Few-shot and Many-shot Fusion Learning in Mobile Visual Food Recognition

Heng Zhao[1], Kim-Hui Yap[1], Alex C. Kot[1], Lingyu Duan[2], Ngai-Man Cheung[3]
[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798
[2]School of Electronics Engineering and Computer Science, Peking University, China 100080
[3]Singapore University of Technology & Design, Singapore 487372

***Abstract*—Mobile visual food recognition is emerging as an important application in food logging and dietary monitoring in recent years. Existing food recognition methods use conventional many-shot learning to train a large backbone network, which refers to the use of sufficient number of training data to train the network. However, these methods firstly do not consider the cases where certain food categories have limited training data. Therefore, they cannot use the conventional training using many-shot learning. Further, existing solutions focus on improving the food recognition performance by implementing state-of-the-art large full networks, and do not pay much attention to reduce the size and computational cost of the network. As a result, they are not amenable for deployment on mobile devices. In this paper, we address these issues by proposing a new few-shot and many-shot fusion learning for mobile visual food recognition, it has a compact framework and is able to learn from existing dataset categories, and also new food categories given only a few sample images. We construct a new Indian food dataset called NTU-IndianFood107 in order to evaluate the performance of the proposed method. The dataset has two parts: (i) a Base Dataset of 83 classes of Indian food images with over 600 images per class to perform many-shot learning, and (ii) a Food Diary of 24 classes captured in restaurants with limited number to simulate the few-shot learning on new food categories. The proposed fusion method achieves a Top-1 classification accuracy of 72.0% on the new dataset.**

***Index Terms*—Food recognition, Few-shot learning, Fusion learning, Compact network**

I. Introduction

Daily food intake greatly impacts on our health. Hence monitoring dietary behavior can help cultivate a healthy lifestyle and evade many illnesses, such as cardiovascular dieases, diabetes and obesity. With the recent progress in e-health, various fitness applications have been developed on mobile devices, which help users maintain healthy dietary habits. MyFitnessPal [1] and Noom Coach [2] are two popular such applications. However, these applications require users to manually log their diets, which is time consuming and tedious. As opposed to manual data entry, some researchers have proposed to log the diets by capturing the food images using smart phone cameras. Therefore, it can provide faster feedback to users via image recognition.

Deep convolutional neural networks (DCNNs) are currently the state-of-the-art technique in image recognition. Compared to traditional hand-crafted features based approaches, such as SIFTs, HOG, DCNNs are capable of constructing better image feature representations, which is based on learnt knowledge of different datasets. In recent years, a number of DCNN architectures have been explored and applied in food recognition. Pouladzadeh et al. [3] proposed a food recognition system based on extracted CNN features, it can recognize multiple food in image by region mining. Hassannejad et al. [4] modified Google's Inception module to evaluate its performance across different food image datasets. Aguilar et al. [5] developed a fusion classifer which leverages on two different DCNN architectures to provide food recognition.

Most existing DCNN-based food recognition systems use large full networks, such as VGGNet [6], ResNet [7] or Inception [8] frameworks to provide satisfactory performance. However, the conventional training procedure, or many-shot learning requires sufficient image data, in order to train the large network adequately. Unfortunately, sufficient training images are not always available for some food categories. Therefore, food categories with limited image data are not able to support many-shot learning. Further, existing frameworks have high memory, computational and energy footprints. Hence, most of them are not amenable for mobile implementation.

Few-shot learning, which refers to the practice of learning a model with a very small amount of training data. Contrary to conventional many-shot learning, few-shot learning is designed to provide generalization or classification of image categories given only a few sample images. It is effective in solving recognition problems where there is scarcity of supervised data. In recent years, more researchers have paid attention to developing new few-shot learning framework. Snell Jake et al. [9] proposed a prototypical network to generate a centroid for each few-shot category based on averaging of embedded input image feature vectors. Classification of new images is calculated by distance measurement to different class centroids. Sung et al. [10] presented a relation network, which generates representations of the query and training images through an embedding module. Then the embedded feature vector of the query image is compared with the averaged feature vector of each category. Gidaris et al. [11] proposed a novel attention-based few-shot learning framework, which redesigns the classifier of a simple ConvNet to provide classification. In this paper, we address these issues by proposing a few-shot and many-shot fusion learning framework on food recognition, the method has the following features: (i) it is able to recognize a new food category that has not been seen during previous training, by using only a few sample images

