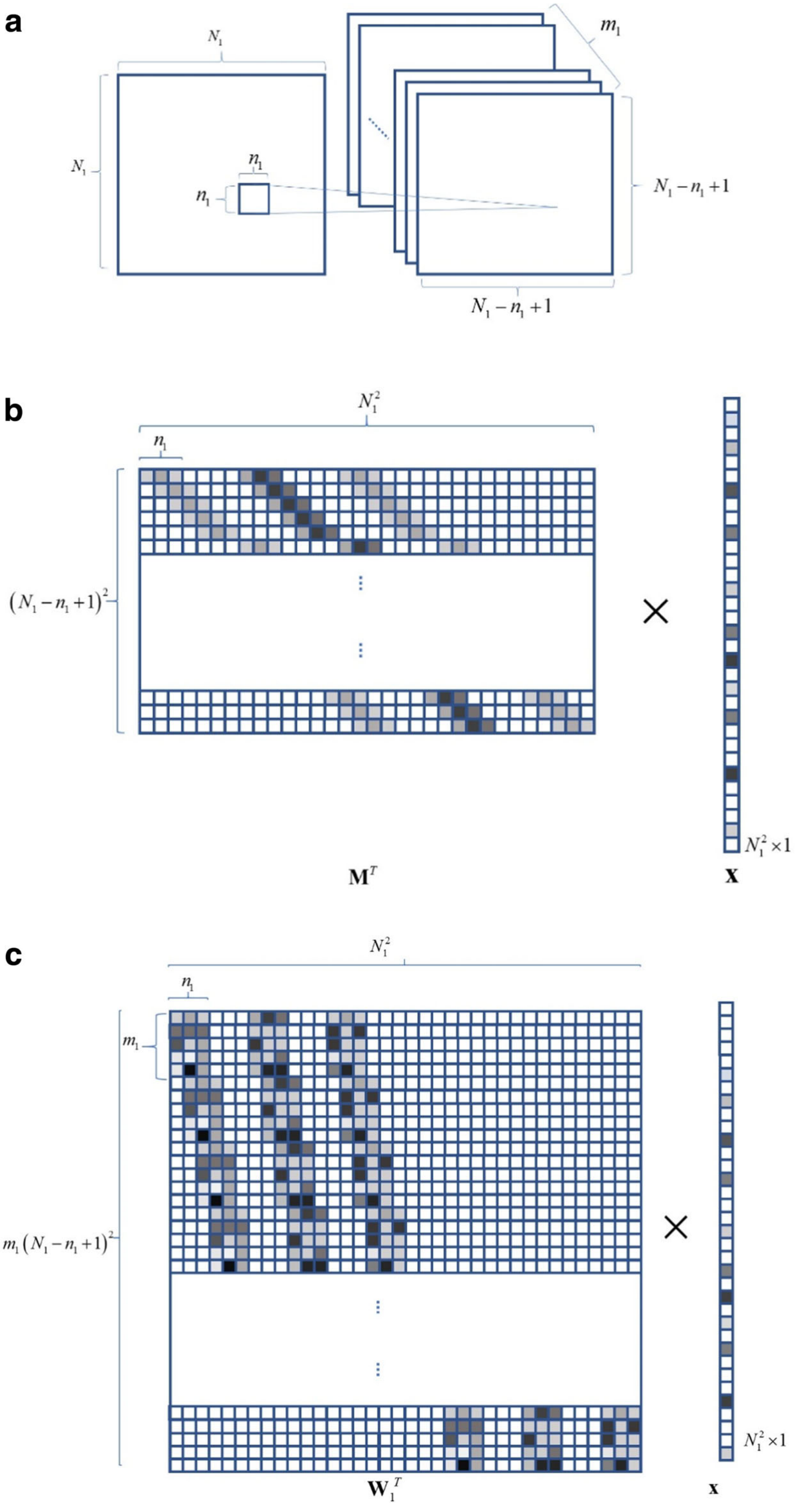
The first convolution layer is followed by a nonlinear function ReLU, which provides a bias denoted by $\mathbf{b}_1$. Thus, the operation of the first layer of CNN can be expressed as Eq. (1)

$$f(\mathbf{x}) = \mathrm{ReLU}\left(\mathbf{W}_1^T\mathbf{x} + \mathbf{b}_1\right) = \max\left(0, \mathbf{W}_1^T\mathbf{x} + \mathbf{b}_1\right) \tag{1}$$

