Few-Shot Food Recognition via Multi-View Representation Learning

SHUQIANG JIANG, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, China
WEIQING MIN, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, China
YONGQIANG LYU, Qingdao KingAgroot Precision Agriculture Technology Co., Ltd, China
LINHU LIU, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, China

This paper considers the problem of few-shot learning for food recognition. Automatic food recognition can support various applications, e.g., dietary assessment and food journaling. Most existing works focus on food recognition with large numbers of labelled samples, and fail to recognize food categories with few samples. To address this problem, we propose a Multi-View Few-Shot Learning (MVFLS) framework to explore additional ingredient information for few-shot food recognition. Besides category-oriented deep visual features, we introduce ingredient-supervised deep network to extract ingredient-oriented features. As general and intermediate attributes of food, ingredient-oriented features are informative and complementary to category-oriented features, and thus play an important role in improving food recognition. Particularly in few-shot food recognition, ingredient information can bridge the gap between disjoint training categories and test categories. In order to take advantage of ingredient information, we fuse these two kinds of features by first combining their feature maps from their respective deep networks, and then convolving combined feature maps. Such convolution is further incorporated into a multi-view relation network, which is capable of comparing pairwise images to enable fine-grained feature learning. MVFLS is trained in an end-to-end fashion for joint optimization on two types of feature learning subnetworks and relation subnetworks. Extensive experiments on different food datasets have consistently demonstrated the advantage of MVFLS in multi-view feature fusion. Furthermore, we extend another two types of networks, namely Siamese Network and Matching Network by introducing ingredient information for few-shot food recognition. Experimental results have also shown that introducing ingredient information into these two networks can improve the performance of few-shot food recognition.


Additional Key Words and Phrases: Food recognition, few-shot learning, visual recognition, deep learning

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1551-6857/2020/1-ART1 $15.00
https://doi.org/10.1145/3391624
1 INTRODUCTION

Food recognition has received a significant amount of attention in various fields, such as computer vision [9], data mining [4] and multimedia communities [26, 58] motivated by many applications in automated food monitoring and dietary management [1, 13], food trend and popularity analysis [2], smart home [66] and food safety [38]. For example, people’s diet and eating behavior have been shown to affect the health issues [42]. This fact has fostered the emergence of many approaches to monitor diet. With the fast development of mobile devices, more and more dietary management systems resort to vision-based methods [5, 43]. One necessary and important step is to automatically recognize the type of food displayed in the image. Another example is that food image recognition is a key enabler for many smart home applications such as smart kitchen and smart personal nutrition log [66].

There are more than 8,000 food categories according to Wikipedia [8]. Like other object categories, there is a long-tailed distribution for real-world food data, where many categories have few food samples. For example, when you search some food names, such as “Wagafi bread” and “Babute” using existing popular search engines, such as Google and Bing, very few relevant images are returned. In other words, we can only obtain few examples for these food categories. A robust food recognition system not only recognizes usual food images, but also unusual ones. However, existing methods for food recognition need large-scale labeled samples for effective model training [9, 40]. As a result, they can not handle food categories with few samples. In order to solve this problem, in this paper, we focus on few-shot learning for food recognition, which aims to recognize novel visual food categories from few examples.

There has been a recent resurgence of interest in one/few-shot learning [10, 19, 34, 51, 56, 57, 60]. Various methods, such as Matching Network [60], Prototypical Network [56] and Relation Network [57] have been proposed for few-shot learning. These methods are applied to different domains, such as alphabet recognition and general object recognition. However, besides category information, most existing few-shot learning methods do not explore other types of context information, such as rich attributes and other side information, to enhance the performance of few-shot learning. In addition, as far as we know, there is no work on few-shot learning for food recognition.

Few-shot learning for food recognition is not trivial. The challenges derive from three-fold: First, food image recognition belongs to fine-grained classification [18, 65]. Similarly, food image recognition encounters the same problem as fine-grained classification, such as subtle differences among different food categories. In addition, we can not directly use existing fine-grained classification methods for food recognition. Existing fine-grained classification methods generally first discover common semantic parts (such as head and breast in the bird dataset), and then fuse features from both global object and semantic parts as final representation. However, the concept of common semantic parts for fine-grained classification methods does not exist in food images. Second, food images do not have any distinctive spatial layout in many cases. Although some food categories such as fruits and hamburgers have regular shapes, many food dishes are lack of rigid structures. Third, few-shot learning for food recognition brings new challenges, such as how to utilize limited samples from some categories to train a robust food recognition model.

In many recipe-sharing websites, there are also associated ingredients available. Ingredients can be one constituent part of food as intermediate attributes. It plays an important role in food recognition. For example, Fig. 1 shows two groups of food images. In Fig. 1 (a), these two images
from the same category have larger difference in the visual appearance. However, if we consider their ingredient attributes “Minced green onion”, “Fish” and “Sweet and sour sauce”, and combined the visual representation supervised by ingredient attributes with ones supervised by the category, the probability of these two images belonging to the same category increases. In Fig. 1 (b), it is difficult to distinguish between two categories from these two images. However, it is easier to distinguish between them from different ingredient attributes “Fish” and “Pork slices”, which are their main characteristics of food categories. If we can use these ingredients to learn to localize relevant image regions, the probability of these two images belonging to the same category probably decreases. Therefore, ingredient information provides discriminative information for food recognition. Particularly for few-shot food recognition, there are many categories with few samples. As general food attributes, ingredients can serve as important complementary information to improve the performance of few-shot food recognition, and also build the connection between disjoint food categories.