978-1-7281-0397-6/19/$31.00 ©2019 IEEE

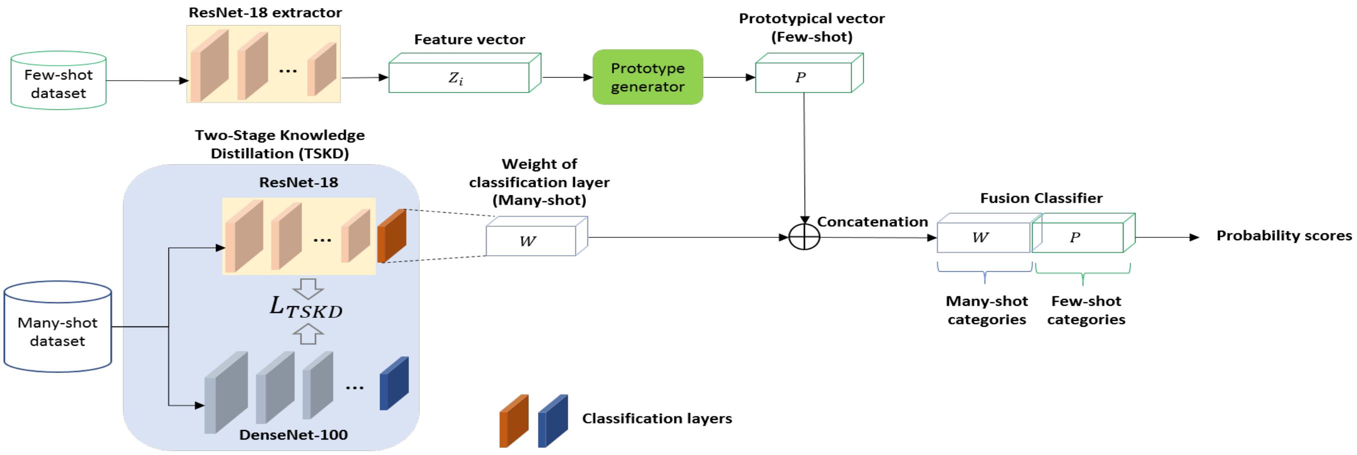

Fig. 1. Overview of the proposed few-shot and many-shot fusion learning. It consists of: (a) a knowledge distilled ResNet-18 as many-shot recognition model, where the weight of the classification layer is extracted after training using may-shot dataset and (b) a few-shot framework using the ResNet-18 as feature extractor to obtain image features, and generate prototypical vector for each few-shot category using a few training images. The fusion classifier provides a unified recognition of both many-shot and few-shot images by concatenating the classification weight vector of many-shot and prototypical vector of few-shot.

and labels provided by users; (ii) it develops a many-shot and few-shot fusion learning so that it can learn both the food categories used in many-shot and new categories of few-shot learning; (iii) it has a compact network architecture which can be deployed on mobile devices with less storage and computation requirements. Section 2 gives an overview of the proposed method and details of methodology. Classification performance of the proposed method is discussed in Section 3 and Section 4 concludes the paper.

II. Proposed few-shot and many-shot fusion learning

A. *NTU-IndianFood107 Dataset*

We construct an Indian cuisine dataset, which consists of two components: (a) a base/many-shot dataset, which has 83 food categories crawled from search engines, Each category has around 600 images. Sufficient amount of training images support the conventional many-shot training. (b) A Food Diary/few-shot dataset, that consists of 24 food categories that are different from those in the base dataset. Food Diary is collected from a Indian food review website [12], where images are users' captured photos in restaurants. The purpose of building a Food Diary is to simulate the scenario of users' provided new category with limited number of images. Food Diary is used in few-shot learning.

In order to perform a fusion learning of both many-shot categories and new categories of few-shot, the proposed system should be able to recognize new categories using few-shot learning, while at the same time, recognizing the many-shot categories. An overview of our proposed framework is provided in Fig. 1.

