JDNet: A Joint-learning Distilled Network for Mobile Visual Food Recognition

Heng Zhao, Kim-Hui Yap, *Member, IEEE,* Chichung Kot, Alex, *Fellow, IEEE* and Lingyu Duan, *Member, IEEE*

***Abstract*—Visual food recognition on mobile devices has attracted increasing attention in recent years due to its roles in individual diet monitoring and social health management and analysis. Existing visual food recognition approaches usually use large server-based networks to achieve high accuracy. However, these networks are not compact enough to be deployed on mobile devices. Even though some compact architectures have been proposed, most of them are unable to obtain the performance of full-size networks. In view of this, this paper proposes a Joint-learning Distilled Network (JDNet) that targets to achieve a high food recognition accuracy of a compact student network by learning from a large teacher network, while retaining a compact network size. Compared to the conventional one-directional knowledge distillation methods, the proposed JDNet has a novel joint-learning framework where the large teacher network and the small student network are trained simultaneously, by leveraging on different intermediate layer features in both network. JDNet introduces a new Multi-Stage Knowledge Distillation (MSKD) for simultaneous student-teacher training at different levels of abstraction. A new Instance Activation Learning (IAL) is also proposed to jointly train student and teacher on instance activation map of each training sample. Experimental results show that the trained student model is able to achieve a state-of-the-art Top-1 recognition accuracy on the benchmark UECFood-256 and Food-101 datasets at 84.0% and 91.2%, respectively, and retaining a 4x smaller network size for mobile deployment.**

***Index Terms*—Mobile food recognition, Compact network, Network optimization, Knowledge distillation**

I. Introduction

Food recognition has attracted much attention among researchers, considering its importance in improving people's nutrition balance and promoting healthy dietary behavior. A careful management of daily food intake is beneficial to not only individual's personal health but also the general wellness of societies at large. With the rapid development of mobile phones, it allows people to manage their food intake in a more convenient and user-friendly way. Various health promotion and intake tracking apps have been developed. A few apps such as MyFitnessPal [1] and LoseIt [2] allow users to log their diets manually. Manual data entry is tedious and time-consuming. Study shows that such applications are difficult to retain their users in the long term [3]. Smart phones nowadays equip with high-definition digital cameras, meal logging and analysis using captured food photos becomes much more convenient. Some apps are available which either just log the meal without any diet analysis or rely on expert nutritionists [4] or crowd sourcing [5] to analyze the images offline. One critical issue with these approaches is that they are unable to provide any immediate information or feedback to the users about their eatings. Since the analysis is done offline, the feedback is delayed and the effectiveness is reduced.

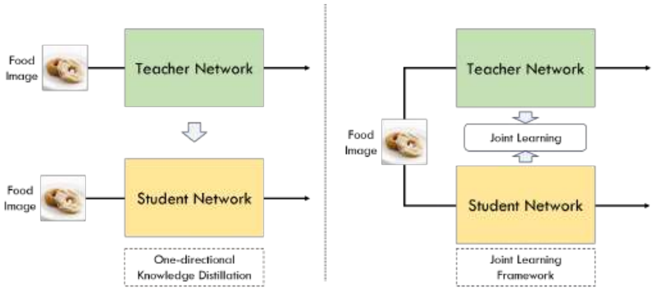

Fig. 1. Comparison with traditional one-directional knowledge distillation. Traditional methods usually leverage on single source of features or information and enhance the performance of the student via a two-step manner by training the teacher first and then distilling the knowledge to the student. In contrast, our proposed joint-learning framework is able to learn from multiple features and train the teacher and student simultaneously.

Visual food recognition is an emerging research field in computer vision. Early solutions focused on image-based visual features and traditional classification methods. Hand-crafted global and local features, SIFT, Local Binary Patterns (LBP) [6] and some contextual information such as where the image was captured have been evaluated for dish recognition [7]. K-Nearest Neighbor (k-NN) [8], Support Vector Machine (SVM) [6], artificial neural networks, and random forest classification methods [9] are amongst the widely used classifiers in the context of food classification. Recent research on deep convlutional networks has shown that such deep architectures can outperform traditional hand-crafted features based methods for food recognition problems in general [10], [11], [12]. Most existing methods use standard convolutional neural network (CNN) architectures and employ deep features extracted directly from the neural networks for image-level food classification [13], [14], [15]. Various studies [16], [15] show that deeper networks such as VGGNet [17], GoogleNet [18] and ResNet [19] are good models for generating food features.

A common deep learning based mobile visual food recognition solution adapts a client-server protocol. The client device snaps the picture of the food and sends it to a central server possibly with some contextual information such as geo-location. The classification happens at the server that runs a deep classification model and the results are sent back to the client. A key limitation of such solutions is that the classification engine at the server is typically based on large architectures that are both computation and memory intensive. Such models are too big to be deployed on client devices. Therefore, the users have to send each food picture to the server for analysis and feedback. This may not be a practical approach since the end-users may not have access to the Internet at all meal times or may not be willing to transfer pictures over the network due to privacy concerns. As a result, it is beneficial to developing a compact network that can be deployed on mobile devices that can provide direct recognition without reliance on the central server with satisfactory recognition performance.

In view of this, we propose a Joint-learning Distilled Network (JDNet) for mobile visual food recognition that aims to achieve high food recognition accuracy, while also retaining small storage footprint. As opposed to traditional one-directional knowledge distillation from a large teacher network to a compact student network, the proposed JDNet uses a novel joint-learning framework where both the teacher and student networks are jointly trained using a loss function consists of parameters from both networks. A new Multi-Stage Knowledge Distillation (MSKD) method is developed to leverage on network features at multiple levels of abstraction. In addition, a new technique called Instance Activation Learning (IAL) is proposed to ensure the consistency of the joint-learning via the Instance Activation Maps (IAMs) of each training image in both teacher and student networks.

The key contribution of this work is the joint-learning framework and simultaneous training of the teacher and student networks using novel Multi-Stage Knowledge Distillation (MSKD) and Instance Activation Learning (IAL) techniques. The resulting student network outperforms state-of-the-art performance on UECFood-256 and Food-101 datasets at more than 4 times reduction in model size. Further, although the proposed JDNet is developed for food recognition, we believe it can also be extended to other application domains.

II. Related Works

A. Hand-crafted features based methods

Early visual food recognition solutions use hand-crafted features and vision-based methods. Techniques such as SVM classifiers with color and HOG feature vectors [20], random forest and SVM classification [13] have been investigated. In [21], a Supervised Extreme Learning Committee (SELC), which is based on structural SVM, was used to find the optimal features of food images. These methods work well in conditions where image background is clean and the food items are well positioned.