![Fig. 1. Examples of food images and associated ingredients from VIREO Food-172.](image)

Taking these factors into consideration, we propose a novel Multi-View Few-Shot Learning (MVFSL) framework to exploit rich food ingredients for few-shot learning in the food domain. As shown in Fig. 2, MVFSL mainly consists of three components: (a) Category and ingredient oriented feature learning; (b) Multi-view feature map fusion and (c) Multi-view relation learning. Particularly, MVFSL first extracts feature maps from both category-supervised deep network and ingredient-supervised one, respectively. As mid-level attributes, ingredient-oriented features are capable of capturing other details of food, which are complementary to category-oriented features. More specifically, the feature maps of a deep convolutional layer tend to be selective of visual concepts [63]. Therefore, we deem feature maps from ingredient-supervised deep networks focus on salient image regions, which are different from category-oriented feature maps. Then, multi-view feature map fusion is conducted to fuse these two types of feature maps. It first combines extracted feature maps from their respective deep networks, and then convolves combined feature maps. Such convolution is finally involved in multi-view relation learning, which is used to compare pairwise images for metric learning. In multi-view relation learning, a multi-view relation network with convolutions and full-connected layers is utilized to apply the convolution to both combined feature maps within each image for multi-view feature fusion and feature maps between two images.
for image-level fine-grained feature learning. Furthermore, MVFSL can be trained end-to-end to enable joint optimization on different subnetworks.

We conduct comprehensive experimental evaluation on various food benchmarks including western food datasets, such as Food-101 [9] and eastern food datasets, such as VIREO Food-172 [11] and ChineseFoodNet [12]. The experimental results demonstrate the effectiveness of MVFSL in multi-view feature fusion. Furthermore, we extend another two few-shot learning networks including Siamese Network and Matching Network by introducing ingredient information for few-shot food recognition. The experimental results also demonstrate the advantage of these two few-shot learning methods using ingredient information.

The contributions of our paper can be summarized as follows:

- To the best of our knowledge, this is the first attempt to apply few-shot learning for food recognition, where rich ingredient information is utilized to improve the performance of few-shot food recognition.
- We propose a Multi-View Few-Shot Learning (MVFSL) framework to exploit rich food ingredients for few-shot food recognition. In MVFSL, multi-view feature map fusion is developed to effectively fuse both ingredient-oriented features and category-oriented ones via the convolution on fused feature maps from multi-view relation subnetwork learning. Furthermore, MVFSL can be trained in an end-to-end manner to enable joint optimization on different subnetworks.
- We conduct comprehensive experimental evaluation on various food benchmarks and experimental results verify the effectiveness of MVFSL. Furthermore, we extend another two few-shot learning methods by introducing ingredient information for few-shot food recognition. The experimental results again demonstrate the advantage in exploiting ingredient information. The source code can be available\(^1\).

The rest of this paper is organized as follows. Section 2 elaborates the proposed Multi-View Few-Shot Learning (MVFSL) framework, where three components are introduced in details, respectively. Section 3 introduced another two types of extended few-shot learning methods using ingredient information, namely Siamese Network and Matching Network, respectively. Experimental results and analysis are reported in Section 4. Section 5 reviews related work. Finally, we conclude the paper and give future work in Section 6.

2 MULTIVIEW FEW-SHOT LEARNING (MVFSL)

As shown in Fig. 2, MVFSL mainly consists of three parts: (a) Category and ingredient oriented feature learning; (b) Multi-view feature map fusion and (c) Multi-view relation learning. We first briefly introduce few-shot learning for completeness before diving deep into MVFSL.

2.1 Few-Shot Learning

For few-shot image learning, there are two types of sets, namely support set \(S\) and query set \(Q\). \(C\) unique classes with \(K\) labeled images for each of \(C\) classes are randomly sampled from the training set to form \(S = \{(x_i, y_i)\}_{i=1}^m\) (\(m=K\times C\)), where \(x_i\) is one sample image and \(y_i\) is its label. The query set \(Q = \{(x_j, y_j)\}_{j=1}^n\) is constructed using remaining samples of selected \(C\) classes. The support set and query set together form a training episode. Typically, \(K\) is a small number for few-shot setting, e.g., \(K = 1\) or \(5\). The task is denoted as \(C\)-way \(K\)-shot learning. In the training, pairwise images from \(Q\) and \(S\) are constructed for model learning. In the test stage, we adopt similar strategy to construct both \(Q\) and \(S\) from the test set and classifies the images from a query set by assigning each image with a label. Note that categories from both training set and test set are disjoint. The

\(^1\) https://github.com/minweiqing/Few-Shot-Food-Recognition-via-Multi-View-Representation.
time cost of few-shot food recognition does not depend on the number of the whole class, but depends on the value of selected $K$ and $C$. Therefore, there is no direct relevance between the time cost of few-shot food recognition and the number of the whole classes for each test sample in the test phrase. Generally, image pairs from $Q$ and $S$ constructed from the training set can be very large, and deep learning networks thus can be utilized for few-shot learning [37, 57].

2.2 Category and Ingredient Oriented Feature Learning

For food category oriented feature learning, we use the training set with categories to fine-tune one deep network. We then extract feature maps $f_\mu(x_i)$ of the last convolution layer, where $\mu$ are model parameters. In addition, we use images from the training set and their associated multi-label ingredients to fine-tune another deep network for multi-label ingredient attribute learning, and extract ingredient-oriented feature maps $f_\nu(x_i)$, where $\nu$ denotes parameters of the ingredient-oriented deep network. In multi-label ingredient attribute learning, we consider multi-label ingredient learning with $M$ ingredients as $M$ binary attribute classification tasks, where $M$ is the size of ingredient vocabulary. Note that any deep networks can be used in this stage. Without loss of generality, we adopt VGG-16 [55] as the backbone network to introduce our method.

2.3 Multi-View Feature Map Fusion

Feature maps of a deep convolutional layer are usually sparse and tend to be selective of higher-level visual concepts, as observed in [54, 63]. In order to introduce our method, we first demonstrate three images from VIREO Food-172 and their visualization results in Fig 3, where the visualization is realized via Grad-CAM [54]. Note that for ingredients, we train a multi-label classification model and then obtain discriminative localization regions via gradients for each ingredient label. From Fig 3, we find that activated regions of many feature maps (highlighted in warm colors) are
Fig. 3. Discriminative localization maps from some food images. Grad-CAM [54] is adopted to implement the category and ingredient discriminative localization region visualization (the warmer the color of the overlay image, the more discriminative that pixel is). From left to right: (1) Category and its ingredients, (2) Original images, (3) Category-discriminative localization maps and (4) Ingredient-discriminative localization maps.

