DeepFood: Automatic Multi-Class Classification of Food Ingredients Using Deep Learning

Lili Pan[1], Samira Pouyanfar[2], Hao Chen[3], Jiaohua Qin[1], Shu-Ching Chen[2]
[1]College of Computer Science and Information Technology
Central South University of Forestry and Technology, Hunan, China
[2]School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
[3]College of Computer Science and Electronic Engineering
Hunan University, Hunan, China

lilypan@csuft.edu.cn, spouy001@cs.fiu.edu, chenhao@hnu.edu.cn, qinjiaohua@163.com, chens@cs.fiu.edu

***Abstract*—Deep learning has brought a series of breakthroughs in image processing. Specifically, there are significant improvements in the application of food image classification using deep learning techniques. However, very little work has been studied for the classification of food ingredients. Therefore, this paper proposes a new framework, called DeepFood which not only extracts rich and effective features from a dataset of food ingredient images using deep learning but also improves the average accuracy of multi-class classification by applying advanced machine learning techniques. First, a set of transfer learning algorithms based on Convolutional Neural Networks (CNNs) are leveraged for deep feature extraction. Then, a multi-class classification algorithm is exploited based on the performance of the classifiers on each deep feature set. The DeepFood framework is evaluated on a multi-class dataset that includes 41 classes of food ingredients and 100 images for each class. Experimental results illustrate the effectiveness of the DeepFood framework for multi-class classification of food ingredients. This model that integrates ResNet deep feature sets, Information Gain (IG) feature selection, and the SMO classifier has shown its supremacy for food-ingredients recognition compared to several existing work in this area.**

Keywords*—*Image classification, Food recognition, Multi-class classification, Deep learning, Feature extraction, Convolutional neural network

I. Introduction

Food has always been essential in human life and attracted people's attention more than before. Currently, food supplies depend on human visual inspection to evaluate the qualified food ingredients and label them properly. This process is extremely laborious, tedious, and costly [1]. Therefore, a food detection system that can automatically classify qualified food ingredients is imperative.

Nowadays, image processing and recognition achieve rapid advancements in different applications [2, 3, 4, 5, 6, 7], such as surveillance systems, medical imaging, remote sensing, to name a few. Various research work has shown that machine learning and data mining techniques can be utilized to classify food images automatically [1, 8, 9]. However, existing food detection approaches mainly focus on diet [9, 10, 11], and available datasets are usually composed of food meals pictures as shown in Fig. 1(a). Currently, there are very few food ingredients datasets available (as shown in Fig. 1(b)), and thus, there is limited work on multi-class classification of food ingredient images in the literature [1]. To effectively classify different food ingredients, in this paper, we propose an automatic multi-class classification framework using Convolutional Neural Networks (CNNs).

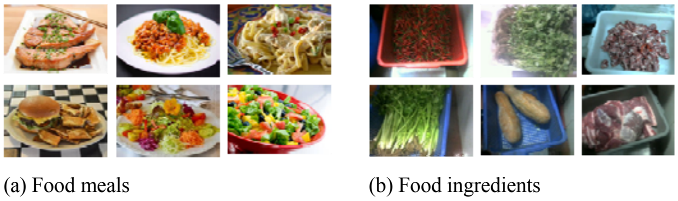
(a) Food meals (b) Food ingredients

Fig. 1. Example of two different food image datasets

In 2006, Hinton et al. proposed that low-dimensional codes can be acquired from high-dimensional data by training a multilayer neural network with a small central layer to generate high-dimensional input vectors [12]. Since then, deep learning has been utilized in many applications, and receiving continuous attention in both academia and industry [13, 14, 15]. Because of the impressive performance of deep learning in image recognition, in this paper, it is applied to the multi-class classification of food ingredients. The recent research studies in deep learning showed that neural networks have been expanded deeper and wider [14, 16]. For example, on the ILSVRC 2015 classification task, the depth of residual nets reached over 150 layers, eight times deeper than VGG nets [17]. With the extension of network's depth and width, it is feasible to extract more thriving and high-level features compared to the shallow networks [18].

One of the main challenges is that it requires a large-scale image data to train a deep learning model from scratch, such as the ImageNet dataset which includes millions of labeled images. Till now, the problem has been addressed by two important methods. The first approach is fine-tuning that takes an already learned model, adapts the architecture, and resumes training from the model weights already trained [8]. Another solution is using a pre-trained deep learning with a large-scale dataset as a fixed feature extractor for a small-scale data. The question is, for the multi-class food ingredients dataset, which

Identify applicable funding agency here. If none, delete this text box.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

technique will generate a better performance. In [1], a fine-tuned CNN model for food ingredients is described, and its best accuracy is only about 60%. Another issue is whether the extracted features from a pre-trained deep learning on a different dataset (e.g., ImageNet) improve the performance of multi-class classification of food ingredients. Several studies have confirmed the effectiveness of deep learning features in various applications [8, 19, 20, 21].

To address the aforementioned problems, this paper presents a new framework, called DeepFood, for multi-class classification of food ingredients using deep learning. This approach extracts rich and effective features using CNNs and classifies food ingredient images. The comprehensive experimental results show that the proposed food classification framework significantly improves the recognition accuracy.

The remainder of the paper is organized as follows. First, an overview of the state-of-the-art research in food detection and CNNs is provided in section 2. Section 3 presents the details of the proposed multi-class food classification. Section 4 analyzes the experimental results on three different CNN models. Finally, the paper is summarized in section 5.