B. *Many-shot Learning using Two-Stage Knowledge Distillation*

A compact network is designed in this paper to have shallower layers and less parameters compared to a full size DCNN. The purpose of knowledge distillation is to enhance the performance of the compact network (ϕ_s) by transferring information from a larger teacher network (ϕ_t). ϕ_t is first trained using a classical softmax loss to achieve high classification accuracy, then ϕ_s is trained such that its output closely matches that of ϕ_t. Hinton et al. [13] proposed to modify a parameter called Temperature (T) in the softmax function to generate so-called "soft target labels" from ϕ_t. These "soft target labels" are used to train ϕ_s to generate similar output.

A generalized softmax layer produces the proability distribution $\boldsymbol{q_i}$ for each input image x_i as follows:

$$M_T(\boldsymbol{a_i}) = \boldsymbol{q_i}, \text{where } q_i^j = \frac{exp(\frac{a_i^j}{T})}{\sum_k exp(\frac{a_i^k}{T})}, \tag{1}$$

where T is the temperature parameter, and $\boldsymbol{a_i}$ denotes the pre-softmax logits of image i generated by ϕ_t, where its dimension is the number of classes in the dataset. For conventional softmax classification, $T = 1$. Similarly, logits from ϕ_s is denoted as $\boldsymbol{b_i}$. A "softer" probability distribution $\boldsymbol{a_i'}$ by ϕ_t can be obtained by increasing the value of T. Therefore, the compact network ϕ_s can be trained in a supervised manner given the image label $\{(x_i, \boldsymbol{l_i})\}$. The loss function is defined as a combination of softmax loss and Kullback Leibler divergence loss between $\boldsymbol{a_i'}$ of ϕ_t and $\boldsymbol{b_i}$ of ϕ_s. The KL divergence loss is defined as:

$$L_{KD}(\phi_t, \phi_s) = \frac{1}{N_c}\sum_{i=1}^{N_c} KL(M_T(\boldsymbol{a_i'})||M_T(\boldsymbol{b_i})), \tag{2}$$

where N_c is the total number of training images in many-shot dataset. The softmax loss using cross-entropy is given as:

$$L_S(\phi_{softmax}) = \frac{1}{N_c}\sum_{i=1}^{N} \mathcal{H}(\boldsymbol{l_i}, M_{T=1}(\boldsymbol{b_i})), \tag{3}$$

Conventional knowledge distillation only leverages on the logits from the last FC layer. In order to capture better structural details of each input image, we also generate logits from intermediate layer conv4 and dense block 3 in both networks, where the intermediate layers produce larger feature maps. By integrating new FC and ReLU layers to respective intermediate layer in both networks. The intermediate logits can be denoted as $\boldsymbol{a_i^m}$ as for ϕ_t and $\boldsymbol{b_i^m}$ as for ϕ_s. During the training process, the intermediate logits and the pre-softmax logits both contribute to the Two-Stage Knowledge Distillation (TSKD) loss:

$$\begin{aligned} L_{TSKD}(\phi_t, \phi_s) = & \frac{1}{N_c}\sum_{i=1}^{N_c} KL(M_T(\boldsymbol{a_i^m})||M_T(\boldsymbol{b_i^m})) \\ & + \frac{1}{N_c}\sum_{i=1}^{N_c} KL(M_T(\boldsymbol{a_i})||M_T(\boldsymbol{b_i})) \end{aligned} \quad (4)$$

The total loss function is a weighted combination of softmax loss function and TSKD loss:

$$L(\phi_t, \phi_s) = \eta L_S(\phi_s) + (1-\eta)L_{TSKD}(\phi_t, \phi_s), \quad (5)$$

where, η defines the weight assigned to the softmax loss $L_S(\phi_s)$ of the compact network ϕ_s, which is combined with the TSKD loss between the two networks L_{TSKD}. For every input many-shot training image x_i, the compact network ϕ_s will extract a D-dimension feature vector z_i, and the raw classification score s_k before the softmax for k-th category:

$$s_k = z_i^T w_k, \quad (6)$$

where w_k denotes the classification weight vector for the k-th category in many-shot dataset. The overall matrix of the classification weight vectors $W = \{w_1, ..., w_k\}$ has a dimension of $D \times K$ with each column representing the classification weight vector of each category. W is updated using the loss function in Eq. 5.