The structure of the feature extraction phase can be extended to multiple layers to acquire higher-level abstracted features. Suppose that we have k convolutional layers, the operation of feature extraction phase is expressed as Eq. (2).

$$f(\mathbf{x}) = \mathrm{ReLU}\left(\mathbf{W}_k^T \cdots \mathrm{ReLU}\left(\mathbf{W}_2^T \mathrm{ReLU}\left(\mathbf{W}_1^T\mathbf{x} + \mathbf{b}_1\right) + \mathbf{b}_2\right) \cdots + \mathbf{b}_k\right) \tag{2}$$

Fig. 2 Explanation of the function of a convolution layer. **a** The first convolution layer uses m_1 kernels and generates m_1 feature maps; **b** Each individual convolution of a learned filter with the input image (the generation of a feature map) can be viewed as the production of a matrix, $\mathbf{M}^T$, and the input image signal $\mathbf{x}$; **c** The operation of the entire convolution layer (or the generation of m_1 feature maps) can be viewed as the production of a convolution matrix $\mathbf{W}_1^T$ and the input $\mathbf{x}$ ▶


a
N_1
N_1
n_1
n_1
m_1
$N_1 - n_1 + 1$
$N_1 - n_1 + 1$
b
N_1^2
n_1
$(N_1 - n_1 + 1)^2$
×
$N_1^2 \times 1$
$\mathbf{M}^T$
$\mathbf{x}$
c
N_1^2
n_1
m_1
$m_1 (N_1 - n_1 + 1)^2$
×
$N_1^2 \times 1$
$\mathbf{W}_1^T$
$\mathbf{x}$

where $\mathbf{W}_k^T$ denotes the k-th convolutional matrix constructed from m_k filters of length $m_{k-1}n_k \times n_k$. The kernel size is $n_k \times n_k$ for the k-th convolutional layer and m_{k-1} is the number of feature maps the $(k-1)$-th convolutional layer output. The matrix size for the k^{th} layer is $m_{k-1}N_k^2 \times m_k(N_k - n_k + 1)^{\ 2}$. Figure 3 shows the structure of the feature extraction phase when k layers are employed.

3.2 Feature combination and classification

The feature combination and classification phase is composed of fully-connected layers. The purpose of using fully-connected layers is to combine high-level features extracted from the previous phase to classify the input image into its corresponding class as specified by the training dataset. This operation combines the extracted features and maps them to l discrete values, where l represents the number of categories or classes to be classified. During training, the goal is to minimize the defined cross-entropy classification loss, as shown in Eq. (3),

$$\mathrm{CE} = -\frac{1}{M}\sum_{i=1}^{M}\log\left(p_{i,L_k}\right) \tag{3}$$

where M is the number of samples. Suppose a sample $\mathbf{x}_i$ with the label L_k is provided to the network, p_{i,L_k} is the probability that the softmax predicts $\mathbf{x}_i$ belongs to the class labelled withL_k. This step is carried out in the fully-connected network as shown in Fig. 1.

4 Experiments and discussion

We created an image dataset of Chinese food for testing the proposed CNN architecture. The performance of the proposed network was verified with a 5-fold cross validation in experiments. We investigated different configurations and the parameters of the architecture, and their corresponding influence on classification performance. We also compared the performance of the proposed architecture with four BoF methods.

4.1 Dataset

The images in the newly created Chinese food dataset were captured with three mobile phones: HM NOTE 1S CMCC of XIAOMI with Android 4.4.4 (KTU84P), iPhone 6 of Apple with IOS 9.3.2 (13F69), and MEIZU MX5 with Android 5.1. All food images were taken at the canteen of SYSU-CMU Shunde International Joint Research Institute under nature lighting,

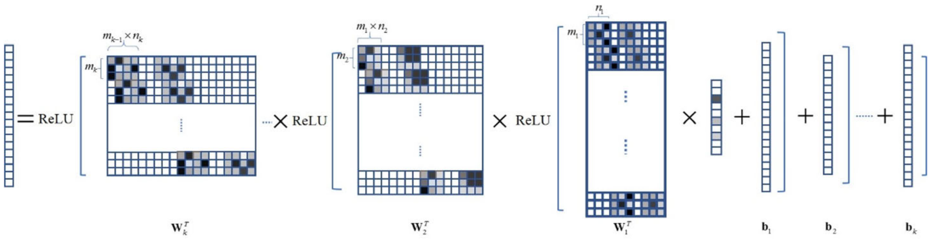


Fig. 3 The illustration of Eq. (2) for a multi-layer feature extraction architecture

and in stainless steel food containers. The photos were taken randomly on different dates and at different times, which provided more diverse brightness, view angle, and background. The constructed dataset of Chinese food image contains 8734 images from 25 food categories. Each category had over 300 samples. The captured Chinese food images were segmented and resized to 256 × 256 before training. Figure 4 shows food image examples in the dataset. Figure 5 lists the number of samples for each food class. We use the letter 'c' following the class name to indicate the class number.