B. Deep learning based methods

Recent visual food recognition research has shown that convolutional neural network (CNN) based deep architectures can outperform traditional hand-crafted features based methods [10], [11], [12]. Most existing methods deploy standard CNN architectures such as GoogLeNet, ResNet and use deep features extracted directly from the neural networks for image-level food classification [13], [14], [15]. Aguilar et al. evaluated fusion of classifiers for food recognition [22]. Martinel et al. [23] proposed to use wide-slice residual networks for food recognition by leveraging on the vertical traits in food. Some works have formulated dish recognition as a multi-task classification problem where dish recognition is attempted along with other attribute recognition such as ingredients, cuisine, course, etc. [24], [25]. Contextual information such as geographical location [26], textual information [27] and ingredients [24] are shown to improve the recognition accuracy. These attributes, however, may not be always available.

Existing research has focused on improving recognition accuracy using deep architectures and contextual information. Unfortunately, deep architectures are not amenable for deployment on mobile devices. Some works have proposed compact neural network architectures [28], [29] but performance of these architectures is significantly inferior to the deeper architectures.

C. Knowledge distillation

Knowledge distillation is a novel idea initially proposed by Hinton et al. [30] to improve the performance of deep learning models. In the process, a large and complex network or an ensemble model (teacher network) is trained to extract useful information or features from the given data that can produce better predictions, then a smaller network (student network) is trained with the assistance of the teacher model. This smaller network will be able to produce comparable results against the large teacher network. Recent development in knowledge distillation involves leveraging on different features or information extracted from the teacher network to enhance the performance the student network. Romero et al. [31] extended knowledge distillation to enhance the training of a deeper and thinner student with FitNet, which introduces a new loss function based on the matching of middle layers' weights from both networks. Zagoruyko et al. [32] proposed to force the student to mimic the attention map of the teacher network to improve the predictions. Yim et al. [33] defined the distilled knowledge to be the flow between layers and teach the student network to minimize the distance loss between the flow of solution procedure (FSP) matrices. Zhang et al. [34] designed a mutual learning structure that allows two student to learn from one another without any large teacher involved. These methods, however, either require the structure of the teacher and student networks (residual blocks in [32], [33]) to be similar, or rely on single source of simple network output or extracted features for knowledge distillation [31], [34]. As a result, such approaches restrict the choices and flexibility of the teacher and student networks, hence limit the efficiency of the transfer learning between the two networks.

This paper addresses these limitations of mobile visual food recognition by proposing JDNet, a jointly trained compact network that achieves high food recognition accuracy. In addition, the proposed JDNet has a small storage footprint and computation cost which is more amenable for future deployment on mobile devices.

III. The proposed JDNet

Fig. 2 provides an overview of the proposed JDNet framework. The framework consists of two networks, a large teacher network and a compact student network. The teacher network is based on a state-of-the-art large full network architecture with good performance such as VGGNet, ResNet, or

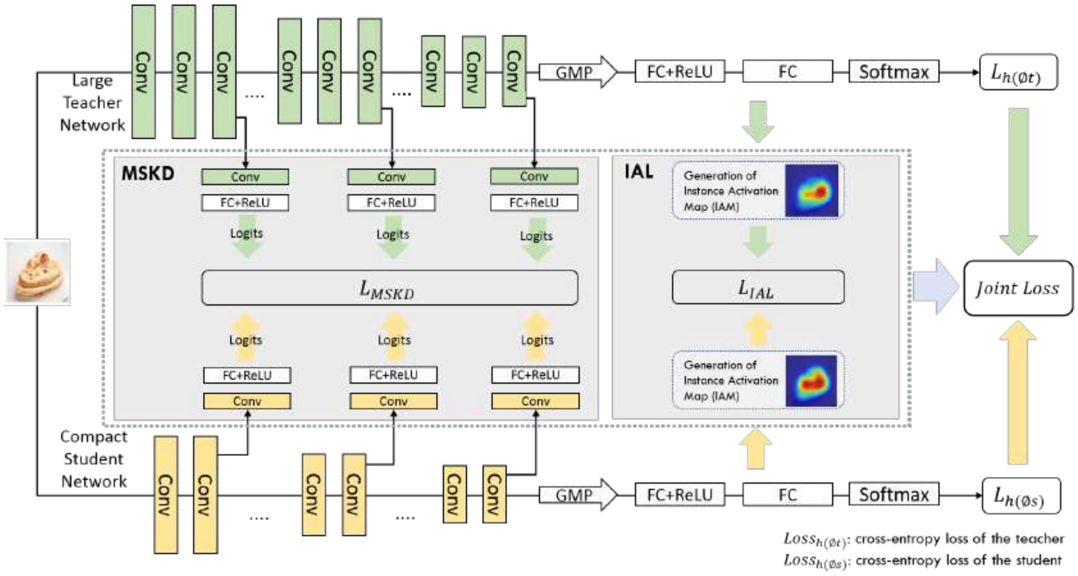

Fig. 2. Overview of the proposed JDNet. A full teacher network and a compact student network are jointly trained using the proposed Multi-Stage Knowledge Distillation (MSKD) and Instance Activation Learning (IAL) methods to achieve high recognition accuracy at low memory footprint.

DenseNet [35] which can better capture the features of images but cost a large storage. The student network has a light network structure and less storage requirement such as MobileNet [28] or SqueezeNet [29]. However, the performance of a standalone student network is usually lower than a state-of-the-art full network. In addition, the traditional one-directional knowledge distillation that used to transfer extra information from the teacher to the student has limited improvements on the compact network. In light of these drawbacks, we design a joint-learning framework which can train both the teacher and student models simultaneously by a joint loss function that consists of parameters from both networks. Two novel techniques are proposed under the framework to boost the performance of the compact network: Multi-Stage Knowledge Distillation (MSKD) which leverages on multiple levels of abstraction within both networks, and Instance Activation Learning (IAL) that is used to jointly train both networks on the instance level using the Instance Activation Maps (IAMs) generated for each input image.

The teacher and the student networks that are used for joint-learning, denoted as ϕ_t and ϕ_s respectively, are jointly trained for classification task on a common food dataset using classical softmax classification loss and two new loss functions defined for MSKD and IAL, shown in Fig. 2: L_{MSKD} and L_{IAL}. Total loss function of the overall JDNet framework is given by:

$$\begin{aligned} L_T(\phi_t, \phi_s) = L_h(\phi_t) + L_h(\phi_s) + \alpha L_{MSKD}(\phi_t, \phi_s) \\ + \beta L_{IAL}(\phi_t, \phi_s) \end{aligned} \tag{1}$$

where $L_h(\phi_t)$ is the cross-entropy loss between the predicted and ground-truth label distribution of ϕ_t and $L_h(\phi_s)$ is the cross-entropy loss of ϕ_s, L_{MSKD} is the MSKD loss and L_{IAL} is the IAL loss. The losses L_{MSKD} and L_{IAL} are weighted using factors α and β.