Semantically meaningful. For example, the activated region of “Rice” tends to localize at the rice region of “Fried Sweet and Sour Tenderloin” and “Barbecued Pork with Rice”. The activated region of “Parsley” tends to localize at the parsley region of “Braised Intestines in Brown sauce”. Therefore, ingredient-oriented features are capable of capturing additional details, which are complementary to category-oriented features. For example, for “Braised Intestines in Brown sauce”, the ingredient region covers different parts of the image, such as activated regions of “Parsley” and “Minced green onion”, which are complementary with activated regions of the category.

Based on above-mentioned observation and analysis, we could combine category and ingredient activated regions for enhanced feature representation. Particularly, we calculate the combined feature map representation $\zeta(f_{\mu}(x_i), f_{\nu}(x_i))$ via the operator $\zeta(\cdot)$. There are many ways for feature combination. In this paper, the operator $\zeta(\cdot)$ is the concatenation of feature maps in depth. For example, the extracted feature maps form the last convolution layer with $14 \times 14 \times 512$ from two types of fine-tuned deep networks, the size of combined features will be $14 \times 14 \times 1024$ via $\zeta(\cdot)$.

After the combination between feature maps from two different types, we then conduct the convolution on these combined feature maps. Meanwhile, the convolution is conducted with the following multi-view relation learning together for parameter learning and will give more details in the following subsection. Our adopted multi-view feature map concatenation and convolution fusion is similar to [16]. However, they focus on spatiotemporal fusion from video for activation classification. In contrast, we conduct both ingredient-oriented and category-oriented fusion for few-shot food image recognition. In addition, the fused features contain richer information from food attributes, and thus bridge the gap between disjoint training categories and test categories for the performance improvement of few-shot food recognition.
2.4 Multi-View Relation Learning

Multi-view relation learning is used to compare query images against labeled sample images to determine if these images are from matching categories or not based on the image-level relation score.

For multi-view relation learning, we sample the fused multi-view representation $\xi(f_\mu(x_i), f_\nu(x_i))$ from the support set $S$ and $\xi(f_\mu(x_j), f_\nu(x_j))$ from the query set $Q$. These feature maps $\xi(f_\mu(x_i), f_\nu(x_i))$ and $\xi(f_\mu(x_j), f_\nu(x_j))$ are combined via $\tau(\cdot)$, where the operator $\tau(\cdot)$ is also the concatenation of feature maps in depth. For example, the fused feature maps from both $x_i$ and $x_j$ are $\xi(f_\mu(x_i), f_\nu(x_i))$ and $\xi(f_\mu(x_j), f_\nu(x_j))$, respectively. Their size is $14 \times 14 \times 1024$. After concatenation $\tau(\cdot)$, the final dimension is $14 \times 14 \times 2048$. The combined feature maps of samples from the query and support set are further processed via the relation subnetwork $h_\phi$ with some convolutional blocks and full-connected layers to generate the relation score:

$$y_{i,j} = h_\phi(\tau(\xi(f_\mu(x_i), f_\nu(x_i)), \xi(f_\mu(x_j), f_\nu(x_j)))) \quad (1)$$

The mean square error is then used to train the model, regressing the relation score $y_{i,j}$ to the ground truth: matched pairs have similarity 1 and mismatched pairs have similarity 0. The final objective is as follows:

$$\argmin_\phi \sum_{i=1}^{m} \sum_{j=1}^{n} (y_{i,j} - 1(y_i == y_j))^2 \quad (2)$$

where $m$ is the number of images from the support set $S$ and $n$ is the number of images from the query set $Q$.

Note that the proposed MVFSL is inspired by Relation Network (RN) \cite{8} for few-shot learning, but with two important differences: (1) RN only learns category-oriented visual features for few-shot learning, while MVFSL can further utilize ingredient-oriented features for few-shot learning. (2) RN only applies convolution to category-oriented feature maps. In contrast, we utilize the convolution on fused multi-view feature maps for multi-view relation learning. Through the convolution on feature maps with two different types, category-oriented and ingredient-oriented features are effectively fused. In addition, the convolution is further conducted based on feature maps between two images for image-level relation learning.

2.5 Optimization

We introduce two settings for training MVFSL. In the first setting, we first fine-tune food category-supervised deep network and ingredient-supervised deep network, respectively. The corresponding feature maps are extracted. These two types of feature maps are then fused and the fused features are finally fed into the relation network for multi-view relation learning. Such setting is Loosely Combined, and named as MVFSL-LC. In the second setting, after fine-tuning both food category-supervised deep network and ingredient-supervised deep network, the whole training on these two types of subnetworks and relation subnetworks in MVFSL are further conducted in an end-to-end fashion for joint optimization. This setting is Tightly Combined and named as MVFSL-TC.

In the test stage, we obtain combined features from the support set and query set via multi-view feature map fusion, and $h_\phi$ is then used to generate the relation score between query set and each of support set. Finally we can make the prediction according to the maximum relation score.

3 BASELINE METHODS OF INGREDIENT BASED FEW SHOT LEARNING NETWORKS

The ingredients can also be fused to other few-shot learning methods. In this section, we introduce ingredients into another two popular networks of few-shot learning methods, namely Siamese
Network (SN) [34] and Matching Network (MN) [60], where multi-view relation learning is replaced with fixed cosine distance calculation between two images.

3.1 Siamese Network

When Siamese Network (SN) is used for few-shot learning, it first randomly samples image pairs with the same class or different classes from the training set to learn the model. In the test stage, the trained model is used to classify the unlabelled image $X$ into one of $C$ categories from the test set. Given an unlabeled image $X$ and other images $\{X_c\}_{c=1}^C$, where $X_c$ represents one example with the label $c$. The image pairs $(X, X_c)_{c=1}^C$ are fed into the trained model to make the prediction for $X$ according to the cosine similarity between this image and other samples from those $C$ categories.