4.2 Model and performance

We trained our model using a batch size of 256 for 2000 iterations corresponding to roughly 58 epochs. This took 8 min on one NVIDIA GTX Titan X(Pascal) GPU. We evaluated the performance of our CNN architecture in terms of classification accuracy and visualized the learned filters of the first convolutional layer to show what important features the architecture was able to extract. We also investigated the impact of the network depth on classification accuracy. Furthermore, we evaluated our model by calculating precision rate and recall rate.

4.2.1 Classification accuracy

Our first CNN architecture, CNN_5, contained three convolutional layers and two fully-connected layers. The first convolutional layer used 64 kernels of size 9 × 9 with a stride of 4 pixels, followed by a 3 × 3 max pooling layer with a stride of 2. It produced 64 feature maps of size 31 × 31 after the max-pooling. The second convolutional layer used 128 filters of size 5 × 5 with a stride of 1 pixel and padding of 2 pixels on the edges. The corresponding pooling layer had the same specification as the first one. The size of feature maps became 15 × 15 after the second convolutional-max pooling layer. The third convolutional layer used 256 filters of size 3 × 3 with stride of 1. The pooling layer followed the third convolutional layer also had the same configuration as the previous two. The output of the third pooling layer finally provided high level visual features of the input food image for classification. Our first CNN architecture had two fully-connected layers after the three convolutional layers. The first fully-connected