The MSKD loss captures Kullback-Leibler (KL) divergence between *logits* of ϕ_t and ϕ_s at multiple layers in the networks (Fig. 2). This enables joint-learning at multiple levels of abstraction between the two networks. The IAL loss enforces joint-learning based on Instance Activation Maps (IAMs) for each input image on instance level. By minimizing IAL, we make sure that the two networks focus on the same image parts while processing an input image.

A. Proposed Multi-Stage Knowledge Distillation (MSKD)

Conventional knowledge distillation transfers knowledge from a teacher network (ϕ_t), to the student network (ϕ_s), in a one-directional manner, which involves two steps. The ϕ_t is first trained using a classical cross-entropy softmax loss to achieve high classification accuracy. As a second step, ϕ_s is trained such that the output of ϕ_s closely matches that of ϕ_t. To achieve this, the class probabilities generated by ϕ_t in the first phase are used as the "soft target labels" while training the ϕ_s in the second phase. Hinton et al. [30] proposed to use a parameter called temperature (T) in the softmax function to generate these soft target labels from the teacher.

Given an image-label pair$\{x_i, l_i\}$ from the training set, where l_i belongs to one of the C classes in the dataset $l_i \in \{1, 2, ..., C\}$. A generalized softmax layer in ϕ_t produces a probability distribution for image x_i:

$$M_T(v_i) = f_{softmax}(v_i) = \frac{exp(\frac{v_i^c}{T})}{\sum_{j=1}^{C} exp(\frac{v_i^j}{T})}, \tag{2}$$

where $\boldsymbol{v}$ denotes the logits or log probability before normalization by the teacher network ϕ_t. T is the temperature parameter. For traditional classification task, $T = 1$. Similarly, the student network ϕ_s generates a student logits vector $\boldsymbol{w_i}$ and the corresponding probability distribution $M_T(\boldsymbol{w_i})$. By increasing the temperature T in ϕ_t, a "softer" probability distribution $\boldsymbol{z_i}$

over all C classes is obtained. Existing literature proposed to minimize the KL divergence between the soft probability distribution of the teacher and the normal probability distribution of the student [30]:

$$L_{KD}(\phi_t, \phi_s) = \frac{1}{N} \sum_{i=1}^{N} KL(M_T(\boldsymbol{z_i}) || M_T(\boldsymbol{w_i})), \tag{3}$$

where $KL(\boldsymbol{x}||\boldsymbol{y})$ represents the KL divergence between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ and N is the total number of training images. When a set of image-label pairs $\{(x_i, l_i)\}$ are given, the ϕ_s can be trained in a supervised manner using a weighted combination of cross-entropy and KL divergence losses:

$$L(\phi_t, \phi_s) = \eta L_h(\phi_s) + (1 - \eta) L_{KD}(\phi_t, \phi_s), \tag{4}$$

where η is the weighting factor. The cross-entropy loss $L_h(\phi_s)$ for the student network is defined as:

$$L_h(\phi_s) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(l_i, M_{T=1}(\boldsymbol{w_i})), \tag{5}$$

where $\mathcal{H}$ denotes the cross-entropy function. However, the traditional knowledge distillation only leverages on the pre-softmax logits, which usually corresponds to the layer at the very end of the network. Since the logits are generated based on small deeper layer features, the probability distribution in the logits v_i and w_i may not capture good enough structural information for each input image. Therefore, we propose to generate multiple logits from intermediate layers in both ϕ_t and ϕ_s. These intermediate convolutional layers offer multi-level abstraction of representation for input images, which can better capture the details of images using larger features. By incorporating logits from intermediate layers, both ϕ_t and ϕ_s are trained at the same time to perform a more effective joint knowledge distillation using different levels of abstraction in both networks.

To extract the logits from multiple intermediate convolutional layers in both networks, we introduce new fully-connected and ReLU layers to the intermediate convolutional layers in ϕ_t and ϕ_s (Fig. 2). For an input food image x_i, we can choose M intermediate layers in both ϕ_t and ϕ_s. The extracted logits vectors, $\boldsymbol{z_i^m}$ generated by ϕ_t and $\boldsymbol{w_i^m}$ generated by ϕ_s can be represented as $(\boldsymbol{Z_i}, \boldsymbol{W_i}) = ((\boldsymbol{z_i^1}, \boldsymbol{w_i^1}), (\boldsymbol{z_i^2}, \boldsymbol{w_i^2}), \dots (\boldsymbol{z_i^M}, \boldsymbol{w_i^M}))$.

During the training process, the logits from all selected intermediate layers contribute to the MSKD loss. The MSKD loss with the temperature parameter T can be defined as follows:

$$L_{MSKD}(\phi_t, \phi_s) = \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{M} KL(M_T(\boldsymbol{z_i^m}) || M_T(\boldsymbol{w_i^m})), \tag{6}$$

where M is the number of selected stages and N is the total number of training images. In practice, the value of M is set between 2-4. Detailed analysis on the effect of M is provided in the experiment section IV-D. Further, the student network ϕ_s can be trained using a weighted loss of cross-entropy and MSKD similar to Eq. 4:

$$L(\phi_t, \phi_s) = \eta L_h(\phi_s) + (1 - \eta) L_{MSKD}(\phi_t, \phi_s), \tag{7}$$

B. Proposed Instance Activation Learning (IAL)

Activation maps extracted from a CNN provide a good visualization on how the network model is able to classify the input image. Recent research has shown that activation maps output from the intermediate convolution and their corresponding activation layers can be used for significantly improving the performance of CNNs, such as classification of images [32] and object localization and detection [36], [37], [38].

Activation maps generated by different convolutional layers in the network have different visualization results. It is observed that layers that are deeper in the network focus more on training data specific features, while the earlier layers focus on general features or patterns, such as edges, texture, etc. Therefore, later stage activation maps play a crucial role in recognition of different categories.

The teacher and student networks in JDNet consist of a series of convolutional layers and their corresponding ReLU activations. For each input training image, we extract the activation maps A_i^t and A_i^s from both networks at their respective last convolutional layers. However, these raw activation maps may not be accurate enough to identify the discriminative parts of the image by using only image information.

We propose a new approach called Instance Activation Learning (IAL) to enhance the joint-learning between the teacher and student networks by incorporating the category/label information into the activation maps, to generate the Instance Activation Maps (IAMs) for each input image. To better leverage on the label information of input training images, we propose to use global max pooling (GMP) to pool the activation maps A_i^t and A_i^s to their following fully-connected layers in both networks, where the weights of the fully-connected layer contain a raw classification probability distribution for the given image. Therefore, we extract the weights from the fully-connected layer and project back to A_i^t and A_i^s.