II. Related Work

A. Food detection

Regarding the food recognition, there are several advancements in the literature, especially in the last few years. In [22] and [23], both global and local features of food images are extracted for food classification. The former uses the k-nearest neighbors and vocabulary trees, while the latter integrates the local appearance and structural information of food objects for the food image classification task. Farinella et al. [24] used visual word distributions (Bag of Textons) to represent food images and a Supported Vector Machine (SVM) to classify them. Bettadapura et al. [25] utilized the context of where the picture was taken as the features to classify the food being consumed. That dataset is composed of real-world food images that are labeled based on the foods from five different countries (American, Indian, Italian, Mexican, and Thai).

Joutou et al. [9] used a Japanese food dataset for food recognition. They introduced a multiple kernel learning to integrate several kinds of image features such as color, texture, and Scale Invariant Feature Transform (SIFT), and achieved 61.34% classification rate based on the 50 kinds of hand-selection food images from the Web. Hoashi et al. [10] recognized 85 food items, achieving 62.5% accuracy for the recognition of Japanese food images. They used multiple kernel learning for feature fusion as their learning method. The Pittsburgh Fast-food Image Dataset (PFID) [26] is the first publicly available food dataset that contains 101 classes and has three instances per class. Chen et al. [27] described the food classification on a dataset with 50 Chinese food categories. Zhu et al. [28] proposed a food recognition method using a small dataset, which was intended to be used in a smartphone-based food-logging system as part of their Technology Assisted Dietary Assessment project. Bossard et al. [29] introduced a novel method to simultaneously mine discriminative food image superpixels (components) using Random Forest and evaluated the method on Food-101 (downloaded from foodspotting.com).

Recently, deep learning has become very effective for large-scale object recognition and has been leveraged widely in the food image recognition applications. A fast auto-clean CNN model for the classification of food ingredients is proposed in [1]. The framework describes a fine-tuning technique with CNNs for online prediction of food ingredients. Yanai et al. [8] used 1000 food-related categories for the pre-training of deep CNNs (DCNNs) and achieved a better performance on food classification. First, they selected the 1000 food-related categories from ImageNet with 2000 categories and combined them with the ILSVRC 1000 ImageNet Categories. Second, they pre-trained two DCNNs on the two datasets respectively using Caffe, as well as fine-tuning the two pre-trained DCNNs. The experimental results show that the fine-tuned DCNN with 2000 categories reaches to the best result. It is worth noting that the fined-tuned DCNN was trained based on the former pre-trained DCNN. Kagaya et al. [11] applied a CNN for food detection and recognition and constructed their dataset of ten food items in a publicly available food-logging system. In [30], the authors leveraged CNN to classify food/non-food images based on three different datasets. Christodoulidis et al. [31] described food recognition with a patch-wise manner and a voting technique using a 6-layer CNN. Ciocca et al. [32] introduced a food recognition algorithm with a food dataset called UNIMIB2016 that contains 73 food classes and a total of 3616 food instances. They utilized multi features to classify food, and their experiment results showed that the CNNs-based features obtained a better performance.

B. Convolutional neural networks

Recently, deep learning has become a popular topic in the literature which is originated from traditional artificial neural networks. Currently, Convolutional Neural Networks (CNNs) are commonly utilized in computer vision. Specifically, deep convolutional neural networks have led to a series of breakthroughs in image classification [13, 14, 33].

AlexNet [13] is the first architecture leveraging deep convolutional layers for image classification. The framework developed by Alex Krizhevsky et al. outstandingly performed better than the other advanced methods in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. It includes several convolutional and pooling layers stacked on top of each other rather than a single convolutional layer followed by a pooling layer.

GoogLeNet [16] is the ILSVRC 2014 winner that is a deeper and wider CNN developed by Google. Its main module is also known as Inception that dramatically reduces the number of parameters in the network. In addition, GoogLeNet uses average pooling instead of fully connected layers at the top of the CNN, which eliminates a large number of network parameters. The accuracy using GoogleLeNet is 43.9% in the 2014 ILSVRC, which is much higher than top results (22.6%) in the 2013 ILSVRC.

To date, a new deep architecture called "Deep Residual Learning" [14] becomes the milestone of CNNs. Residual

Network (ResNet) developed by Kaiming He et al. from Microsoft was the winner of ILSVRC 2015 and COCO 2015 competitions on ImageNet detection and localizations, as well as COCO segmentation and detection. It is extremely deeper than the previous frameworks. The residual architecture is proved to easily design a substantially deeper CNN than before because it provides a residual learning that decreases the degradation.

All the aforementioned CNNs, as well as other well-known deep learning architectures, have achieved a series of breakthroughs in image processing. It is worth noting that very large-scale datasets are needed to train a deep CNN model. However, collecting and annotating such datasets are difficult and cumbersome, and training deep learning using small datasets is necessary but very challenging. Therefore, this paper presents a multi-class classification framework for small and medium scale food ingredients datasets using transfer learning.

III. The Proposed DeepFood Framework

In this study, we propose an automatic multi-class classification of food ingredients using deep learning feature extraction, feature selection, and SMO classifier. The framework is shown in Fig. 2, which includes three main modules: (1) two-level feature extraction using CNNs, (2) feature selection, and (3) classification.