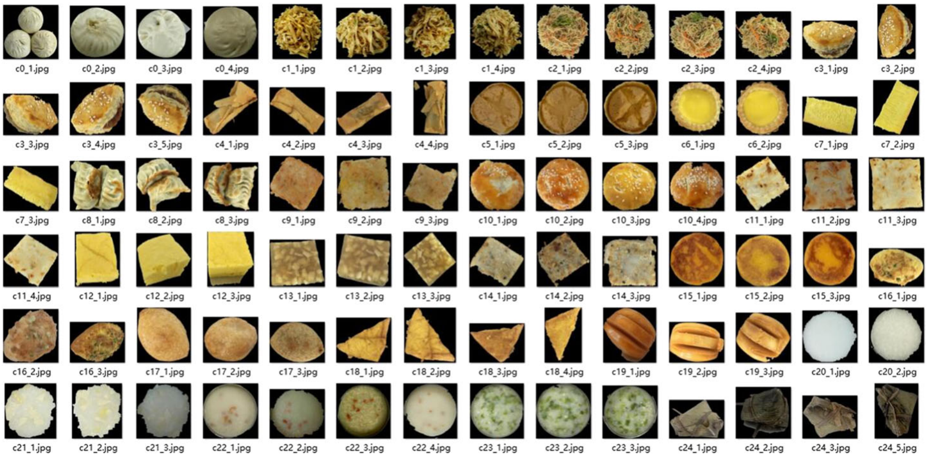


Fig. 4 Sample images of the dataset

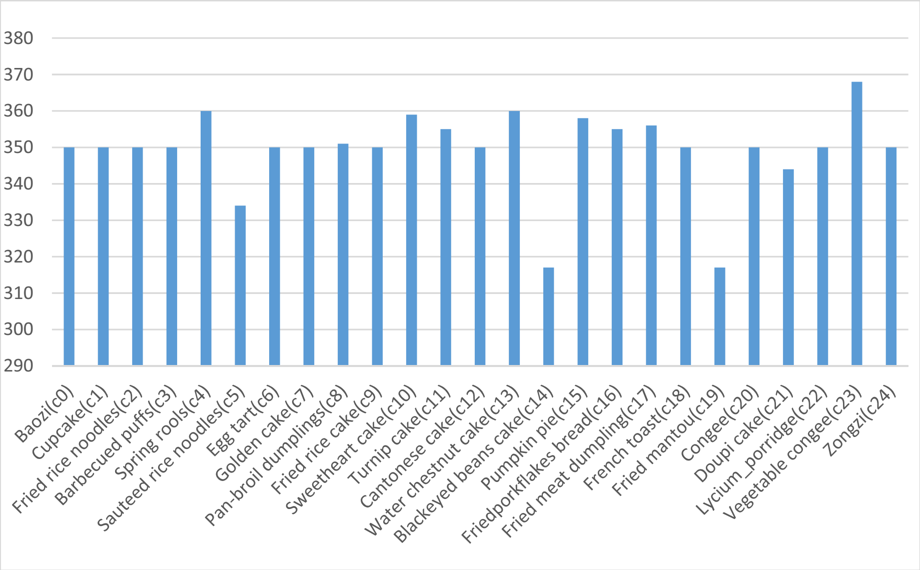


Fig. 5 Number of samples for each food class

layer had 512 output neurons followed by a dropout layer. The second fully-connected layer had 25 output neurons and was followed by a softmax output layer with 25 outputs for the 25 food categories.

We evaluated the performance of the proposed architecture in terms of recognition accuracy within the top-1 and top-5 candidates by employing 5-fold cross validation. As shown in Table 1, five-fold classification top-1 accuracy was 97.22% and top-5 accuracy was 99.87%.

4.2.2 Visualization of learned filters

Figure 6 shows the kernels learned by the first convolutional layer. The network learned a variety of kernels with different information, e.g., color and edge detectors in different directions, which extract low-level features of the food image.

4.2.3 Impact of network depth

Normally, the performance of a CNN benefits from the increased network layers. To compare the performance of CNNs with different number of layers, we designed another two architectures with three and seven layers, and named them CNN_3 and CNN_7, respectively.

Table 1 Five-fold classification accuracy in top-1 and top-5 of CNN_5

Fold number	Top-1	Top-5
Fold 1	97.84%	100.00%
Fold 2	95.80%	99.80%
Fold 3	97.72%	100.00%
Fold 4	97.54%	99.61%
Fold 5	97.22%	99.93%

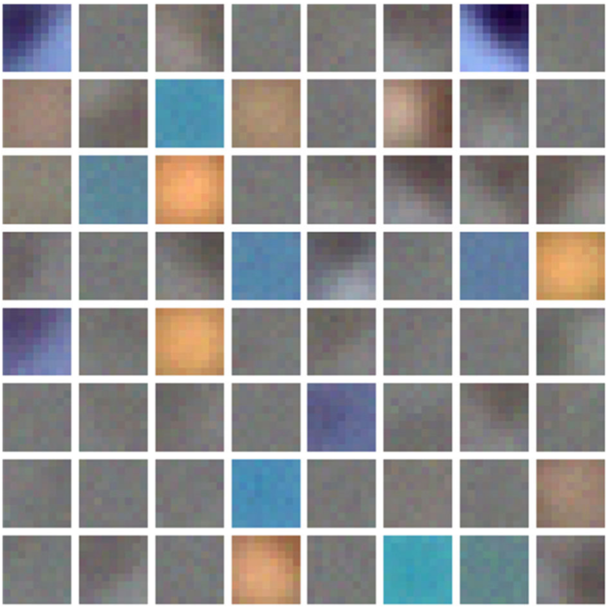


Fig. 6 Filters learned by the first convolutional layer

CNN_3 contained one convolutional-max pooling layer and two fully-connected layers. The convolutional layer had 48 kernels with the size of 9 × 9 pixels. The first fully-connected layer of CNN_3 contained 128 output neurons, while the second fully-connected layer had 25 output neurons. CNN_7 had four convolutional layers followed by three fully-connected layers. The four convolutional layers were configured to have 256, 512, 768, 512 kernels, whose sizes were set to 11 × 11, 5 × 5, 3 × 3, and 3 × 3, respectively. The three fully-connected layers of CNN_7 were set with 2048, 1024 and 25 output neurons respectively. All these networks used Rectified Linear Units (ReLUs) as their activate functions and shared the same configured parameters for the max-pooling layers, i.e., the kernel size and stride were set to 3 × 3, and 2, respectively.