For a C-class food classification problem, the training image set consists of image-label pairs $\{x_i, l_i\}$, where $l_i \in \{1 \dots C\}$. For normal softmax function, let $f_k(x, y)$ denotes the activation of the k-th unit in the last convolutional layer at the spatial location (x, y), the raw classification scores of image x_i before the softmax layer can be expressed as:

$$scores = \sum_k g_k^{l_i} F^k, where \quad F^k = \max_{(x,y)} f_k(x, y) \tag{8}$$

$g_k^{l_i}$ is the weight of unit k that corresponds to l_i, and F^k is the output of unit k after GMP.

To obtain the IAM from the network, we extract the weights from corresponding fully-connected layer. It is denoted as matrix $P \in \mathbb{R}^{\mathbb{D} \times \mathbb{C}}$, where each column $p^{l_i} \in \mathbb{R}^{\mathbb{D}}$ of P represents the overall weights for class $l_i \in C$. We define I_i as the IAM for image x_i:

$$I_i = \boldsymbol{p^{l_i}} \sum_k g_k^{l_i} f_k(x, y) \tag{9}$$

Fig. 3 shows some sample IAMs from both the teacher and student networks. IAMs is generated using label information of input image, which provides a better generalization over

Algorithm 1 Training loss computation of JDNet. N is the number of examples in the training set S_T. $\boldsymbol{v_i}$ and $\boldsymbol{w_i}$ are the logits from the teacher and the student networks respectively. $\boldsymbol{z_i}$ is the "softer" logits of teacher. M is the number of stages used in MSKD.

```
for image $x_i \in S_T$ do
    $L_h(\phi_t) \leftarrow (\mathcal{H}(l_i, M_{T=1}(\boldsymbol{v_i})))$    ▷ Cross-entropy loss of the teacher
    $L_h(\phi_s) \leftarrow (\mathcal{H}(l_i, M_{T=1}(\boldsymbol{w_i})))$    ▷ Cross-entropy loss of the student
    $I_i^t \leftarrow IAL(x_i)$    ▷ Instance activation map of Image data $x_i$ from the teacher
    $I_i^s \leftarrow IAL(x_i)$    ▷ Instance activation map of Image data $x_i$ from the student
    $L_{IAL}(\phi_t, \phi_s) \leftarrow ||I_i^t - I_i^s||_F$    ▷ IAL loss computation using Frobenius norm
    for $m \in \{1, 2, ..., M\}$ do
        $L_{MSKD}(\phi_t, \phi_s) \leftarrow \sum_{m=1}^{M} KL(M_T(\boldsymbol{z_i^m})||M_T(\boldsymbol{w_i^m}))$    ▷ MSKD loss across M stages
    end for
end for
$L_T(\phi_t, \phi_s) \leftarrow 0$    ▷ Initialize loss
for $i \in \{1, 2, ..., N\}$ do
    $L_T(\phi_t, \phi_s) \leftarrow \frac{1}{N} \sum_{i=1}^{N} (L_h(\phi_t) + L_h(\phi_s) + L_{MSKD}(\phi_t, \phi_s) + L_{IAL}(\phi_t, \phi_s))$    ▷ Update loss
end for
```

different classes. IAMs extracted from both the teacher and student networks are used to ensure the consistent learning of the activation between the two networks. We denote the instance activation map for an input image and label pair (x_i, l_i) generated by the teacher and the student networks as I_i^t and I_i^s, respectively. The IAL loss function is defined as:

$$L_{IAL}(\phi_t, \phi_s) = \frac{1}{N} \sum_{i=1}^{N} ||I_i^t - I_i^s||_F \tag{10}$$

where N is the total number of training images, and $||I_i^t - I_i^s||_F$ denotes the Frobenius norm between the IAMs $I_{l_i}^t$ and $I_{l_i}^s$ from the teacher and student models, respectively. Integrating the MSKD and the IAL loss functions, the joint loss function of the JDNet in 1 can be re-written as:

$$\begin{aligned}
&L_T(\phi_t, \phi_s) \\
&= L_h(\phi_t) + L_h(\phi_s) + \alpha L_{MSKD}(\phi_t, \phi_s) \\
&\quad + \beta L_{IAL}(\phi_t, \phi_s) \\
&= \frac{1}{N} \sum_{i=1}^{N} (\mathcal{H}(l_i, M_{T=1}(\boldsymbol{v_i})) + \mathcal{H}(l_i, M_{T=1}(\boldsymbol{w_i}))) \\
&\quad + \alpha \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{M} KL(M_T(\boldsymbol{z_i^m})||M_T(\boldsymbol{w_i^m})) \\
&\quad + \beta \frac{1}{N} \sum_{i=1}^{N} ||I_i^t - I_i^s||_F
\end{aligned} \tag{11}$$

Pseudocode to compute the total loss function L_T is given in Algorithm 1.

Experimental results show that JDNet is fairly insensitive to the weighting factors α and β that we use to control the loss weights, as a result, both α and β are set equal to give same level of importance on MSKD and IAL. To make the distillation process as effective as possible, more emphasis should be put on the MSKD loss and IAL loss while keeping the weighting of the cross-entropy losses small. Therefore, for a consistent experimental settings, we set the weights α and β to be 0.45.

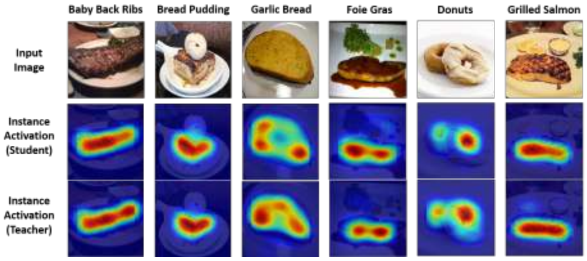

Fig. 3. Instance activation maps (IAMs) of some sample food images.

IV. Results

A. Datasets

To evaluate the performance of the proposed framework, we select two popular benchmark datasets.

UECFood-256. The UECFood-256 dataset [39] contains 256 food categories and up to around 32,000 images from different countries. Dishes from different cuisines, such as Japanese, Western, Chinese, etc. are included in the dataset. This dataset was constructed to implement a practical food recognition system which is based on Android smartphones, so it is suitable for benchmark testing.

Food-101. The Food-101 [13] dataset consists of 101 most popular Western dishes. 1,000 images are provided for each dish. This dataset is split into training set and testing set, which contains 75,750 images and 25,250 images, respectively. Food-101 is one of the most popular datasets for benchmark testing.

Fig. 4 shows some samples images selected from the benchmark datasets.

TABLE I
Stages used in MSKD.