In order to exploit ingredient information, for category oriented feature learning, we replace the original network of SN with VGG-16 for fair comparison, and then fine-tune the network under the few-shot learning setting to extract more discriminative visual features $f_\mu(x_i)$ for each test image $x_i$. For ingredient oriented feature learning, each food image is associated with multiple ingredients. However, SN can not be trained for multi-label classification. To solve this problem, we first fine-tune this network for multi-label ingredient classification. We then initialize SN using this fine-tuned network, and further fine-tune the last layer of SN for few-shot food recognition. Finally, we use the fine-tuned SN to extract ingredient-oriented features $f_\nu(x_i)$ for each test image $x_i$. A multi-view feature fusion $\zeta(\cdot)$ is used to combine these two kinds of features $f_\mu(x_i)$ and $f_\nu(x_i)$ into a unified representation. Because we adopt fixed cosine similarity calculation, the corresponding features from the fine-tuned network are from the fc7 layer. We finally use the fused features to predict the unlabelled image $x$ according to the cosine similarity between this image and other images from $C$ categories.

3.2 Matching Network

Matching Network (MN) learns a network that maps a small labelled support set and an unlabeled query example to its label. In the training stage, it learns a classifier $c_{S}(x)$ from the support set $S_{train}$ that constructed from the training set. The classifier can be defined as one mapping function $P_{\theta}(x|y, S_{train})$, where $\theta$ are parameters of the model, and should be learned in the training. In the test stage, given an unlabeled example $x_t$ and a support set $S_{test}$ from the test set, MN uses learned $P_{\theta}(\cdot|x_t, S_{test})$ to predict its label $y_t$.

We adopt similar strategy to extend MN by exploiting ingredient information. We obtain fused features for each test sample under the subjective function of MN. Then cosine similarity between the query sample and each sample in the support set is calculated to make prediction.

4 EXPERIMENT

In this section, we first describe the experimental setting including the dataset and implementation details. We then evaluate the performance of MVFSL qualitatively and quantitatively on different food datasets. Next, we evaluate the performance of another two extended few-shot learning methods Siamese Network and Matching Network. Finally, we give additional analysis and discussions.

4.1 Experimental setting

4.1.1 Dataset. Since there is no food dataset for few shot learning, we use the following three food datasets, namely Food-101 [9], VIREO Food-172 [11] and ChineseFoodNet [12] to simulate few-shot food recognition.

Food-101 contains 101,000 images with 101 classes in total, where most categories belong to the western food. In order to use the dataset to simulate few-shot food recognition, similar to
we randomly split Food-101 into 71 classes and 30 classes for the training set and test set, respectively. For ingredient information, we adopt the ingredients from Ingredients101 \cite{8} with 446 ingredients in total and 9 ingredients for each image on average.

**VIREO Food-172** contains 172 categories. All the images in the data set are Chinese food. Similarly, we randomly split the data set into 132, 40 classes for the training and test set. There are 353 ingredients in total with 3 ingredients for each image on average.

**ChineseFoodNet** covers many popular Chinese food items from different styles of cooking. This dataset contains 185,628 images with 208 Chinese dish categories. We randomly split the data set into 158 classes for training and 50 classes for testing, respectively. Both VIREO Food-172 and ChineseFoodNet belong to Chinese cuisine, and thus share lots of ingredients. Considering the ingredient information is not provided in ChineseFoodNet, we adopt the trained ingredient-supervised deep network from VIREO Food-172 as the ingredient model for ChineseFoodNet.

Fig. 4 shows some samples with ingredients from three datasets, respectively. Note that many works, such as \cite{57, 60} conduct few-shot learning in the miniImageNet dataset with 100 classes, each having 600 examples. 80 classes are used for training and the remaining 20 classes for test. The few-shot food recognition belongs to this scenario. These food dataset is similar to this miniImageNet with similar scale and similar training-test class split, and therefore can be regarded as in real scenarios for few-shot learning.

### 4.1.2 Implementation Details

In MVFSL, there are three subnetworks, category-oriented subnetwork, ingredient-oriented subnetwork, and multi-view relation subnetwork. The first two types of subnetworks adopt VGG-16 network without fully-connected layers and the classification layer. Similar to \cite{57}, multi-view relation subnetwork consists of two convolutional blocks and two fully-connected layers, where each of convolutional blocks is a $3 \times 3$ convolution with 64 filters followed by the batch normalization, the ReLU non-linearity and $2 \times 2$ maxpooling. The two fully-connected layers are 8 and 1 dimensional, respectively.

For our model, following existing few-shot learning setting \cite{57}, an episode-based training strategy is adopted. We randomly select $C = 5$ classes and sample one image $K = 1$ from each selected category as the support set $S$. That is, a common 5-way 1-shot setting is adopted in the training stage. For the query set $Q$, 15 query images are sampled from each selected category. There are $15 \times 5 + 1 \times 5 = 80$ images in each episode. In the training stage, we sample 100,000 episodes from the training set. The Adam \cite{33} is used to perform stochastic optimization over few-shot learning.
with the initial learning rate $10^{-4}$ and reduced by half for every 20,000 episodes. The accuracy of few-shot classification is computed by averaging over 1,000 episodes randomly generated from the test set.

### 4.2 Experimental Evaluation for MVFSL

#### Table 1. Performance comparison on MVFSL

<table>
<thead>
<tr>
<th>Model</th>
<th>Food-101</th>
<th>VIREO Food-172</th>
<th>ChineseFoodNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN-Category [57]</td>
<td>53.9%</td>
<td>74.0%</td>
<td>63.8%</td>
</tr>
<tr>
<td>RN-Ingredient</td>
<td>53.5%</td>
<td>70.5%</td>
<td>64.0%</td>
</tr>
<tr>
<td>MVFSL-LC</td>
<td>55.1%</td>
<td>74.8%</td>
<td>65.8%</td>
</tr>
<tr>
<td>MVFSL-TC</td>
<td>55.3%</td>
<td>75.1%</td>
<td>66.1%</td>
</tr>
</tbody>
</table>

#### 4.2.1 Quantitative Evaluation. Considering there are no methods for few-shot food recognition, we design the following baselines to demonstrate the effectiveness of MVFSL:

- Relation Network-Category (RN-Category) [57]: This baseline uses images and their categories to train the Relation Network.
- Relation Network-Ingredient (RN-Ingredient): This baseline uses the ingredient information and images to train the Relation Network.