We compared the performance of these three models and investigated the impact of the depth of the network. The results are listed in Table 2. Table 2 shows that CNN_7 obtained the best performance with top-1 accuracy of 98.96% and top-5 accuracy of 99.95%. CNN_5 achieved the second best performance of accuracy, 97.12% for top-1 and 99.86% for top-5. The obtained accuracy of CNN_3, which has only one convolutional layer, was the worst in the experiment. Experiment results show the model size of CNN_7 is the largest (65,756 KB), while the model size of CNN_3 is the smallest (2, 027 KB) in the comparison. The model size of CNN_5 is 27,155 KB.

Table 2 Comparison of classification accuracy of different network depths

Model	Depth	Top-1	Top-5
CNN_3	3 layers	74.83%	97.39%
CNN_5	5 layers	97.12%	99.86%
CNN_7	7 layers	98.96%	99.95%

Although the performance was improved by increasing the depth of the network, a CNN with moderate number of layers is preferred if real-time performance is desired. As shown in Table 2, CNN_5 and CNN_7 obtained comparable accuracy. The training time for CNN_5 was much faster than CNN_7 because fewer layers were required to be trained.

4.2.4 Precision and recall evaluation

We evaluated the performance of the trained CNN_5 model in terms of precision, recall and accuracy, using a test set which was randomly chosen from our dataset. The definitions of precision, recall and accuracy are formulated as Eq. (4) to (6).

$$precision = \frac{T_p}{T_p + F_p} \tag{4}$$

$$recall = \frac{T_p}{T_p + F_n} \tag{5}$$

$$accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{6}$$

where T_p denotes the number of true positive, F_p the number of false positive, F_n the number of false negative, and T_n the number of true negative. Table 3 shows our evaluation. For each input image, the proposed CNN architecture outputs a probability corresponding to each class of food in the test. If the highest probability for a class is greater than the threshold τ, the image will be classified to be that class. The threshold can be adjusted for different tradeoffs between false positives and false negatives. Table 3 shows the best accuracy of 96.75% was achieved with a 100% recall rate and a 96.80% precision rate.

4.3 Comparison and discussion

In this section, we compare CNN_5's performance to a series of BoF methods and three mainstream networks: AlexNet, GoogLeNet and NIN, on our dataset of food images. We also demonstrate the performance of the proposed CNN_5 on the UEC-FOOD100 dataset [21].

The BoF methods were implemented following the scheme provided in [3]. Color histograms (HistRGNorm), and four texture features, including Gabor feature, HOG feature,

Table 3 Performance evaluation of CNN_5 model for different threshold values τ

Threshold τ	0.99	0.90	0.80	0.60	0.40
True positives	1329	1558	1607	1649	1665
True negatives	52	36	27	13	1
False positives	4	20	29	43	55
False negatives	336	107	58	16	0
Precision	99.70%	98.73%	98.23%	97.46%	96.80%
Recall	79.82%	93.57%	96.51%	99.03%	100%
Accuracy	80.24%	92.62%	94.94%	96.57%	96.75%

LBP feature, and SURF feature, were employed as local feature descriptors. Rather than used k-means or hierarchical k-means, K-SVD [1] was employed in the BoF methods to learn the visual dictionary for representing local descriptors. As shown in Table 4, CNN_5 achieved a superior classification accuracy of 97.12% among the five methods compared.

We also compared CNN_5 with three other mainstream networks: AlexNet, GoogLeNet, and NIN [17, 29] on our dataset. Experiment results show that AlexNet achieved the top-1 accuracy of 97.32% while GoogLeNet achieved top-1 accuracy of 97.84%, which are comparable with the performance of the proposed CNN_5 architecture. Besides the classification accuracy, we also compared their trained model size. CNN_5's model size is 27,155 KB, approximately 10% of the AlexNet (275,791 KB) and less than half of the GoogLeNet (57,872 KB). The model sizes of AlexNet and GoogLeNet are much larger than CNN_5 because of the complexity of their architectures. AlexNet has eight layers including five convolutional layers and three fully-connected layers. The last four convolutional layers of AlexNet all have more than 256 convolutional kernels and its three fully-connected layers have 4096, 4096 and 25 neurons respectively for classifying 25 classes. Although it applies pooling and drop-out operation to reduce the number of parameters, AlexNet is a complex model compared to CNN_5. Similar analysis can be done on the GoogLeNet. As for NIN, it has the smallest model size (8103 KB) among the above four networks because of its simple architecture, but its classification accuracy is the worst (94.67%), which is 2.45% lower than CNN_5. For food recognition on a resource-limited system, the memory requirement and computational cost are critical [29]. Compared to the other three networks, the proposed CNN_5 architecture obtained promising recognition accuracy with a much smaller model size (Table 5).