	Layer (T)	**Layer (S)**	**Size of conv layer in MSKD**
m_1	Res5c	Block6_4	$7 \times 7 \times 1024$
m_2	Res4$fb22$	Block5_2/sep	$14 \times 14 \times 512$
m_3	Res3$b3$	Block4_6/sep	$28 \times 28 \times 256$
m_4	Res2c	Block3_1/sep	$56 \times 56 \times 128$

B. Experimental settings

We evaluate our proposed framework with ResNet-101 as the basic structure of the teacher network and MobileNet-v2 as the student network.

For the implementation of MSKD, we choose $M = 4$ pairs of intermediate layers that are used to calculate the logits from different levels of abstraction in both the teacher and student networks. Details of the selected layers are shown in Table I. In order to have consistent logits generation, the outputs at these intermediate layers are resized to have a same output dimension. Symbol *sep* represents the special separable convolutions used in MobileNet architecture [28]. Further, effect of using different number of stages for MSKD is investigated in Section IV-D.

We generate IAMs for each input image based on the last convolutional layers in both the teacher and student networks. IAMs from both networks are made to ensure the consistency of activation generation defined as IAL loss. After the training of the proposed framework, the JDNet student network ϕ_s is used for the following experiments. The performance evaluation is shown as Top-1 and Top-5 recognition accuracy. All experiments are carried out on NVIDIA Titan Xp and GTX 1080ti GPUs.

Image Data. For the two benchmark datasets, images are resized to 224 × 224. For UECFood-256 dataset, in order to have a fair comparison with other works, the results are based on the provided ground-truth cropped images. Since both UECFood-256 Food-101 provide training and testing split, we use the same partition in the following experiments.

Network Optimization. Both the teacher and student networks are trained at the same time. Training is performed via stochastic gradient descent (SGD) with batch size of 15 samples. The initial learning rate has been set to 0.001 and halved after every 10k iterations. Momentum is set to 0.9 and a weight decay set at 0.00001. Each training runs for 200k iterations.

C. Experimental results

In this section, we evaluate the JDNet student model, ϕ_s with existing works on UECFood-256 and Food-101.

UECFood-256. Table III summarizes the Top-1 and Top-5 recognition accuracy of the trained JDNet student network (ϕ_s), as compared to other food recognition methods using UECFood-256. Our proposed JDNet outperforms methods using traditional features [39], [14]. Compared to more recent works based on various deep learning frameworks, such as Inception [11], Fusion of ResNet and Inception [22], our compact student network (ϕ_s) achieves a better performance, which also surpasses the current state-of-the-art WISeR [23]. In addition, previous approaches heavily focus on improving the food recognition performance by exploring popular large architectures, such as variants of ResNet and Inception networks. Although large networks can provide good performance, it is difficult to be used on mobile devices, which limits their real-time application. In contrast, our JDNet offers a better performance at 84% and 96.2% for Top-1 and Top-5 accuracy respectively and still maintains a light network structure.

Table II summarizes the network architectures used in existing literature and their corresponding model size and computational complexity in terms of number of parameters and FLOPs. For example, WISeR [23] is based on a Wide ResNet and has a storage size around 260 MB, it contains 50.2 million of parameters which requires 9 GFLOPS for a single forward pass. Our proposed JDNet student network ϕ_s which is based on MobileNet-v2 structure, has a storage of 13.5 MB, it has 3.4 million of parameters which only needs 550 MFLOPs for a complete forward pass. Further, the performance of ϕ_s is greatly enhanced using the proposed joint-learning framework.

Food-101. Table IV summarizes the recognition performance of the trained JDNet student network (ϕ_s) and comparison with other food recognition methods using the benchmark Food-101. Compared to the first two methods which used traditional hand-crafted features, our proposed JDNet outperforms them significantly. Most of existing deep learning methods utilize large deep architectures to achieve high performance, notably WISeR [23], which is based on wide ResNet. Compared to these approaches, our proposed JDNet improve the performance of the student network to achieve better recognition accuracy at 91.2% and 98.8% for Top-1 and Top-5, respectively. In addition, ϕ_s retains a compact network structure.

D. Network analysis and ablation study

Performance comparison with other knowledge distillation approaches. One-directional knowledge distillation approaches require two stages to perform knowledge distillation: (i) training of the teacher network, followed by (ii) knowledge distillation from the teacher network to the student network. As opposed to them, our proposed JDNet trains both the teacher and student simultaneously by introducing a joint loss function, which consists of MSKD and IAL. Further, our framework doesn't require the network structure of both networks to be similar, which allows a more flexible knowledge distillation.

The comparison of the training complexity for the proposed method with other distillation approaches[30], [31], [32], [33], [34] is given in Table V. GFLOPs abd GPU memory consumption are used to estimate the training complexity. To have a fair comparison, the results are based on similar experimental configurations, conducted on Nvidia Titan Xp using Food-101 dataset. It is noted that GFLOPs is calculated based on one forward-pass batch (batch size of 15). Since the first four approaches are one-directional distillation, their training complexity consists of both training of the teacher network and training of the student network with additional complexity introduced by their respective techniques. Our FitNet implementation followed the idea of hint-based training using the teacher and student's middle layers as the hint and guided layers. We applied the training strategy of minimizing the distance loss between FSP matrices of the teacher and student networks for FSP approach. We used the same layers in I for AT and FSP. Two Mobilenet-v2 student networks are used for DML, best performance of the two students are chosen for comparison.

Fig. 4. 20 sample dishes selected from each of the benchmark datasets: UECFood-256 and Food-101. All images are collected from online search engines

TABLE II
COMPARISONS OF NETWORK ARCHITECTURE: MODEL SIZE, NUMBER OF PARAMETERS AND INFERENCE FLOPS.

Method	Architecture	Size (MB)	No. Params (Millions)	GFLOPs
DCNN-Food [14]	Modified AlexNet	425	111.3	1.3
DeepFood [15]	Modified GoogLeNet	51	7.0	3.2
Inception-V3 [11]	Inception-V3	96	23.7	6.0
Classifiers Fusion [22]	Inception-V3+ResNet-50	189	49.3	10.0
WISeR [23]	Modified Wide-ResNet	260	50.2	9.0
Proposed JDNet	**JDNet Student Network (ϕ_s)**	**13.5**	**3.4**	**0.55**

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY ON THE BENCHMARK UECFOOD-256 DATASET. FIRST 2 ROWS SHOW THE RESULTS BASED ON HAND-CRAFTED FEATURES, THE FOLLOWING 8 ROWS USE DEEP LEARNING BASED APPROACHES.