C. Few-shot Learning

Common techniques in many-shot learning use SGD to update the weights in the last classification layer slowly. However, few-shot learning cannot leverage on the conventional network training procedure that uses a large amount of training images. Therefore, our few-shot recognition aims to create a prototype centroid for each few-shot category by utilizing the feature vectors of training images. Raw classification score of image is obtained by measuring the similarity between the image and the protoyipical vector of each category. However, the magnitude of these prototypical vectors depends on the input image feature vectors, the raw classification scores of many-shot and that of few-shot are also different in magnitude. Therefore, it is difficult to build a reliable unified recognition between many-shot and few-shot frameworks. Gidaris et al. [11] propose to use cosine similarity to calculate s_k instead of standard dot-product by normalizing the magnitude of the prototypical vector.

$$s_k = \alpha\cdot \overline{z_i}^T \overline{w}_k, \quad (7)$$

where $\overline{z_i}$ and $\overline{w}_k$ are the l_2-normalized vectors. α is a learnable scalar value to adjust the range of cosine similarity to fit the softmax function. As a result, the classification weights of few-shot learning are no longer affected by the magnitude of image features. Further, by removing the ReLU layer, prototypical vector can also take both positive and negative values same as the classification weight vector.

The feature vector of each training image in k'-th category can be denoted as $\{\overline{z_1}, ...\overline{z_n}\}$. The prototypical vector of the k'-th category is formulated as:

$$p_{(k', avg)} = \frac{1}{N'}\sum_{i=1}^{N'} \overline{z_i}, \quad (8)$$

Similar approach has been used in [9] and it gives good results. However, such prototypical vector based on averaging input image feature vectors may not be able to generate accurate representation of that category due to limited number of training images. To handle that, we introduce some enhancements by leveraging on the trained many-shot compact network. For the k'-th few-shot category, the prototypical vector is modified by referring to the many-shot classification weight vector w_k for each many-shot category. Since w_k after normalization represents the feature vector of its category, w_k also incorporates visual similarity. Therefore, the prototypical vector of a few-shot category can be updated by finding the most similar many-shot classification weight vectors.

$$v_{(k', sim)} = \frac{1}{N'}\sum_{i=1}^{N'}\sum_{k=1}^{K} f(\overline{z_i}, w_k)\cdot w_k \quad (9)$$

where the function $f()$ defines a cosine similarity function and followed by a softmax probability distribution to find how similar $\overline{z_i}$ is to each classification weight vector w_k, the output is a weighted combination of the most similar many-shot classification weight vectors. The final prototypical vector of the k'-th few-shot category is a weighted sum of the averaging based vector and the similarity based vector:

$$p'_k = \theta_{avg} \circ p_{(k', avg)} + \theta_{sim} \circ v_{(k', sim)}, \quad (10)$$

where $\circ$ is the Hamadard product. θ_{avg} and θ_{sim} define the trainable weighting vector with dimension of D. The value is updated during the fusion learning.

D. Fusion Learning

Let $P = \{p_1, ..., p'_k\}$ represents the prototypical vectors for all K' few-shot categories. It then can be concatenated with the many-shot classification weight vectors $W = w_1, ..., w_k$. A fusion classifier with classification weight $W_{fusion} = [W, P] = [w_1, ..., w_k, p_1, ..., p'_k]$ is built to perform unified food recognition.

During the training stage, we first train the many-shot model using TSKD on the base dataset. In the next step, we remove the last classification layer and fix the parameters of the network to use as the feature extractor in few-shot learning (Fig. 1). During this stage, we train the learnable parameters of the few-shot prototypical vector θ_{avg} and θ_{sim} while continue to train the many-shot recognition. We randomly select N training images from the base dataset and N' training images from the Food Diary per category to train the network during each training episode. The prototypical vector of each few-shot category is generated using the selected N' training images. The total loss function of the fusion learning is a cross-entropy loss defined based on the negative log-probability $loss(x, y)$ of both many-shot and few-shot categories:

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N} loss(x_{(i,k)},k)+\frac{1}{K'}\sum_{k'=1}^{K'}\frac{1}{N'}\sum_{i=1}^{N'} loss(x'_{(i,k')},k') \tag{11}$$

where K and K' denote number of categories in the many-shot and few-shot datasets, respectively. $x_{(i,k)}$ denotes image i in the category k.

After the training is completed, the classifier is finetuned based on the classification weight W_{fusion}. During test phase, we extract the feature vector of the query image, and calculate its raw classification score (Eq. 7). The prediction result is obtained after passing through the final softmax function.