We tested the proposed CNN_5 architecture on UEC-FOOD100 dataset [21] as well. UEC-FOOD100 is a 100-class food image dataset including around 100 or more images for each category and bounding box information which indicates food location within each photo. We compared the test result of CNN_5 with other reported works [15, 16] using UEC-FOOD100. As shown in Table 6, the proposed CNN_5 achieved the top-1 accuracy of 60.90% and top-5 accuracy of 86.15%, which are higher than the accuracy of other compared methods except "FV + DCNN" [15]. "FV + DCNN" integrated DCNN features and hand-crafted features that are specialized for the dataset. As stated in [15], classification using only CNN feature would not work well in dataset without enough samples for training, that is the reason it combined hand-crafted features for classification on UEC-FOOD100, which has only approximately 100 images in each category (total 12,905 images for 100 categories). Considering our method does not include any hand-crafted features, we compared CNN_5 to the DCNN of [15] and obtained a better performance in terms of classification accuracy. Additionally, DCNN has five convolutional layers and three fully-connected layers, and is much more complex than our proposed CNN_5. The

Table 4 Comparison of recognition accuracy of CNN_5 with four BoF methods

Method	Accuracy
CNN_5	97.12%
HistRGNorm + Gabor	83.90%
HistRGNorm + HOG	84.00%
HistRGNorm + LBP	83.85%
HistRGNorm + SURF	85.00%

Table 5 Comparison of accuracy and model size on our dataset with other works

Method	Accuracy	Model size
CNN_3	74.83%	2,027 KB
CNN_5	97.12%	27,155 KB
CNN_7	98.96%	65,756 KB
NIN	94.67%	8,103 KB
GoogLeNet	97.84%	57,872 KB
AlexNet	97.32%	275,791 KB

experiment shows that our CNN_5 model can keep a good balance between complexity and recognition accuracy on a more challenging dataset.

Chen et al. [7] and Yanai et al. [28] reported good experimental results on UEC Food-100 dataset. The employed AlexNet contains an 8-layer architecture, and the employed VGG has a 16-layer architecture [7]. The method presented [28] is a fine-tuned DCNN with 8 layers which was pre-trained with 2000 categories in the ImageNet including 1000 food-related categories. All three proposed architectures are deeper and more complex than our CNN_5. We performed experiments by deepen our model to 7 (CNN_7) and 9 (CNN_9) layers, and tested them on the UEC FOOD-100 dataset. CNN_7 model, as descripted in Subsection 4.2.3, achieved top-1 accuracy of 66.41% and top-5 accuracy of 88.40%. CNN_9 had six convolutional layers and three fully-connected layers. The five convolutional layers were configured to have 96, 256, 384, 384, 256, 256 kernels, whose sizes were set to11 × 11, 5 × 5, 3 × 3, 3 × 3, 3 × 3 and 3 × 3, respectively. It has three fully-connected layers with 4096, 4096 and 100 output neurons respectively. The CNN_9 model achieved top-1 accuracy of 75.80% and top-5 accuracy of 92.95%, which were comparable to the AlexNet reported in [7] and Yanai et al. in [28]. VGG still holds the highest performance for its complex network architecture. The experiment results show that our method is able to achieve higher recognition accuracy by deepening our architecture. However, we prefer to keep a compact model for resource-limited systems such as Nvidia's Jetson TX2 for instance, as the memory requirement and computational cost are critical in resources-limited systems for which complicated architectures are not suitable.

Table 6 Comparison of recognition accuracy on UEC-FOOD100 with other works

Method	Top-1	Top-5
CNN_5	60.90%	86.15%
CNN_7	66.41%	88.40%
CNN_9	75.80%	92.95%
DCNN [15]	57.87%	/
RootHoG FV [15]	50.14%	/
Color FV [15]	53.04%	/
FV + DCNN [15]	72.26%	92.00%
SURF-BoF + ColorHistogram [16]	42.0%	68.2%
HOG Patch-FV + Color Patch-FV [16]	51.9%	79.2%
AlexNet [7]	75.62%	92.43%
VGG [7]	81.31%	96.72%
Yanai et al. [28]	78.77%	95.15%