	Top-1(%)	**Top-5(%)**
FoodCam [39]	41.6	64.0
Color-FV+HOG-FV [14]	52.9	75.5
FAM [40]	63.2	85.6
DeepFoodCam [12]	63.8	85.8
DCNN-Food [14]	67.6	89.0
DeepFood [15]	63.8	87.2
Hassannejad et al. [11]	76.2	92.6
Classifiers Fusion [22]	76.7	–
ResNet-200 [19]	79.1	93.0
WISeR [23]	83.2	95.4
Proposed JDNet-ϕ_s	**84.0**	**96.2**

TABLE IV
COMPARISON OF CLASSIFICATION ACCURACY ON THE BENCHMARK FOOD-101 DATASET. FIRST 2 ROWS SHOW THE RESULTS BASED ON HAND-CRAFTED FEATURES, THE FOLLOWING 8 ROWS USE DEEP LEARNING BASED APPROACHES.

	Top-1(%)	**Top-5(%)**
RFDC [13]	50.8	–
SELC [21]	55.9	–
Bossard et al. [13]	56.4	–
DCNN-Food [14]	70.4	–
DeepFood [15]	77.4	93.7
FAM [40]	79.2	94.1
Classifiers Fusion [22]	83.8	–
Hassannejad et al. [11]	88.3	96.9
ResNet-200 [19]	88.4	97.85
WISeR [23]	90.3	98.7
Proposed JDNet-ϕ_s	**91.2**	**98.8**

Results in Table V show that, apart from DML, our proposed joint training framework has less total computation cost (GFLOPs) and achieves comparable memory consumption as compared with other distillation methods. DML uses two student networks to learn from one another, without involvement of large teacher network. As opposed to DML, we believe a large backbone teacher network is necessary to achieve efficient knowledge distillation. Results in Table VI show that our proposed method outperforms other distillation methods by $1.2\% - 6\%$ for Top-1 accuracy on both food datasets. With respect to other one-directional knowledge distillation approaches, our proposed JDNet provides information exchange between the teacher and student networks during the joint training, which leverages on different feature information, such as intermediate logits and Instance Activation Map (IAM). In addition, JDNet trains both networks simultaneously in a single stage, while one-directional knowledge distillation approaches require two stages to perform knowledge distillation. As compared to DML that relies on direct network output from two light student networks, our method achieves a better performance by leveraging on generated multiple intermediate logits and IAM from a large teacher network and a compact student network. Further, the training of the two student networks in DML are conducted iteratively. In contrast, our proposed JDNet trains the teacher and student networks simultaneously.

Performance comparison with standalone student and standalone teacher networks. To better demonstrate the effectiveness of our proposed JDNet, we also conducted experiments which compare the performance of the student ϕ_s

TABLE V
COMPARISON OF TRAINING COMPLEXITY OF JDNET FRAMEWORK WITH FIVE OTHER KNOWLEDGE DISTILLATION APPROACHES

Method	GFLOPs	Memory Consumption (GB)
KD [30]	136.5	8.7
FitNet [31]	139.1	10.3
AT [32]	136.5	9.6
FSP [33]	136.5	10.2
DML [34]	17.1	6.8
Proposed JDNet	**128.3**	**9.7**

TABLE VI
COMPARISON OF CLASSIFICATION ACCURACY OF JDNET-ϕ_s WITH FIVE OTHER KNOWLEDGE DISTILLATION APPROACHES ON UECFOOD-256 AND FOOD-101

Dataset	Method	**Top-**1(%)	**Top-**5(%)
UECFood-256	KD [30]	78.0	93.8
	FitNet [31]	81.9	94.0
	AT [32]	82.2	94.1
	FSP [33]	82.8	95.2
	DML [34]	81.3	93.7
	Proposed JDNet-ϕ_s	**84.0**	**96.2**
Food-101	KD [30]	84.6	96.4
	FitNet [31]	85.7	97.2
	AT [32]	86.0	97.2
	FSP [33]	87.6	97.9
	DML [34]	84.5	96.3
	Proposed JDNet-ϕ_s	**91.2**	**98.8**

of JDNet against a standalone MobileNet-v2, to verify the improvement in the student network after implementation of the joint-learning distillation framework. Results in Table VII shows that the proposed JDNet-ϕ_s outperforms the standalone MobileNet-v2 by 8.5% and 8.3% on UECFood-256 and Food-101, respectively. It is observed that the proposed JDNet student ϕ_s obtains significant performance boost using the proposed MSKD and IAL techniques.

TABLE VII
COMPARISON OF CLASSIFICATION ACCURACY OF JDNET-ϕ_s WITH A STANDALONE MOBILENET-V2 ON UECFOOD-256 AND FOOD-101

Dataset	Method	**Top-**1(%)	**Top-**5(%)
UECFood-256	MobileNet-v2	75.5	91.8
	Proposed JDNet-ϕ_s	**84.0**	**96.2**
Food-101	MobileNet-v2	82.9	95.3
	Proposed JDNet-ϕ_s	**91.2**	**98.8**

Performance comparison with different teacher and student networks. To verify the flexibility of the proposed method across different network combinations so that it can yield stable enhancement. We also tested scenarios that using networks such as ResNet-50, VGG-16 as the teacher and mobilenet-v1 as the student network. $M = 4$ intermediate layers are chosen to extract the multiple logits from different teacher-student combinations for MSKD. IAMs are generated based on last convolutional layer of each network. Experimental results on UECFood-256 and Food-101 are shown in Table VIII and IX. The results show the generalization ability of the proposed JDNet to enhance the performance of different student networks (MobileNet-v1 and v2) using different teacher networks, which leverages on the proposed novel knowledge distillation techniques. All six different combinations give better food recognition accuracy than its respective standalone student network. Further, combination using ResNet-101 as the teacher and MobileNet-v2 as the student gives the overall best performance on both food datasets compared to other network combinations.

TABLE VIII
COMPARISON OF CLASSIFICATION ACCURACY (TOP-1 (%)) OF JDNET-ϕ_s USING DIFFERENT TEACHER AND STUDENT NETWORKS ON UECFOOD-256 DATASET.

Teacher Student	VGG-16	ResNet-50	ResNet-101
MobileNet-v1	82.6	83.5	83.7
MobileNet-v2	83.0	83.6	**84.0**

TABLE IX
COMPARISON OF CLASSIFICATION ACCURACY (TOP-1 (%)) OF JDNET-ϕ_s USING DIFFERENT TEACHER AND STUDENT NETWORKS ON FOOD-101 DATASET.

Teacher Student	VGG-16	ResNet-50	ResNet-101
MobileNet-v1	88.4	90.4	90.6
MobileNet-v2	88.9	90.7	**91.2**

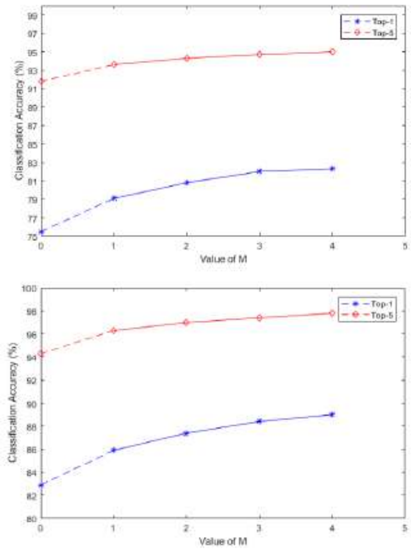

Fig. 5. Effect of multiple stages in MSKD on UECFood-256 and Food-101 datasets. Upper: UECFood-256. Lower: Food-101.