A. *Feature Extraction using CNNs*

In many real-world problems, it is common to classify a small-scale dataset where training a deep learning from scratch is not possible. Instead, transfer learning is a popular technique for classifying medium and small-size datasets. In the deep learning area, transfer learning is the process of applying a pre-trained deep model such as CNN which is originally trained on a large-scale dataset (e.g., ImageNet dataset) and used as a fixed feature extractor for a small-scale data. The raw images are given as the input of a pre-trained CNN and then activation vectors are derived from its intermediate layers. CNN vectors are propagated into the upper layers and the extracted vectors can be regarded as the image features. The CNN features are commonly extracted from the last output layers of the pre-trained CNN.

CNN is a multilayer artificial neural network which incorporates both unsupervised feature extraction and classification. A CNN takes raw images as the input and generates the final classification scores. Specifically, the neurons in a layer are arranged in three dimensions: width, height, and depth, which are only connected to a small region of its previous layer. The last layer of CNN reduces the full image into a single vector of class scores. In general, a CNN is a series of layers and the most popular ones include Convolutional (Conv), Pooling, Rectified Linear Units (ReLU), and Fully-Connected (FC) layers. The Convovlutional layer is characterized by the sparse connectivity and sharing weights. It computes the output of neurons connected to the local regions in the inputs from the previous layer and shares the weights within each feature map corresponding to the kernels in the same layer.

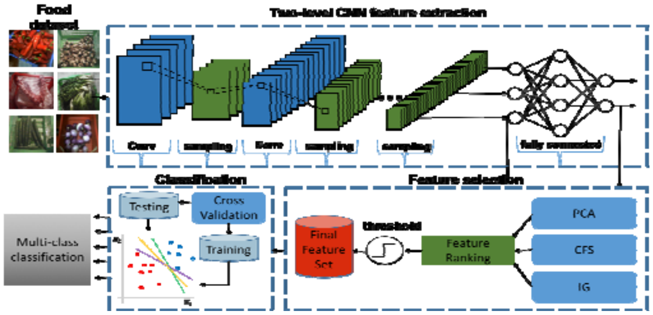

Fig. 2. The proposed multi-class classification framework

The output of convolution passes through an activation function that produces nonlinearities in an element-wise fashion. A pooling layer makes a downsampling along the spatial dimension (width, height) of the previous layer volume. For instance, a [32*32*10] input will be subsampled to [16*16*10] in a pooling layer with a filter size 2 and stride 2. Max or mean pooling replaces the input values with the maximum or the mean value, respectively. A ReLU layer applies an elementwise activation function, i.e. max(0, x). The FC layer computes the final class scores in which all neurons are connected to the ones in the previous volume and the number of the output volume will be equal to the number of classes. This part of the CNN performs the supervised classification. When a CNN is built with a cascade of layers above in a proper way, the CNN transforms the original pixel values of a raw image to the final class scores. Similar to the traditional Multilayer Perceptron, in the Conv/FC layers, the parameters are trained with gradient descent so that the neuron weights are updated in each iteration to reduce the final classification errors regarding the training set. A gradient descent method is applied using the back propagation technique for training a CNN. In Fig. 2, the two-level CNN feature extraction module utilized in this paper is shown. The last two layers in each CNN model, either pooling or FC, are used as two different sets of features which are further used for the feature selection and classification. In this paper, three popular and state-of-the-art CNNs are applied to extract visual feature vectors as described below.

AlexNet [13]: was developed in 2011 and considered as the first deep network disseminated deep learning in computer vision area. It downsamples the raw RGB images into a fixed resolution of 256*256 pixels. The network contains eight layers, where the first five are convolutional layers, including ReLU and pooling layers, and the remaining three are fully connected layers. The numbers of features in the seventh and eighth layers are 4096 and 1000, respectively.

CafffeNet [34]: a reproduction of AlexNet with some improvements. It is developed and trained by the Berkeley Vision group. CaffeNet is trained with the relighting data-augmentation, and the pooling layer in the architecture is done before the normalization. The numbers of features in the last two layers of the net are the same as those in Alexnet.

ResNet [14]: was developed by Microsoft research in 2015. It includes special residual connections and heavily uses batch normalization. The last two layers in RestNet-50 are a pooling layer and an FC layer, where the output numbers of features are 2048 and 1000, respectively. The network shows better performance than the smaller (34-layers) ones. So far, many visual image applications benefit from ResNet.

In Fig. 2, iteratively using a function across the local-region of the whole input in a convolutional layer produces a lot of feature maps. That means the input data is convoluted with linear filters coming after nonlinear activation functions. As shown in Equation 1, the k^{th} feature vector at the k^{th} layer is defined as x^k_{ij}, where k is the given layer, i and j are the sizes of the input data, x^{k-1}_{ij} is the input data from the former layer, θ is an activation function like sigmoid, and the filters of the k^{th} layer are denoted as ω^{k}_{ij} (weights) and β^k_j (bias). A pooling layer is a nonlinear down-sampling going behind each convolutional layer. It decreases the number of feature elements by presenting sparseness, as well as provides extra sturdiness to the CNN. This layer gets a small piece from the former convolutional layer and generates an individual output as shown in Equation 2, where δ^k_{ij} is a multiplicative bias and down(.) is a subsampling function such as max.

$$x_{ij}^k = \theta((\omega_{ij}^k * x_{ij}^{k-1}) + \beta_j^k \quad (1)$$

$$x_{ij}^k = \theta(\delta_{ij}^k down(x_{ij}^{k-1}) + \beta_j^k) \quad (2)$$

In the DeepFood framework, the feature extraction takes the advantage of the transfer learning technique using CNNs. First, the dataset is separated into training T and testing T'. T is defined as T={(t_1, c_1), (t_2, c_2), …, (t_N, c_N}, where t_i is the i^{th} training instance, N is the total number of training instances, and $c_i \in C$ is the i^{th} food image's class label. Classes C={lab_1, lab_2, …, lab_{Nc}} and Nc is the total number of food classes. Second, the pre-trained model (e.g., ResNet) and its last two layers h={1, 2} are used for unsupervised feature extraction. In addition, the extracted feature sets are stored in $F_h = \{f_h^1, f_h^2, \ldots, f_h^{Nh}\}$ where f_h^i is the i^{th} feature vector from the layer h, and Nh is the number of extracted features from this layer.