5 Conclusion

This paper presents a small and efficient convolutional neural network architecture for Chinese food recognition. Experiment results show that the proposed CNN_5 architecture achieved top-1 accuracy of 97.12% and top-5 accuracy of 99.86% on the newly created Chinese food dataset. Our architecture's model size is approximately 10 and 50% of the AlexNet and GoogLeNet, respectively. This small model size is critical for performing food recognition on a resource-limited system such as a smartphone. We also verify that the network depth impacts recognition accuracy. Increasing network depth does improve accuracy slightly but for the price of network complexity and slower classification speed. We compare our CNN_5 architecture with four BoF methods which require experts to design or select useful descriptors to characterize the specifics of the food image. We confirm that CNNs learn efficient filters implicitly and automatically with their intrinsic unified optimization and thus achieve superior performance for Chinese food recognition. We also attempted to correlate the proposed CNN architecture with traditional BoF methods. We conclude that CNNs actually perform every step of BoF in convolutional layers and fully-connected layers and provide a unified optimization to the entire network.

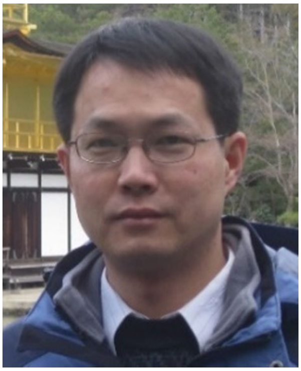
Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Aharon M, Elad M, Bruckstein A (2006) K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311–4322
2. Andrew P (2013) AMA recognizes obesity as a disease. The New York Times, p.10
3. Anthimopoulos MM, Gianola L, Scarnato L et al (2014) A food recognition system for diabetic patients based on an optimized bag-of-features model. IEEE Journal of Biomedical and Health Informatics 18(4): 1261–1271
4. Ashkan A, Forouzanfar Mohammad H, Reitsma Marissa B et al (2017) Health effects of overweight and obesity in 195 countries over 25 years. N Engl J Med 377(1):13–27
5. Bleich S, Cutler D, Murray C et al (2008) Why is the developed world obese? Annu Rev Public Health 29(1):273–295
6. Chen M, Dhingra K, Wu W, et al (2009) PFID: Pittsburgh fast-food image dataset. IEEE International Conference on Image Processing. pp. 289–292
7. Chen J, Ngo CW (2016) Deep-based ingredient recognition for cooking recipe retrieval. ACM Multimedia Conference, ACM, 32–41
8. Chen M, Yang Y, Ho C, et al (2012) Automatic Chinese food identification and quantity estimation. In: SIGGRAPH Asia 2012 Technical Briefs. ACM, pp 29
9. Ciocc G, Napoletano P, Schettini R (2017) Food Recognition: A new dataset, experiments, and results. IEEE Journal of Biomedical and Health Informatics 21(3):588–598
10. Dong C, Loy CC, He K et al (2016) Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis & Machine Intelligence 38(2):295
11. Farinella GM, Moltisanti M, Battiato S (2014) Classifying food images represented as bag of textons. In: Image Processing (ICIP), 2014 IEEE International Conference. IEEE, pp.5212–5216
12. Giovany S, Putra A, Hariawan AS et al (2017) Machine learning and sift approach for Indonesian food image recognition. Procedia Computer Science 116:612–620
13. Haslam DW, James WP (2005) Obesity. Lancet (Review) 366(9492):1197–1209. https://doi.org/10.1016/S0140-6736(05)67483-1
14. Kawano Y, Yanai K (2013) Real-Time Mobile Food Recognition System. In Computer Vision and Pattern Recognition Workshops. IEEE:1–7