Performance analysis of MSKD and IAL in JDNet. In order to verify the respective contribution of each term in the joint loss to the final performance of the student network, we conducted additional experiments under following scenarios: JDNet with only MSKD (M=4), JDNet with only IAL and JDNet with both MSKD+IAL. Results using the same experimental settings are shown in Table X. Baseline of using standalone student network is also included for better comparison.

Table X shows the individual contribution of each term of the joint loss used in JDNet. The results show that both components of MSKD and IAL contribute to the final performance significantly. For example, by only using MSKD, the

TABLE X
ANALYSIS OF THE PERFORMANCE CONTRIBUTION OF MSKD AND IAL TO JDNET-ϕ_s

Dataset	Method	**Top-**1(%)	**Top-**5(%)
UECFood-256	Standalone student	75.5	91.8
	with **MSKD**	82.3	95.2
	JDNet with **IAL**	77.9	93.0
	with **MSKD+IAL**	**84.0**	**96.2**
Food-101	Standalone student	82.9	94.3
	with **MSKD**	88.7	97.8
	JDNet with **IAL**	84.9	96.6
	with **MSKD+IAL**	**91.2**	**98.8**

Top-1 performance of the trained JDNet is improved by 6.8% with respect to the standalone student network on UECFood-256. The performance is improved by 2.4% by only using IAL on UECFood-256. JDNet with both MSKD and IAL on the same dataset gives larger improvement of 8.5%. Similar performance improvements can be observed on Food-101. Detailed performance analysis on individual MSKD and IAL is conducted in the following sections.

Effect of number of stages in MSKD. MSKD leverages on multiple logits from intermediate layers at different abstraction. As have been shown in Table I, four stages are created which correspond to different intermediate layers in both teacher and student networks. For example, stage m_1 corresponds to the logits extracted from deeper layers from both networks, while stage m_4 represents logits from early layers. The corresponding improvements of using only MSKD in Table X (M=4) are 6.8% and 5.8% for Top-1 accuracy on UECFood-256 and Food-101, respectively. To further investigate the effect of MSKD on the proposed JDNet performance, we conducted experiments using MSKD only, by implementing different numbers of stages.

We started with $M = 0$, where MSKD is not used, then gradually added in MSKD stages from late layers to earlier layers in both networks. Fig. 5 shows that compared to a standalone MobileNet-v2 ($M = 0$), the performance increases by implementing MSKD with more stages. For example, when $M = 1$, where stage m_1 is used, the Top-1 accuracy on Food-101 is 85.8%, while the accuracy of a standalone MobileNet-v2 is 82.9%. Adding more stages in MSKD does improve the performance significantly in the beginning. However, as more stages are added, the new stages bring less effective improvements. Furthermore, since adding more stages requires additional computation, we define $M = 4$ for the implementation of MSKD in experiments.

Effect of IAL. IAL aims to enhance the network performance by forcing the consistent generation of IAMs from both the teacher and student networks. To verify its effectiveness in food recognition, we tested the proposed JDNet-ϕ_s on UECFood-256 and Food-101 with IAL only. Table X shows the performance of JDNet with only IAL, as compared to a standalone student network. Results show that student network achieves 2.4% improvement of Top-1 accuracy on UECFood-256 dataset by using only IAL, and 2% improvement on Food-101 by using only IAL.

To better showcase the impact of IAL on different categories of food images, Fig 6 shows the 10 most difficult and 10 most easy dishes to classify in Food-101. The blue bars show the Top-1 food recognition accuracy using MSKD+IAL, and the orange bars show the corresponding accuracy without IAL. Results show that by implementing IAL based on IAMs, the overall performance of the 10 most difficult and 10 most easy food categories are enhanced. The average improvement of the most easy categories is 1.8%, the maximum improvement is 4.1%. The average improvement of the most difficult categories is 1.0%, and the maximum improvement is 2.1%. Results in Table X and Fig 6 show that the proposed IAL can bring effective improvements to the student network.

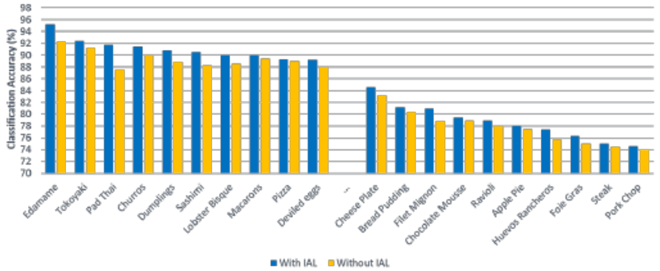

Fig. 6. Per category Top-1 recognition accuracy on the 10 easiest and 10 most difficult categories from Food-101. Blue bars represent the result of JDNet with MSKD+IAL, orange bars results are without IAL.

V. CONCLUSION

This paper proposes a Joint-learning Distilled Network (JDNet) for mobile visual food recognition. The proposed JDNet achieves high food recognition performance while retaining a compact network structure for mobile deployment. It leverages on a joint-learning of student and teacher networks which allows an effective knowledge transfer between two networks. Instead of the traditional one-directional knowledge distillation, JDNet introduces Multi-Stage Knowledge Distillation (MSKD) for a better distillation learning at multiple levels of abstraction. A new Instance Activation Learning (IAL) is proposed to jointly learn instance activation maps (IAMs) from both networks. The compact student network of JDNet outperforms previous state-of-the-art methods in benchmark testing, which demonstrates its effectiveness in food recognition.

ACKNOWLEDGEMENT

This research is partially 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 ROSE Lab under the NTU-PKU Joint Research Institute, a collaboration between Nanyang Technological University and Peking University that is sponsored by a donation from the Ng Teng Fong Charitable Foundation. This work is also partially supported by the National Natural Science Foundation of China under Grant 61661146005.

References

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

[2] LoseIt!, "Weight loss that fits," 2018, www.loseit.com.

[3] F. Cordeiro, D. A. Epstein, E. Thomaz, E. Bales, A. K. Jagannathan, G. D. Abowd, and J. Fogarty, "Barriers and negative nudges: Exploring challenges in food journaling," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1159–1162.