We conduct comprehensive evaluation on three datasets, respectively. Note that we should utilize samples from VIREO Food-172 to learn ingredient-oriented model for ChineseFoodNet. In the training stage, we first remove test classes of ChineseFoodNet from the training set of VIREO Food-172 dataset, and then use the rest training images and their corresponding ingredients from VIREO Food-172 to learn ingredient based model, which can be used to extract ingredient-oriented features for ChineseFoodNet. Table 1 summarizes the experimental results. We can see that MVFSL-LC and MVFSL-TC on three datasets perform better than their corresponding baselines. Particularly, for Food-101, MVFSL-LC achieves better performance compared with RN-Category and RN-Ingredient, and outperforms those two baselines by 1.2% and 1.6%, respectively; MVFSL-TC achieves best performance compared with MVFSL-LC. We can see similar trends for both VIREO Food-172 and ChineseFoodNet. All these experimental results validated the effectiveness of MVFSL in fusing both category-oriented and ingredient-oriented feature representations. In addition, ingredient oriented information can bridge the gap between disjoint training categories and test categories to enable the performance improvement. In addition, MVFSL-TC jointly optimizes parameters from all subnetworks and thus can achieve the best performance.

#### 4.2.2 Qualitative Evaluation for MVFSL. We further demonstrated the effectiveness of MVFSL by showing some cases. Fig. 5 shows some examples from MVFSL. We can see that (1) RN-Ingredient could make more reasonable predictions compared with RN-Category in many cases. For some cuisines, such as “Mixed rice”, “Duck neck” and “Spinach and pork liver soup”, RN-Category failed to make the correct prediction, while RN-Ingredient made the correct prediction. (2) MVFSL could make accurate prediction for some cuisines that are quite difficult to recognize from the support set, such as “Spring rolls”, “Clam chowder” and “Fired Sweet and Sour Tenderloin”. Both RN-Category and RN-Ingredient failed to make the correct prediction while MVFSL has made the correct prediction by fusing category-oriented and ingredient-oriented visual feature information. This further verified that category oriented features and ingredient oriented features are complementary, and MVFSL is capable of fusing these two types of information to improve the performance of few-shot food recognition.
Fig. 5. Some experimental results from MVFSL and other two baselines. From left to right: (1) category and its ingredient list, (2) query images and (3) images from the support set. In each example, we show the relation scores from MVFSL and other two baselines. The higher relation score, the more relevant the image is. The ground truth of food images is highlighted with red box.

4.2.3 Multi-view Relation Learning with Different Convolution Layers. For Relation Network, the combined feature maps are fed into the relation subnetwork to obtain the relation score. In this experiment, we conducted the performance analysis when varying the number of convolution blocks and filters. Table 2 shows experimental results on Food-101 from MVFSL-LC and MVFSL-TC. We can see that: (1) When we fix the number of filters, with the increase of the number of convolution layers, there is a consistent increase in the performance. (2) The network with 128 filters achieves better performance than the one with 64 filters. For example, in MVFSL-LC, with the increase of the number of convolution layers, the network with 128 filters achieves better
performance than the network with 64 filters and outperforms it by 0.2%, 1.2%, 0.8% and 1.0% for different convolutional layers, respectively. (3) At every group of parameter setting, MVFSL-TC achieves better performance than MVFSL-LC. (4) MVFSL achieved the best performance at the setting of 3 convolution layers and 128 filters. With the increase of the number of layers and filters, the performance of MVFSL reduces. This is because the increased complexity of the model probably leads to overfitting.

Table 2. The performance of MVFSL-LC and MVFSL-TC with different relation network settings on Food-101.

<table>
<thead>
<tr>
<th>Model</th>
<th>64 filters</th>
<th>128 filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVFSL-LC (1 Conv layer)</td>
<td>54.7%</td>
<td>55.9%</td>
</tr>
<tr>
<td>MVFSL-TC (1 Conv layer)</td>
<td>55.1%</td>
<td>56.0%</td>
</tr>
<tr>
<td>MVFSL-LC (2 Conv layers)</td>
<td>55.1%</td>
<td>56.3%</td>
</tr>
<tr>
<td>MVFSL-TC (2 Conv layers)</td>
<td>55.3%</td>
<td>56.4%</td>
</tr>
<tr>
<td>MVFSL-LC (3 Conv layers)</td>
<td>56.1%</td>
<td>56.9%</td>
</tr>
<tr>
<td>MVFSL-TC (3 Conv layers)</td>
<td>56.6%</td>
<td>57.6%</td>
</tr>
<tr>
<td>MVFSL-LC (4 Conv layers)</td>
<td>54.6%</td>
<td>55.6%</td>
</tr>
<tr>
<td>MVFSL-TC (4 Conv layers)</td>
<td>55.7%</td>
<td>56.3%</td>
</tr>
</tbody>
</table>

Table 3. The performance with different networks for feature learning on Food-101, VIREO Food-172 and ChineseFoodNet.