In sum, high-level and efficient features are extracted by applying the proposed two-level CNN feature extraction module with transfer learning (as shown in Fig. 2).

B. Feature selection

In the past fifteen years, studies on generic object recognition have proposed various feature representations [35, 36, 37]. Especially, Histogram of Oriented Gradients (HOG) [35], Bag-of-Features [38], and Scale Invariant Feature Transform (SIFT) [37] are powerful features that have been widely applied in computer vision. However, recently, deep learning has been rapidly developed for image recognition and has exceedingly raised the performance levels [38]. CNNs try to represent visual data with the high-level abstractions by using architectures composed of multiple non-linear transformations. In [39], the effectiveness of CNN features is confirmed with the experiments on Caltech-101 and SUN-397 datasets. CNN features have been proved to be valid for image classification in many applications [8, 13, 14, 21]. Therefore, this paper extracts rich and high-level features with transfer learning based on the pre-trained CNN models.

One important fact remained is the high-dimension of deep features compared to the hand-crafted ones. Therefore, how to optimize the deep features in an efficient manner is a critical challenge. For this purpose, we leverage several attribute evaluators and search techniques in the feature selection module of the DeepFood framework (shown in Fig. 2). Since the number of deep features using deep learning is numerous, it will cost too much time to classify food ingredients. In addition, the deep features are naturally sparse and may include lots of irrelevant information. Therefore, feature selection is a proper way that reduces the dimensionality of the feature space and gets rid of redundant, slightly useful or noisy features. Our proposed framework utilizes three different feature evaluators including Principal Component Analysis (PCA), Correlation Feature Selection (CFS), and Information Gain (IG). PCA [40] is a common and useful statistical technique that decreases the image representation dimensionality and aims at minimizing the losses in the variance of raw data. As a domain independent and unsupervised technique, PCA is applied to a diversity of data [41]. CFS [42], on the other hand, is a simple and fast feature selector that removes redundant, irrelevant and noisy data in less computational time than the state-of-the-art feature selection algorithms (such as PCA). It selects a subset of raw data based on the following hypothesis: good feature subsets are uncorrelated to each other, yet highly correlated with the classification. IG is also a very effective and popular approach that evaluates features respectively for category prediction. Specifically, it measures the entropy or uncertainty in a feature set and selects the one with the highest information. Yang and Pedersen [43] showed that IG is more effective than other feature selection algorithms including mutual information, term strength, etc.

Algorithm 1 illustrates the proposed feature selection procedure which creates a series of feature subsets using PCA, CFS, and IG. The input includes a matrix containing all training instances T and feature sets F_h, as well as all attribute evaluators A including PCA, CFS and IG, and the Threshold

Algorithm 1. Feature Selection Based on Deep Features

```
Input: Feature sets $F\{f_h$, h=1, 2}, Training T{($t_i$, $c_i$), i=1, 2, …, $N$}, Attribute
       Evaluators $A\{(a_j)$, j=1, 2, 3}, Threshold matrix $V\{V_j\}$, $V_j$={$v_s$, $s$=1,
       2, …, $N_v$}, p=0, q=0
Output: Feature sets: $FT$ and $N_{ft}$
for all $f_h$ do
    for all $a_j$ do
        $fs_p \leftarrow a_j (T, f_h)$;
        $fr_p \leftarrow$ Feature Ranking($fs_p$);
        p = p+1;
        for all $v_s$ in $V_j$
            $ft_q \leftarrow$ Threshold ($fr_p$, $v_s$);
            q = q+1;
        end for
    end for
end for
$N_{ft}$ = q;
return $ft_q$ and $N_{ft}$
```

Matrix V. In this algorithm, A = {(a_j), j=1, 2, 3}, where a_j is the j^{th} attribute evaluator, and V is defined as V={V_j} in which each V_j is a one-dimensional vector composed of N_v thresholds corresponding to a_j. The algorithm's output includes the selected feature subsets FT, and its size N_{ft}. First, different attribute evaluators are used to evaluate all the deep feature sets (as shown in the first and second loops in Algorithm 1). Each raw feature set is evaluated by each attribute evaluator a_j in line 3. Then, feature ranking is applied based on feature evaluation scores, and a newly ranked feature set fr_p is generated in line 5. Last, an efficient feature subset, ft_q is created using the Threshold function in line 8. The Threshold is used to remove features from fr_p whose evaluation score is less than v_s.

C. Classification

After feature subsets are produced with the attribute selectors, how to optimally train a model and classify the data is a key problem. For this purpose, the DeepFood framework leverages Sequential Minimal Optimization (SMO) with cross validation for training models (shown in Fig. 2). SMO is an improved algorithm for training Support Vector Machines (SVM) on classification tasks. SVM is an optimized model that simultaneously minimizes both the prediction error and model complexity. However, SVM has been hindered due to the complex and expensive Quadratic Programming (QP) solvers for years. Afterwards, SMO [44] is proposed as an efficient solution to iteratively solve QP by breaking it into smallest possible sub-problems.