[4] C. K. Martin, H. Han, S. M. Coulon, H. R. Allen, C. M. Champagne, and S. D. Anton, "A novel method to remotely measure food intake of free-living individuals in real time: the remote food photography method," *British Journal of Nutrition*, vol. 101, no. 3, pp. 446–456, 2008.

[5] J. Noronha, E. Hysen, H. Zhang, and K. Z. Gajos, "Platemate: crowd-sourcing nutritional analysis from food photographs," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 1–12.

[6] 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.

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

[8] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, "Analysis of food images: Features and classification," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2744–2748.

[9] M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, "A food recognition system for diabetic patients based on an optimized bag-of-features model." *IEEE J. Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1261–1271, 2014.

[10] 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.

[11] 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*. ACM, 2016, pp. 41–49.

[12] R. Tanno, K. Okamoto, and K. Yanai, "Deepfoodcam: A dcnn-based real-time mobile food recognition system," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. ACM, 2016, pp. 89–89.

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

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

[15] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment," in *International Conference on Smart Homes and Health Telematics*. Springer, 2016, pp. 37–48.

[16] Y. Kawano and K. Yanai, "Foodcam: A real-time food recognition system on a smartphone," *Multimedia Tools and Applications*, vol. 74, no. 14, pp. 5263–5287, 2015.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[21] N. Martinel, C. Piciarelli, and C. Micheloni, "A supervised extreme learning committee for food recognition," *Computer Vision and Image Understanding*, vol. 148, pp. 67–86, 2016.

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

[23] N. Martinel, G. L. Foresti, and C. Micheloni, "Wide-slice residual networks for food recognition," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 567–576.

[24] J. Chen and C.-W. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 32–41.

[25] J.-j. Chen, C.-W. Ngo, and T.-S. Chua, "Cross-modal recipe retrieval with rich food attributes," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1771–1779.

[26] M. Merler, H. Wu, R. Uceda-Sosa, Q.-B. Nguyen, and J. R. Smith, "Snap, eat, repeat: a food recognition engine for dietary logging," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. ACM, 2016, pp. 31–40.

[27] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, "Recipe recognition with large multimodal food dataset," in *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[29] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

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

[31] R. Adriana, B. Nicolas, K. S. Ebrahimi, C. Antoine, G. Carlo, and B. Yoshua, "Fitnets: Hints for thin deep nets," *Proc. ICLR*, 2015.

[32] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017. [Online]. Available: https://arxiv.org/abs/1612.03928

[33] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133–4141.

[34] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks."

[36] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[37] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[38] Y. Hao, Y. Fu, Y.-G. Jiang, and Q. Tian, "An end-to-end architecture for class-incremental object detection with knowledge distillation," *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2019.

[39] Y. Kawano and K. Yanai, "Foodcam-256: a large-scale real-time mobile food recognitionsystem employing high-dimensional features and compression of classifier weights," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 761–762.

[40] M. Bolanos and P. Radeva, "Simultaneous food localization and recognition," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 3140–3145.

Heng Zhao Heng Zhao received the Bachelor degree of Electrical and Electronic Engineering (University Medal Winner and First Class Honours) from Nanyang Technological University, Singapore in 2016. He is currently pursuing his PhD degree at the same university. His main research interests are image processing, deep learning and food recognition.

Kim-Hui Yap Dr. Kim-Hui Yap received the Bachelor of Electrical Engineering and PhD, both from the University of Sydney, Australia. He is currently an Associate Professor at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. His main research interests include artificial intelligence, data analytics, image/video processing and computer vision. He has authored more than 100 technical publications in various international peer-reviewed journals, conference proceedings and book chapters. He has also authored a book entitled *”Adaptive Image Processing: A Computational Intelligence Perspective, Second Edition”* published by the CRC Press.

Dr. Yap has served as Associate Editor and Editorial Board Member for a number of international journals. He participated in the organization of various international conferences, serving in different capacities including Technical Program Co-Chair, Finance Chair, and Publication Chair in these conferences.

Alex Chichung Kot Prof. Alex Chichung Kot has been with the Nanyang Technological University, Singapore since 1991. He headed the Division of Information Engineering at the School of Electrical and Electronic Engineering for eleven years and served as Associate Chair/ Research and Vice Dean Research for the School of Electrical and Electronic Engineering and eight years as Associate Dean for College of Engineering. He is currently Professor and Director of Rapid-Rich Object SEarch (ROSE) Lab and NTU-PKU Joint Research Institute. He has published extensively in the areas of signal processing for communication, biometrics, image forensics, information security and computer vision and machine learning.

Dr. Kot served as Associate Editor for more than ten journals, mostly for IEEE transactions. He has served the IEEE SP Society in various capacities such as the General Co-Chair for the 2004 IEEE International Conference on Image Processing and the Vice-President for the IEEE Signal Processing Society. He received the Best Teacher of the Year Award and is a co-author for several Best Paper Awards including ICPR, IEEE WIFS and IWDW. He was elected as the IEEE Distinguished Lecturer for the Signal Processing Society and the Circuits and Systems Society. He is a Fellow of IES, a Fellow of IEEE, and a Fellow of Academy of Engineering, Singapore.

Lingyu Duan Lingyu Duan is a Full Professor with the National Engineering Laboratory of Video Technology (NELVT), School of Electronics Engineering and Computer Science, Peking University (PKU), China, and has served as the Associate Director of the Rapid-Rich Object Search Laboratory (ROSE), a joint lab between Nanyang Technological University (NTU), Singapore, and Peking University (PKU), China since 2012. He is also with Peng Cheng Laboratory, Shenzhen, China, since 2019. He received the Ph.D. degree in information technology from The University of Newcastle, Callaghan, Australia, in 2008.

His research interests include multimedia indexing, search, and retrieval, mobile visual search, visual feature coding, and video analytics, etc. He has published about 200 research papers. He received the IEEE *ICME* Best Paper Award in 2019, the IEEE *VCIP* Best Paper Award in 2019, and *EURASIP Journal on Image and Video Processing* Best Paper Award in 2015, the Ministry of Education Technology Invention Award (First Prize) in 2016, the National Technology Invention Award (Second Prize) in 2017, China Patent Award for Excellence (2017), the National Information Technology Standardization Technical Committee ”Standardization Work Outstanding Person” Award in 2015. He was a Co-Editor of MPEG Compact Descriptor for Visual Search (CDVS) Standard (ISO/IEC 15938-13) and MPEG Compact Descriptor for Video Analytics (CDVA) standard (ISO/IEC 15938-15). Currently he is an Associate Editor of *ACM Transactions on Intelligent Systems and Technology* (ACM TIST) and *ACM Transactions on Multimedia Computing, Communications, and Applications* (ACM TOMM), and serves as the area chairs of ACM MM and IEEE ICME. He is a member of the MSA Technical Committee in IEEE-CAS Society.