III. Experimental results

The proposed method aims to address the issue in mobile food recognition where the network is trained using large dataset, but the new food categories provided by users have limited image number. Therefore the proposed fusion method leverages on both many-shot and few-shot learning to provide classification on both scenarios. We evaluate the performance of the proposed fusion learning with respect to both many-shot and few-shot recognition using NTU-IndianFood107. We applied a 5-way-5-shot few-shot training procedure.

A. Evaluation of Many-shot Recognition

We evaluate our proposed fusion learning on the base/many-shot dataset after training the system. The base dataset consists of 83 different Indian food categories, with around 600 images per category. 30% images of each categories are used for testing. Table I summarizes the Top-1 and Top-5 recognition accuracy with comparison to [11].

TABLE I
Table 1: Recognition performance of proposed method on many-shot images

Performance of many-shot recognition		
	Top-1(%)	**Top-5(%)**
Dynamic few-shot [11]	73.0	84.2
Proposed method	**74.8**	**85.7**

All experiments are carried out under the same settings. As shown in Table I, the proposed method is able to achieve a Top-1 accuracy of 74.8% when test on the many-shot test images. Compared to [11], our method obtains a better accuracy since the proposed TSKD provides better classification generalization.

B. Evaluation of Few-shot Recognition

We also evaluate our proposed fusion learning method on the Food Diary/few-shot dataset alone. Food Diary has 24 Indian food categories, all of them are not included in the base dataset. Each of the 24 categories is split into 70% training and 30% testing. The classification accuracy on the testing set of Food Diary is shown in Table II:

TABLE II
Table 2: Recognition performance of proposed method on few-shot images

Performance of few-shot recognition		
	Top-1(%)	**Top-5(%)**
Dynamic few-shot [11]	67.6	78.3
Proposed method	**70.1**	**81.5**

For the recognition performance on Food Diary, our proposed method achieves a Top-1 accuracy of 70.1%, which outperforms [11] by 2.5% under the same experimental settings. Results indicate that the proposed method is able to extract accurate feature vectors of few-shot images to provide a better classification performance.

C. Evaluation of Fusion Recognition

In this section, we demonstrate the performance of our proposed method when handling images from both the many-shot and few-shot categories, where images from both base dataset and Food Diary are randomly selected with equal number during testing. The overall recognition accuracy on NTU-IndianFood107 is shown in Table III:

TABLE III
Table 3: Recognition performance of proposed method on both many-shot and few-shot images

Performance of fusion recognition		
	Top-1(%)	**Top-5(%)**
Dynamic few-shot [11]	69.2	80.9
Proposed method	**72.0**	**83.4**

Results show that our proposed method can achieve a Top-1 accuracy of 72.0% across all 107 food categories. Compared to the dynamic few-shot method, we achieve a 2.8% improvement. Results demonstrate the effectiveness of implementing the fusion learning to unify few-shot and many-shot training process.

IV. Conclusion

In this paper, we propose a fusion learning of few-shot and many-shot for mobile visual food recognition. It is able to learn a new food category using the few-shot framework, and also recognize the categories of many-shot training. We evaluate the performance of the proposed method on a new Indian food dataset called NTU-IndianFood107, where we demonstrate the effectiveness of the proposed fusion learning on handling both few-shot and many-shot food images at the same time.

Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under the NRF-NSFC grant NRF2016NRF-NSFC001-098. The research work was done at the Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore.

References

[1] MyFitnessPal, "Free calorie counter, diet and exercise journal," 2018. www.myfitnesspal.com.

[2] NoomCoach, "Stop dieting. get life-long results," 2018. https://www.noom.com/.

[3] P. Pouladzadeh and S. Shirmohammadi, "Mobile multi-food recognition using deep learning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 3s, p. 36, 2017.

[4] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, "Food image recognition using very deep convolutional networks," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, pp. 41–49, ACM, 2016.

[5] E. Aguilar, M. Bolaños, and P. Radeva, "Food recognition using fusion of classifiers based on cnns," in *International Conference on Image Analysis and Processing*, pp. 213–224, Springer, 2017.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[9] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[10] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

[11] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

[12] Zomato, "Indian restaurant search and discovery service." www.zomato.com.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.