In this paper, the classification module contains two main steps: training and testing. First, the dataset is split into training T and testing T' using three-fold cross validation. T is already defined in Section III (A). T'= {(t_1, c_1), (t_2, c_2), …, (t_{Nt}, c_{Nt}}, where t_i is the i^{th} testing instance, and Nt is the total number of testing instances.

In the training phase, multiple SMOs are trained to classify multi-class food ingredients using the training instances T and different feature sets FT described in Section III (B). In testing, all the trained SMO models are utilized to predict the label of each testing instance as shown in Algorithm 2. The inputs of the testing algorithm include testing data T' and the corresponding feature representation FT, as well as all the trained models $SMOs$. Its output is a predicted label set PL, and an accuracy set Acc. In Algorithm 2, the accuracy is calculated for each SMO model. The j^{th} test instance is predicted as PL_{ij} using SMO_i in line 3. The test instances whose labels are correctly predicted using SMO_i are summed as NC_i in line 5. Then, the accuracy of SMO_i is calculated as Acc_i in line 6. Last, all the predicted labels and the accuracy values are returned in line 8.

Alogrithm 2. Evaluating the multi-class classification framework

```
Input: Testing instances $T'$ {($t_j$, $c_j$), j=1, 2, …, $Nt$}, and Feature sets $FT${$ft_q$,
       q=1, 2, …, $N_{ft}$}, Trained models $SMOs$
Output: Predicted labels $PL_{ij}$ and average accuracy $Acc_i$
1:  for all $SMO_i \in SMOs$ do
2:      for all $(t_j, c_j) \in T'$
3:          $PL_{ij} \leftarrow SM\ O_i(T')$;
4:      end for
5:      $NC_i \leftarrow \sum$Correct label instances
6:      $Acc_i = \frac{NC_i}{N_t}$
7:  end for
8:  return $PL_{ij}$, $Acc_i$
```

IV. Experimental Analysis

A. Food Dataset

In this research, the dataset of food ingredients is provided by a large food supply chain platform in China[1] called Mealcome (MLC dataset) [1]. The initial food ingredient pictures were taken in the field which contains a mixture of various backgrounds and food ingredients. Some of the original images are easy to be distinguished by human vision, while others are difficult to be classified into different food classes because of blurriness, noise, illumination, overexposure, or some other reasons. Therefore, we removed noisy images and picked clearly distinguished images and classified them into different food classes. Finally, the MLC-41 dataset was constructed including 41 food labels and 100 images for each label, and each image resolution is changed to 640*480 pixels to have a more accurate feature extraction and food recognition. Although this dataset is balanced, the number of classes is very high in comparison with the number of images in each class. This makes the training process more challenging. Some samples of our dataset are shown in Fig. 3.

B. Experimental Setup

In a multi-class classification, how to evaluate the framework is important. In general, metrics such as Precision, Recall, and F1 measure are suitable for binary classification, especially imbalanced datasets. Since our dataset is balanced and the task is multi-class classification, we utilize the accuracy metric to evaluate our proposed DeepFood framework.

Caffe [34] is a popular deep learning tool which includes modern and advanced deep learning techniques. In addition, it contains a series of pre-trained reference models, such as

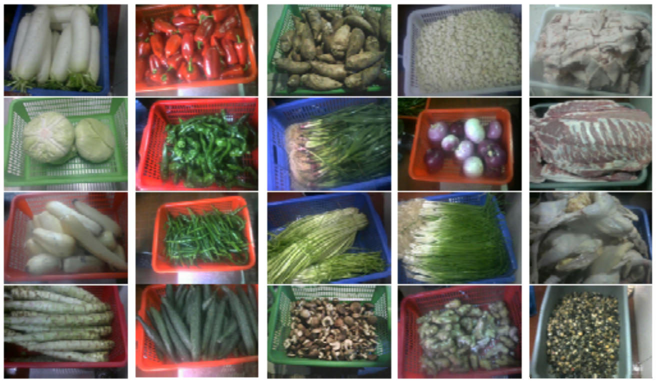

Fig. 3. Image Samples of MLC-41

[1] www.mealcomm.com

AlexNet, CaffeNet, GoogleNet, and ResNet. To evaluate our proposed framework, we compared our proposed feature extractor with AlexNet and CaffeNet models. More specifically, in Section II-B and III-A, these pre-trained CNN models are described. In this work, deep features are extracted from the last two layers of each model. For instance, layers “fc7” and “fc8” of CaffeNet and AlexNet, and “pool5” and “fc1000” of ResNet are the output layers utilized in the feature extraction module. The layer “fc7” generates a 4096-dimension feature vector, “fc8” and “fc1000” generates a 1000-dimension feature vector, and “pool5” produces a 2048-dimension vector from the three selected models. Additionally, different benchmark classifiers such as Random Forest, Bagging, and BayesNet are compared with SMO which is used in the proposed DeepFood framework.

C. Experimental Results

The framework for multi-class classification of food ingredients is evaluated on the MLC-41 dataset. This experiment not only applies several feature evaluators including PCA, CFS, and IG, but also uses ranking metrics as a filter to select the best deep feature subsets. All the features extracted from the deep learning models are selected to build a series of new feature subsets using the attribute evaluators. Each classifier is tuned to reach to its highest performance on the MLC-41 dataset and evaluated using the 3-fold cross validation.