<table>
<thead>
<tr>
<th>Model</th>
<th>Food-101</th>
<th>VIREO Food-172</th>
<th>ChineseFoodNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AlexNet VGG16 VGG19</td>
<td>AlexNet VGG16 VGG19</td>
<td>AlexNet VGG16 VGG19</td>
</tr>
<tr>
<td>RN-Category [57]</td>
<td>48.6% 53.9% 54.7%</td>
<td>68.8% 74.0% 74.6%</td>
<td>59.5% 63.8% 62.9%</td>
</tr>
<tr>
<td>RN-Ingredient</td>
<td>51.4% 53.5% 55.1%</td>
<td>70.3% 70.5% 73.6%</td>
<td>59.4% 64.0% 65.6%</td>
</tr>
<tr>
<td>MVFSL-LC</td>
<td>51.8% 55.1% 55.9%</td>
<td>70.8% 74.8% 75.2%</td>
<td>59.9% 65.8% 66.5%</td>
</tr>
<tr>
<td>MVFSL-TC</td>
<td>52.1% 55.3% 56.5%</td>
<td>71.0% 75.1% 75.3%</td>
<td>60.2% 66.1% 66.7%</td>
</tr>
</tbody>
</table>

4.2.4 The depth of basic network for MVFSL. We show that the performance of our method with different layers of deep networks. As shown in Table 3 on Food-101, we can see that: (1) With the increase of layers, there is consistent increase in the performance. For example, our method and baselines with the VGG16 network achieve better performances than AlexNet, and outperform AlexNet based RN-Category, RN-Ingredient, MVFSL-LC and MVFSL-TC by 5.3%, 2.1%, 3.3% and 3.2%, respectively. The reason is that the deeper network can extract more discriminative features, which are helpful for few-shot food recognition. (2) The performance of MVFSL is always better than other baselines for the same deep architecture. Particularly, for the AlexNet network, the performance of MVFSL-LC outperforms RN-Category and RN-Ingredient by 3.2% and 0.4%. For the VGG16 Network, there is also the performance improvement compared with RN-Category and RN-Ingredient, and outperforms those two models by 2.1% and 1.6%, respectively. We can see similar performance improvement for the model with VGG19 Network. (3) MVFSL-TC with different layers of deep networks achieves consistent increase in the performance compared with MVFSL-LC. For example, for VGG19 based MVFSL, MVFSL-TC outperforms MVFSL-LC by 0.6%.

Table 3 also provides comparative results on VIREO Food-172 and ChineseFoodNet. Though there are different categories between ChineseFoodNet and VIREO Food-172, both of them are Chinese food and their ingredients are similar, and we thus could still obtain the ingredient representation...
of ChineseFoodNet using the trained ingredient model. We can observe that: (1) With the increase of layers, MVFSL and other baselines similarly obtain consistent performance increase. The MVFSL and other baselines with the VGG16 network outperform those models with AlexNet by 0.2% to 5.2%. Similarly, in most cases, the performance of MVFSL and other baselines with VGG19 network is better than those model with VGG16 network. There is only one exception that RN-Category with VGG19 achieves lower performance compared with VGG16. One probable reason is that there is overfitting in the network training on this dataset. (2) There is consistent increase for MVFSL compared with the baselines for each architecture with the same layers. (3) For MVFSL-TC with AlexNet, VGG16 and VGG19, there is consistent performance increase compared with MVFSL-LC. This again verified the effective of end-to-end training.

4.3 Experimental Evaluation for SN and MN

For the evaluation setting of SN, MN and RN, we conduct the comparison according to [57] for fair evaluation. For SN, we need to construct positive image pairs and negative image pairs, which should conform the setting of 5-way 1-shot with 1 query. Participially, we randomly select 5 categories and sample one image from each selected category as the support set. One query image is sampled from one of those selected categories. We construct the positive image pair and negative image pair by combining the query image with each image from the support set, where the positive image pair means these images are from the same class and negative image pair means they are from different classes. There are 4 negative pairs and 1 positive pair in each 5-way 1-shot with 1 query setting. We randomly sample 50,000 image pairs with 10,000 positive pairs and 40,000 negative pairs in each training epoch. The classification accuracy is computed by averaging over 5,000 randomly generated image pairs from the test set. Under the 5-way 1-shot with 1 query setting, 5,000 randomly generated image pairs are equivalent to 1000 tasks. Nesterow Momentum is used to perform stochastic optimization with the initial learning rate $5 \times 10^{-5}$ and momentum value 0.9.

For MN, we randomly select 5 categories and sample one image from each selected category as the support set, and also one query image from one of those selected categories as the query set, which means there are $5 \times 1 + 1 = 6$ images in each episode. In the training stage, we sample 100,000 episodes from the training set. Adam [33] is used for stochastic optimization with the initial learning rate $5 \times 10^{-4}$ and half the learning rate for every 20,000 episodes. The few-shot classification accuracy is computed by averaging over 1,000 randomly generated episodes from the test set.

To further demonstrate the effectiveness in introducing the ingredient information, we use the following baselines to evaluate SN-Multiview and MN-Multiview:

- Siamese Network-Category (SN-Category) [34]: This baseline uses images and their categories to train the Siamese Network.
- Siamese Network-Ingredient (SN-Ingredient): This baseline uses images and their ingredients to train the Siamese Network.
- Matching Network-Category (MN-Category) [60]: This baseline uses images and their categories to train the Matching Network.
- Matching Network-Ingredient (MN-Ingredient): This baseline uses images and their ingredients to train the Matching Network.

Table 4 and Table 5 provide the comparative results. We can observe that there is a consistent increase for both SN-Multiview and MN-Multiview compared with their corresponding baselines. Particularly, for SN-Multiview, we can achieve performance improvement compared with the baselines and outperforms them by 4% to 5%. Similarly, MN-Multiview also achieves performance improvement compared with the baselines, and outperform them by 0.4% to 8.2%. This further verified the effectiveness of our proposed method in exploiting ingredient information.
Table 4. Performance comparison on SN

<table>
<thead>
<tr>
<th>Model</th>
<th>Food-101</th>
<th>VIREO Food-172</th>
<th>ChineseFoodNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN-Category [34]</td>
<td>49.1%</td>
<td>60.3%</td>
<td>50.5%</td>
</tr>
<tr>
<td>SN-Ingredient</td>
<td>54.5%</td>
<td>65.5%</td>
<td>62.5%</td>
</tr>
<tr>
<td>SN-Multiview</td>
<td>55.0%</td>
<td>63.8%</td>
<td>64.4%</td>
</tr>
</tbody>
</table>