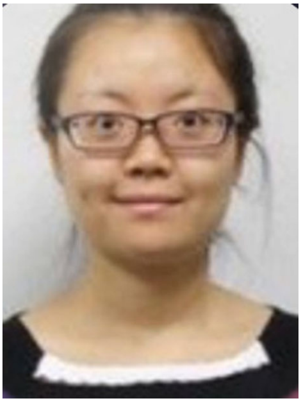
15. Kawano Y, Yanai K (2014) Food image recognition with deep convolutional features. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. ACM, pp 589–593
16. Kawano Y, Yanai K (2015) Foodcam: a real-time food recognition system on a smartphone. Multimedia Tools & Applications 74(14):5263–5287
17. Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint arXiv:1312.4400
18. Luan S, Zhang B, Chen C et al (2017) Gabor Convolutional Networks. IEEE Trans Image Process 27(9): 4357–4366
19. Luppino FS, de Wit LM, Bouvy PF et al (2010) Overweight, obesity, and depression: a systematic review and meta-analysis of longitudinal studies. Arch Gen Psychiatry 67(3):220–229
20. Martin CK, Kaya S, Gunturk BK (2009) Quantification of food intake using food image analysis. In: Engineering in Medicine and Biology Society. EMBC 2009. Annual International Conference of the IEEE. pp 6869–6872
21. Matsuda Y, Hoashi H, Yanai K (2012) Recognition of multiple-food images by detecting candidate regions. In Multimedia and Expo, pp 25–30
22. Matthew W (2013) The facts about obesity. H&HN. American Hospital Association. Retrieved June 24
23. Papyan V, Romano Y, Elad M (2016) Convolutional neural networks analyzed via convolutional sparse coding. arXiv preprint arXiv:1607.08194
24. Shroff G, Smailagic A, Siewiorek DP (2008) Wearable context-aware food recognition for calorie monitoring. In: Proc. 12th IEEE Int. Symp. Wearable Comput, pp 119–120
25. USDA (2008) Food and Nutrient Database for Dietary Studies, 3.0. Agricultural Research Service, Food Surveys Research Group, Beltsville
26. Thai Van Phat, Dang Xuan Tien, Quang Pham, et al. (2017) Vietnamese food recognition using convolutional neural networks. International Conference on Knowledge and Systems Engineering, pp 124–129
27. Wang L, Zhang B, Han J et al (2016) Robust object representation by boosting-like deep learning architecture. Signal Process Image Commun 47:490–499
28. Yanai K, Kawano Y (2015) In: Proceedings of 2015 IEEE Int. Conf. on Multimedia and Expo Workshops, Trino, pp. 1–6
29. Yanai K, Tanno R, Okamoto K, et al (2016) Efficient mobile implementation of a CNN-based object recognition system. ACM on Multimedia Conference. ACM, pp 362–366
30. Yang S, Chen M, Pomerleau D, et al (2010) Food recognition using statistics of pairwise local features. In: Computer Vision & Pattern Recognition, pp 2249–2256
31. Yang H, Zhang D, Lee D-J, et al (2016) A sparse representation based classification algorithm for Chinese food recognition. In: International Symposium on Visual Computing Springer, pp 3–10
32. Zhang W, Zhao D, Gong W, et al (2016) Food image recognition with convolutional neural networks. Ubiquitous Intelligence and Computing and 2015 IEEE, Intl Conf on Autonomic and Trusted Computing and 2015 IEEE, Intl Conf on Scalable Computing and Communications and ITS Associated Workshops. IEEE, pp 690–693
33. Zhao J, Han J, Shao L Unconstrained Face Recognition Using A Set-to-Set Distance Measure on Deep Learned Features. IEEE Transactions on Circuits and Systems for Video Technology. https://doi.org/10.1109/TCSVT.2017.2710120
34. Zhu F, Marc B, Insoo W et al (2010) The use of mobile devices in aiding dietary assessment and evaluation. IEEE Journal of Selected Topics in Signal Processing 4(4):756–766

Jianing Teng received his B.S.E.E. degree from Sun Yat-sen University, China, in 2017. He is currently a postgraduate student in the School of Electronics and Information Technology at Sun Yat-sen University. His research interests include machine learning and image processing.



Dong Zhang received his B.S.E.E. and M. S. degrees from Nanjing University, China, in 1999 and 2003, respectively, and Ph.D. degree from Sun Yat-sen University, China, in 2009. He is currently an associate professor in the school of Electronics and Information Technology, Sun Yat-sen University. His research interests include image processing, pattern recognition and information hiding.

Dah-Jye Lee received his B.S. degree from National Taiwan University of Science and Technology in 1984, M.S. and Ph.D. degrees in electrical engineering from Texas Tech University in 1987 and 1990, respectively. He also received his MBA degree from Shenandoah University, Winchester, Virginia in 1999. He worked in the machine vision industry for eleven years prior to joining BYU in 2001. He is currently a Professor in the Department of Electrical and Computer Engineering at Brigham Young University. His research work focuses on object recognition, hardware implementation of real-time vision algorithms and machine vision applications.



Yao Chou received the B.S. degree from Sun Yat-sen University, China, in 2013. She is now pursuing her Ph.D. degree in the Department of Electrical and Computer Engineering at Brigham Young University (Provo, USA). She is interested in machine learning and computer vision.