The average accuracy of various deep feature sets integrated with different deep learning models and feature selection methods are shown in Table 1. According to this Table, the deep features extracted using CNNs improve the performance for the multi-class classification of food ingredients. Most importantly, ResNet beats another two CNN models (AlexNet and CaffeNet) when they are applied to extract features from our dataset. The average accuracy of the multi-class classification of food ingredients using the DeepFood framework with RestNet and IG feature selection attains the highest value, where the average accuracy reaches to 87.78%. The average performance of AlexNet is very close to CaffeNet. The original deep features maintain a better average accuracy than other selected feature sets with PCA, CFS and IG using AlexNet and CaffeNet. However, using the presented framework with ResNet, IG gets the best feature set and attains the highest average accuracy. Thus, the DeepFood framework enhances the overall performance for multi-class classification of food ingredients.

Table 2 shows the accuracy comparison between different deep-learning layers and feature evaluators. In the DeepFood framework, we extract features from the last two layers of deep learning. As can be seen from Table 2, both layers in ResNet achieve much higher accuracy (almost 10%) than other deep learning benchmarks. A similar behavior is shown in Tables 1 and 3. Specifically, the “fc7” and “pool5” layers have better performance than the last layer in each network (e.g., “fc8” or “fc1000” layer). Therefore, based on the experiment, it can be concluded that the second last layer is better for feature extraction than the last layer of a CNN. The main reason is that the last layer generates the final score (probability) for each class in the pre-trained models. Since the classes in the original dataset (e.g., ImageNet) may be very different with the current dataset (MLC-41), the final scores are not as discriminative and general as the outputs of the former layers. In addition, the last layer generates the same number of features as the classes (1000 for ImageNet) which is less than the second to last layer (4096 or 2048 features). Table 3 describes the accuracy comparison between various CNN models and classifiers. Our experiments show that the DeepFood framework integrating ResNet deep feature sets, Information Gain (IG) feature selection, and the SMO classifier is superior to other techniques for food-ingredients recognition. As can be inferred from Table 3, the best average accuracy results are 80.42%, 80.76%, and 87.78% from SMO using different deep feature sets. The next best results have the average accuracy of 71.32%, 71.10% and 82.07% from the BayesNet classifier. Based on the experimental results, one can conclude the effectiveness of the proposed framework which significantly improves the accuracy of the food-ingredients classification. The best model is acquired with an integration of the ResNet feature sets with the IG feature

Table 1. Accuracy comparison between different deep learning and feature selection methods

Deep learning model	**Feature selection**							
	original		***PCA***		***CFS***		***IG***	
AlexNet	Fold1	78.419	Fold1	76.664	Fold1	76.664	Fold1	78.346
	Fold2	80.248	Fold2	77.980	Fold2	77.029	Fold2	80.395
	Fold3	82.576	Fold3	78.038	Fold3	77.525	Fold3	82.503
	Avg	80.415	**Avg**	77.561	**Avg**	77.073	**Avg**	80.415
CaffeNet	Fold1	79.590	Fold1	77.761	Fold1	76.152	Fold1	79.444
	Fold2	81.419	Fold2	80.321	Fold2	78.419	Fold2	81.858
	Fold3	81.259	Fold3	77.672	Fold3	75.988	Fold3	80.600
	Avg	80.756	**Avg**	78.585	**Avg**	76.853	**Avg**	80.634
ResNet	Fold1	86.466	Fold1	84.784	Fold1	86.174	Fold1	87.052
	Fold2	88.588	Fold2	86.466	Fold2	88.880	Fold2	88.222
	Fold3	87.847	Fold3	86.896	Fold3	88.213	Fold3	88.067
	Avg	87.634	**Avg**	86.048	**Avg**	87.756	**Avg**	**87.780**

Table 2. Accuracy comparison between different deep-learning layers and feature selection methods

Deep learning model	**Feature selection**							
	original		***PCA***		***CFS***		***IG***	
AlexNet	**fc7**	80.415	**fc7**	77.561	**fc7**	76.317	**fc7**	80.415
	fc8	74.878	**fc8**	75.000	**fc8**	75.097	**fc8**	74.976
CaffeNet	**fc7**	80.756	**fc7**	78.585	**fc7**	76.390	**fc7**	80.634
	fc8	75.268	**fc8**	75.804	**fc8**	74.609	**fc8**	75.122
ResNet	**pool5**	87.634	**pool5**	86.049	**pool5**	87.756	**pool5**	87.780
	fc1000	84.780	**fc1000**	84.975	**fc1000**	84.585	**fc1000**	84.878