Table 5. Performance comparison on MN

<table>
<thead>
<tr>
<th>Model</th>
<th>Food-101</th>
<th>VIREO Food-172</th>
<th>ChineseFoodNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN-Category [60]</td>
<td>45.6%</td>
<td>73.6%</td>
<td>48.9%</td>
</tr>
<tr>
<td>MN-Ingredient</td>
<td>46.8%</td>
<td>65.9%</td>
<td>52.0%</td>
</tr>
<tr>
<td>MN-Multiview</td>
<td>47.5%</td>
<td>74.1%</td>
<td>53.0%</td>
</tr>
</tbody>
</table>

Table 6. Performance comparison on different methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Food-101</th>
<th>VIREO Food-172</th>
<th>ChineseFoodNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN-Multiview</td>
<td>55.0%</td>
<td>65.8%</td>
<td>64.4%</td>
</tr>
<tr>
<td>MN-Multiview</td>
<td>47.5%</td>
<td>74.1%</td>
<td>53.0%</td>
</tr>
<tr>
<td>MVFSL-LC</td>
<td>55.1%</td>
<td>74.8%</td>
<td>66.5%</td>
</tr>
<tr>
<td>MVFSL-TC</td>
<td>55.3%</td>
<td>75.1%</td>
<td>66.7%</td>
</tr>
</tbody>
</table>

Table 6 further summarized experimental results among MVFSL-TC, MVFSL-LC, SN-Multiview and MN-Multiview. We can see that (1) The performance of SN-Multiview is better than MN-Multiview in most cases. This trend is consistent with experiment results on other datasets [57] (2) The performance of MVFSL-TC is better than MVFSL-LC. The reason is that MVFSL-TC is capable of making deep feature learning and relation learning reinforce each other. (3) MVFSL-TC achieves the best performance compared with other three methods. For Food-101, MVFSL-TC achieves the best performance compared with MVFSL-LC, SN-Multiview and MN-Multiview, and outperforms them by 0.2%, 0.3% and 7.8%. Similarly, MVFSL-TC achieves the best performance for VIREO Food-172 and ChineseFoodNet.

4.4 Experimental Evaluation for Different C-way K-shot

In this section, we conducted the comparison under different C-way K-shot settings. Table 7 provide experimental results on three datasets. We can observe that (1) there is consistent increase for MVFSL-LC and MVFSL-TC on different C-way K-shot setting compared with RN-Category and RN-Ingredient. Furthermore, there is marginal improvement for end-to-end training and MVFSL-TC achieves best performance. For example, on the Food-101, for the 5-way 1-shot setting, MVFSL-TC achieves better performance compared with RN-Category and RN-Ingredient, and outperforms those two baselines by 1.2% and 1.6%, respectively and MVFSL-TC achieves better performance compared with MVFSL-LC. Also similar trends for other C-way K-shot settings.

4.5 Discussions

4.5.1 What causes the different performances? We can notice that the performance on both VIREO Food-172 and ChineseFoodNet are better than the Food-101 dataset. The probable reason lies in the difference between the training set and test set. In the few-shot learning, the classes from the training set and test set are disjoint. Although introducing the ingredient information can release this problem, if there is larger difference between the distribution of the training set and the test data,
Table 7. The performance on different networks with different C-way K-shot settings on three datasets

<table>
<thead>
<tr>
<th>Model</th>
<th>Food-101</th>
<th>VIREO Food-172</th>
<th>ChineseFoodNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5-way 1-shot</td>
<td>5-way 3-shot</td>
<td>5-way 1-shot</td>
</tr>
<tr>
<td>RN-Category</td>
<td>53.9%</td>
<td>67.4%</td>
<td>63.8%</td>
</tr>
<tr>
<td>RN-Ingredient</td>
<td>53.5%</td>
<td>67.5%</td>
<td>70.5%</td>
</tr>
<tr>
<td>MVFSL-LC</td>
<td>55.1%</td>
<td>68.1%</td>
<td>74.8%</td>
</tr>
<tr>
<td>MVFSL-TC</td>
<td>55.3%</td>
<td>68.3%</td>
<td>75.1%</td>
</tr>
</tbody>
</table>

Fig. 6 shows some examples selected from Food-101 and VIREO Food-172. For Food-101, the appearance of the training set are quiet different from test set. In contrast, for VIREO Food-172, there is a more similarity in some aspects (e.g.,the appearance and color) for some categories, such as “Roast Leek” from the training set and “Salt Green Tender” from the test set, and “Fired Sweet and Sour Tenderloin” of the training set and “Braised Pork” of the test set. Such similar distribution between training set and test set enables transfer learning between these two sets for performance improvement.

4.5.2 Visualization. In this section, we qualitatively analyze the results of relation learning. We randomly sample 2 categories with 100 images from the Food-101 dataset and all the images are projected to 2D by PCA. Fig. 7(a) shows real sample images colored by matching (red) or mismatching (green) images and query image (yellow). We can see that comparing embeddings of original images are very challenging. In Fig. 7(b-d), we then qualitatively show the relation representations of RN-Category, RN-Ingredient and MVFSL-LC based matched(red) and mismatched(green) query-sample pairs, respectively. Similar to [57], we plot each query-sample pair that represented by relation module pair representations. We can see that relation network has mapped query-sample pairs into a linearly separable space. In addition, MVFSL-LC based (mis)matched query-sample pairs are more linearly separable.

![Fig. 6. Examples of Food-101 and VIREO Food-172](image-url)
Fig. 7. Examples of Food-101 few-shot problem visualizations. (a) Matched (red) and mismatched (green) sample embeddings for a given query (yellow); (b) RN-Category: Matched (red) and mismatched (green) relation module pair representations; (c) RN-Ingredient: Matched (red) and mismatched (green) relation module pair representations; and (d) MVFSL-LC: Matched (red) and mismatched (green) relation module pair representations.

5 RELATED WORK

Our work is closely related to the following two research areas: 1) food image classification, and 2) few-shot learning.