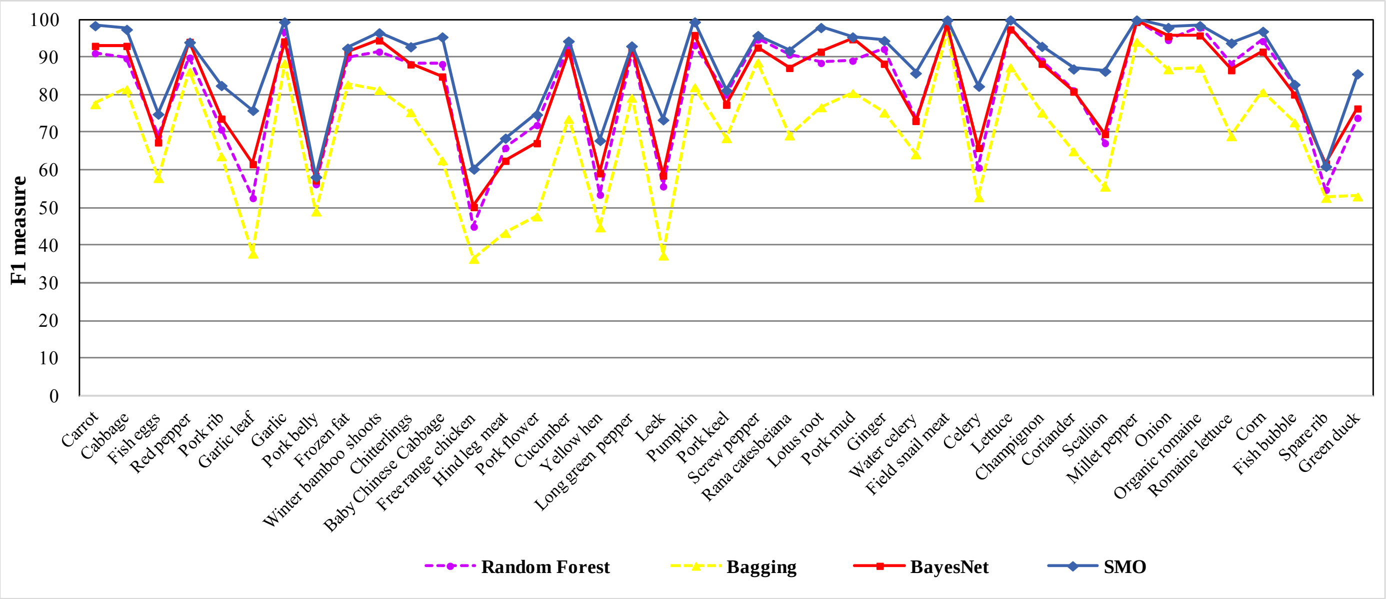

Fig. 4. Performance evaluation for different classifiers on various food ingredients classes

Table 3. Accuracy comparison between different deep learning and classifiers

Deep learning model	**Classifiers**			
	Random Forest	***Bagging***	***BayesNet***	***SMO***
AlexNet	71.561	56.683	71.317	80.415
CaffeNet	71.000	56.854	71.098	80.756
ResNet	81.585	69.781	82.073	87.781

selection and SMO classification which accurately classifies the food-ingredients in this dataset. To further evaluate the proposed DeepFood framework, it is compared with another work used the MLC-41 dataset. In [1], the top1 accuracy using AlexNet is about 60% and is less than 50% using CaffeNet, while the average accuracy of the proposed framework using AlexNet is 80.42%, and using CaffeNet is 80.76% as shown in Table 3. It is a significant improvement comparing to the method in [1].

Fig. 3 also shows a visualized performance comparison of the results. In this figure, the F1 Measure of each classifier on all food classes is depicted. As can be seen from the figure, the SMO outperforms other classifiers in all the classes, and the F1 measure reaches to 100% for several classes. Overall, the F1 measure plot of each classifier on almost all food classes is fluctuating in the range of 60% to 100%. Since our proposed framework exploits the extracted features using deep learning, it successfully improves the performance of multi-class classification of food ingredients.

In conclusion, the extensive experimental results demonstrate the effectiveness of the proposed DeepFood framework which has a very high performance for multi-class classification of food ingredients compared to other existing methods. Additionally, the DeepFood model combines the advantages of the ResNet deep feature sets, Information Gain (IG) feature selection, and the SMO classifier.

V. Conclusion

This paper proposes the DeepFood framework, an automatic multi-class classification of food ingredients using deep learning, which integrates different deep feature sets and several feature selections as well as an optimized classifier called SMO. The architecture is designed to classify small or medium datasets, which is a very general and necessary task in the real-world applications. However, it can be easily extended for large-scale data in the future. For a special purpose, it is employed to the multi-class classification of food ingredients.

The performance of the proposed architecture is evaluated with a series of experiments by comparing the accuracy among different deep learning models, various feature selections, and classifiers. The experimental results show the improvement and effectiveness of the DeepFood framework for multi-class classification of food ingredients.

ACKNOWLEDGMENT

This paper is partially supported by National Natural Foundation of China (Grant No. 61772561), Science Research of Hunan Provincial Education Department of China (Grand No. 16C1659), and Teaching Reform Project of Central South University of Forestry and Technology of China (Grant No. 1020208).

REFERENCES

[1] H. Chen, J. Xu, G. Xiao, Q. Wu, and S. Zhang, "Fast auto-clean CNN model for online prediction of food materials," Journal of Parallel and Distributed Computing. Kidlington, 2017, in press.

[2] S. Pouyanfar and S.-C. Chen, "Semantic concept detection using weighted discretization multiple correspondence analysis for disaster information management," in the 17th IEEE International Conference on Information Reuse and Integration, 2016, pp. 556-564.

[3] M.-L. Shyu, C. Haruechaiyasak, S.-C. Chen, and N. Zhao, "Collaborative filtering by mining association rules from user access sequences," in IEEE International Workshop on Challenges in Web Information Retrieval and Integration, 2005, pp. 128-135.

[4] X. Chen, C. Zhang, S.-C. Chen, and S. Rubin, "A human-centered multiple instance learning framework for semantic video retrieval," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 39, no. 2, pp. 228-233, 2009.

[5] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, "Effective supervised discretization for classification based on correlation maximization," in IEEE International Conference on Information Reuse and Integration, 2011, pp. 390-395.

[6] C. Chen, Q. Zhu, L. Lin, and M.-L. Shyu, "Web media semantic concept retrieval via tag removal and model fusion," ACM Transactions on Intelligent Systems and Technology, vol. 4, no. 4, pp. 1-22, 2013.

[7] T. Meng, and M.-L. Shyu, "Leveraging concept association network for multimedia rare concept mining and retrieval," in IEEE International Conference on Multimedia and Expo, 2012, pp. 860-865.

[8] K. Yanai, and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in IEEE International Conference on Multimedia & Expo Workshops, 2015, pp. 1-6.

[9] T. Joutou, and K. Yanai, "A food image recognition system with Multiple Kernel Learning," in 16th IEEE International Conference on Image Processing, 2009, pp. 285-288.

[10] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in IEEE International Symposium on Multimedia, 2010, pp. 296-301.

[11] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 1085-1088.

[12] G. E. Hinton, and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, New York, vol. 313, no. 5786, pp. 504-507, 2006.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.

[14] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition," in the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[15] M. Lin, Q. Chen, and S. Yan, "Network in Network," *CoRR*, vol. abs/1312.4400, 2013. [Online]. Available: http://arxiv.org/abs/1312.4400.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and et al, "Going deeper with convolutions," in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

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

[18] S. Pouyanfar, S.-C. Chen, and M.-L. Shyu, "An efficient deep residual-inception network for multimedia classification," in IEEE International Conference on Multimedia and Expo, 2017, pp. 373-378.

[19] Y. Kawano, and K. Yanai, "Food image recognition with deep convolutional features," in Proceedings of ACM UbiComp Workshop on Workshop on Smart Technology for Cooking and Eating Activities, 2014, pp. 589-593.

[20] S. Pouyanfar, and S.-C. Chen, "Semantic event detection using ensemble deep learning," in the IEEE International Symposium on Multimedia, 2016, pp. 203-208.

[21] 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 Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 157-166.

[22] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp, "Analysis of food images: Features and classification," in IEEE International Conference on Image Processing, 2014, pp. 2744-2748.

[23] D. T. Nguyen, Z. Zong, P. O. Ogunbona, Y. Probst, and W. Li, "Food image classification using local appearance and global structural information," Neurocomputing, vol. 140, pp. 242-251, 2014.

[24] G. Farinella, M. Moltisanti, and S. Battiato, "Classifying food images represented as bag of textons," in IEEE International Conference on Image Processing, 2014, pp. 5212-5216.

[25] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, and I. Essa, "Leveraging context to support automated food recognition in restaurants", in IEEE Winter Conference on Applications of Computer Vision, 2015, pp. 580-587.

[26] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," in IEEE International Conference on Image Processing, 2009, pp. 289-292.

[27] 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 Proceedings of SIGGRAPH Asia Technical Briefs, 2012.

[28] F. Zhu, M. Bosch, I. Woo, S. Kim, C. J. Boushey, D. S. Ebert, and E. J. Delp, "The use of mobile devices in aiding dietary assessment and evaluation," IEEE Journal of Selected Topics in Signal Processing, vol.4, no. 4, pp. 756-766, 2010.

[29] L. Bossard, M. Guillaumin, and L. V. Gool, "Food-101 – mining discriminative components with random forests," in European Conference on Computer Vision, 2014, pp. 446-461.

[30] H. Kagaya, and K. Aizawa, "Highly accurate food/non-food image classification based on a deep convolutional neural network," in International Conference on Image Analysis and Processing, 2015, pp. 350-357.

[31] S. Christodoulidis, M. Anthimopoulos, and S. Mougiakakou, "Food recognition for dietary assessment using deep convolutional neural networks," in International Conference on Image Analysis and Processing, 2015, pp.458-465.

[32] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: A new dataset, experiments, and results," IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 3, pp. 588-598, 2017.

[33] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks," In International Conference on Learning Representations, 2014, pp. 1-16.

[34] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675-678.

[35] N. Dalal, and B. Triggs, “Histograms of oriented gradients for human detection,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 886-893.

[36] G. Csurka, C. Bray, C. Dance, and L. Fan, “Visual categorization with bags of keypoints,” in Europe Conference on Computer Vision’04 Workshop on Statistical Learning in Computer Vision, 2004, pp. 59-74.

[37] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no.2, pp. 91-110, 2004.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587.

[39] 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 International Conference on Machine Learning, 2014, pp. 647-655.

[40] I. T. Jolliffe, “Principal Component Analysis”, ACM Computing Surveys, Springer Verlag, 1986. pp. 1–47.

[41] S. L. Y. Lam, D. L. Lee, “Feature reduction for neural network based text categorization”, in 6th International Conference on Database Systems for Advanced Applications, 1999, pp. 195-202.

[42] M. A. Hall, “Correlation-Based Feature Selection for Machine Learning”, Philosophy Thesis, University of Waikato, 1999.

[43] Y. Yang, J. O. Pedersen, “A comparative study on feature selection in text categorization”, in proceedings of the 14th International Conference on Machine Learning, 1997, pp. 412–420.

[44] J. Platt. Sequetial minimal optimization: A fast algorithm for training support vector machines. In Technical Report MST-TR-98-14. Microsoft Research, 1998.