5.1 Food Image Recognition

Recently, Min et al. [45] provided a comprehensive survey of food recognition and other food-related works. Food recognition is a difficult problem since foods can dramatically vary in appearance. Such variations may arise not only from changes in illumination and viewpoint, but also from non-rigid deformations, and intra-class variability in shape, texture, color and other visual properties. Existing food recognition researches can be categorized into two major directions: 1) conventional approaches and 2) deep learning approaches. For conventional approaches, Yang et al. [62] exploited the spatial relationship among different ingredients. The food items are represented by pairwise statistics between local features of the different ingredients of the food items. This approach is bound to work only for standardized meals. Bossard et al. [9] used the random forest method to mine discriminative parts of food images for recognition. Except the such works, plenty of researches has been carried out to find the optimal hand-crafted representation for food recognition. Joutou et al. [29] exploited several kinds of image features together with a multiple kernel learning. They combined Bag-of-SIFT with color histograms and Gabor filters to discriminate between images. Martinel et al. [41] proposed a complex scheme that can independently classify each feature response through an extreme learning machine, and combine the classification results by using a structured SVM. Kawano et al. [32] developed a real-time mobile food recognition system that it performs HoG and color patch feature encoding via fisher vector.

Compared with the conventional approaches, many deep learning based methods have been developed for food recognition [14, 30, 31]. For example, Kawano et al. [31] found that deep features performed significantly better than hand-crafted features. Kagaya et al. [30] extracted the deep visual features for food detection and recognition. These approaches only used deep visual features, but ignored the context. Some works [5, 7, 28, 47, 61] developed context-based recognition by introducing additional information, such as GPS, restaurant menus and ingredients. For example, Xu et al. [61] explored the geolocation and external information about restaurants to simplify the classification problem. Those works can confirm that the contextual knowledge is crucial to improve recognition. Recently, Min et al. [47] utilized ingredients as supervised signals to localize multiple regions with different scales and fused these regional features into the unified feature representation for recognition. There are also some works [49, 50], which developed mobile food
recognition systems for dietary management. In addition, our work is relevant to ingredient-recipe correlation learning and cross-modal retrieval [11, 39, 44, 46]. In contrast, we take food attributes into account for few-shot food recognition.

### 5.2 Few-Shot Learning

Few-shot learning has received more and more attention for recognizing novel visual categories from very few labeled examples. The seminal work [15] proposed a variational Bayesian framework for few-shot learning, which utilized previous learned classes to predict new ones when only one or very few examples are available. A hierarchical Bayesian program learning method [36] was later proposed to match the human level error on the few-shot alphabet recognition task. Recent works have adopted different strategies to deal with the few-shot problem [6, 17, 34, 48, 53, 56, 57, 60], which can be summarized into two main kinds of approaches. The first one is meta learning [3, 59], which tries to extract some transformable information to avoid overfitting in the few-shot learning stage. Different mechanisms, such as attention mechanism [19], memory mechanism [10] and category-agnostic activations-parameters mapping learning mechanism [51], are introduced to improve the performance of few-shot learning. The second type is metric learning based methods [34, 57, 60], which aim to learn a set of projection functions such that when represented in this embedding, images are easy to recognize using simple nearest neighbor or linear classifiers. In addition, some works [48, 53] leverage recurrent neural networks with memories to solve the few-shot learning problem.

Recently, there are metric learning based methods for few-shot learning [34, 57, 60]. For example, Koch et al. [34] employed the Siamese Networks as the embedding networks, and focused on learning the embedding to transform the data such that it can be recognized with a fixed nearest-neighbor. Later, Vinyals et al. [60] proposed the Matching Network, which transformed the support set and query samples into a shared embedding space such that it can be recognized with a fixed classifier. More recently, Sung et al. [57] proposed a model called Relation Network, which uses convolutional neural networks as a nonlinear classifier, and needn’t manually choose the metric, such as the cosine or Euclidean distance [19] to adapt the model or data. Furthermore, through learning a nonlinear similarity metric jointly with the embedding, this model achieves a great performance on miniImageNet [60] and Omniglot [35]. Our work belongs to metric learning based method for few-shot learning. However, all of those works focus on using category information to solve the few-shot learning problem. In this paper, we consider the problem of few-shot learning for food recognition and enhance the few-shot food recognition by leveraging ingredient attributes. In addition, few-shot learning is relevant to weakly supervised learning with small subset of training data, such as object detection and classification in optical remote sensing images [20, 21, 64]. This is because they should cope with the problem of very small subset of training data. Note that our proposed multi-view learning method focuses on learning different types of features from one image, not multiple images rendered from a 3D shape [22–24].

### 6 CONCLUSIONS

In this paper, we have proposed a Multi-View Few-Shot Learning (MVFSL) framework to explore ingredient information for few-shot food recognition. In order to take advantage of ingredient information, these two kinds of features are effectively by first combining their extracted feature maps from the last convolution layer of their respective fine-tuned deep networks, and then conducting the convolution on the combined feature maps. In addition, this convolution is incorporated into a multi-view relation network, which is used to compare query images against labeled samples to obtain the image-level relation score. MVFSL can be trained in an end-to-end way to enable joint optimization. The comprehensive experimental evaluations on three different food datasets have
validated the effectiveness of MVFSL. In addition, we have extended another two types of few-shot learning methods, namely Siamese Network and Matching Network by introducing ingredient information. The experimental results on these food datasets have also demonstrated the advantage in utilizing food ingredients for few-shot food recognition.

There are a number of issues for further study: (1) Exploring more information from food dataset to improve the performance of few-shot food recognition. For example, besides ingredient information, cooking instructions [52] and other types of attribute information, such as regional attributes and cuisine types [44] can also be utilized. (2) In our work, we have found that the difference between the distribution of training set and test set affects the performance of few-shot learning. Therefore, how to deal with this difference is worth studying. (3) We plan to use other types of advanced deep architectures such as ResNet [25] and Densenet [27] in our framework to continue improving the performance. In addition, different feature fusion strategies can also be explored, such as summation and max pooling.

ACKNOWLEDGMENTS
This work was supported in part by the National Natural Science Foundation of China under Grant 61532018, 61972378 and U19B2040, in part by Beijing Natural Science Foundation under Grant L182054, in part by National Program for Special Support of Eminent Professionals and National Program for Support of Top-notch Young Professionals.

REFERENCES
Few-Shot Food Recognition via Multi-View Representation Learning


