Long-Tailed Continual Learning For Visual Food Recognition

Jiangpeng He, *Member, IEEE,* Xiaoyan Zhang, Luotao Lin, Jack Ma,
Heather A. Eicher-Miller, and Fengqing Zhu, *Senior Member, IEEE*

***Abstract*—Deep learning-based food recognition has made significant progress in predicting food types from eating occasion images. However, two key challenges hinder real-world deployment: (1) continuously learning new food classes without forgetting previously learned ones, and (2) handling the long-tailed distribution of food images, where a few common classes and many more rare classes. To address these, food recognition methods should focus on long-tailed continual learning. We introduce VFN186, a dataset comprising 186 food types that provides a more comprehensive representation of the American diet compared to the existing VFN dataset. We also introduce two new benchmark datasets, VFN186-INSULIN and VFN186-T2D, which reflect real-world food consumption for insulin takers and individuals with type 2 diabetes without taking insulin. We propose a novel end-to-end framework that improves the generalization ability for instance-rare food classes using a knowledge distillation-based predictor to avoid misalignment of representation during continual learning. Additionally, we introduce an augmentation technique by integrating class-activation-map (CAM) and CutMix to improve generalization on instance-rare food classes. Our method, evaluated on Food101-LT, VFN-LT, VFN186-LT, VFN186-INSULIN, and VFN186-T2DM, shows significant improvements over existing methods. An ablation study highlights further performance enhancements, demonstrating its potential for real-world food recognition applications.**

***Index Terms*—Continual learning, long-tailed distribution, food recognition, knowledge distillation, data augmentation**

I. Introduction

The emergence of modern deep learning technologies has enabled automatic food nutrition analysis, including image-based dietary assessment [1]–[4], to monitor and improve dietary intake and prevent chronic diseases like diabetes. As the first step in this process, food recognition identifies food types from images, and accurate recognition is critical for overall assessment performance. Despite deep learning-based methods [5]–[8] feature remarkable performance by training off-the-shelf Convolutional Neural Networks (*e.g.* ResNet [9]) using static datasets (*e.g.* Food-101 [10], Food2K [11]), two major challenges remain in real-world applications: (i) updating models as new food classes emerge over time, and (ii) addressing severe class imbalance in long-tailed distributions, where a few classes (head classes) dominate consumption compared with most others (tail classes) [12]. Failing to address these can significantly degrade performance.

Continual learning, also known as incremental or lifelong learning, allows models to learn new classes continuously without catastrophic forgetting [13]. Unlike retraining from scratch whenever encountering a new class, continual learning is more practical, requiring only new class data, which improves time, computation, and memory efficiency [14]. The challenge intensifies when the data follows a long-tailed distribution, requiring the model to address both catastrophic forgetting and class imbalance. While recent work [15] introduced a 2-stage framework to tackle this, its manual fine-tuning and detached training stages pose inefficiencies for real-world use. Additionally, existing methods have not been specifically applied to food images, which could be further challenges due to high intra-class variation and inter-class similarity.

Existing continual learning methods show the effectiveness of applying knowledge distillation and storing a small fixed number of seen images as exemplars to mitigate catastrophic forgetting. However, both techniques become less effective in long-tailed distribution. Specifically, the knowledge distillation [16] may even harm the performance when the teacher's model is not trained on balanced data due to the bias in output logits as shown in a recent study [17]. On the other hand, distilling knowledge through learned representations imposes a new challenge of feature space misalignment [18] as the learned representation needs to evolve during continual learning to accommodate new classes. Regarding using an exemplar set, most classes in long-tailed distribution may contain only a few training samples. Consequently, the overall performance may still be hindered even when all available samples are stored for instance-rare classes due to the poor generalization ability.

In this work, we focus on designing an end-to-end long-tailed continual learning framework for visual food recognition. We leverage feature-based knowledge distillation while incorporating an additional prediction head that projects the current representation space to the past. This addresses the misalignment issue by providing more freedom to the student model and encourages the retention of the learned knowledge. In addition, inspired by the most recent work [19] that uses the context-rich information in head classes to help the tail classes, we introduce a new data augmentation technique by integrating class-activation-map (CAM) and CutMix [20], which cuts the most important region calculated by CAM in instance-rare classes data as foreground and pastes into the instance-rich classes images. With minimal computational overhead, this method significantly enhances the generalization capabilities of tail classes. We evaluate our method on existing long-tailed food image datasets including Food101-LT and VFN-LT [12]. Additionally, we developed VFN186 based on the original VFN [7], expanding the initial 74 food categories by adding 112 more. This allows for a more comprehensive coverage of the typical American diet. Furthermore, we derive three long-tailed versions of VFN186, referred to as VFN186-LT, VFN186-INSULIN and VFN186-T2D, based on different population groups, namely healthy populations, Insulin Takers and those with Type 2 diabetes without taking insulin. Our proposed framework achieves the best performance with a large improvement margin compared to existing methods while not requiring detached training stages. Finally, we conduct an ablation study to evaluate the effectiveness of each component in our proposed method and discuss potential techniques that can boost the accuracy for implementation in real-world applications. The main contributions of this work are summarized in the following:

- We explore long-tailed continual learning related to food recognition in real-world scenarios.
- We introduce the VFN186 food dataset, which contains 186 most frequently consumed food types in America. Additionally, we introduce three new long-tailed benchmark datasets, which reflect the food consumption patterns of different populations.
- We propose a novel framework that utilizes feature-based knowledge distillation with a prediction head and a novel CAM-based CutMix for data augmentation, and an integrated loss function to address catastrophic forgetting and class imbalance.
- We conduct extensive experiments on all long-tailed continual learning benchmarks for food recognition and discuss potential techniques to enhance accuracy that could boost the accuracy for facilitating the deployment in real-world food recognition.

II. Related Word

In this section, we summarize existing methods most related to our work including food recognition, long-tailed recognition, and continual learning.

A. Food Recognition

Food image recognition is a challenging yet practical task with applications like image-based dietary assessment [21], [22], where accurate recognition is crucial for nutritional content analysis, such as energy and macronutrients [23]. Most existing deep learning based work leverage off-the-shelf models [9], [24]–[26] and train on static food image datasets [7], [10], [11], [27]–[29]. To address inter-class similarity and intra-class variability, various hierarchy-based approaches have been proposed [7], [8]. While food recognition has been studied in scenarios like ingredient recognition [30], fine-grained recognition [31], few-shot learning [32], long-tailed recognition [12], [33], and continual learning [34], no existing methods continuously learn new classes in long-tailed distributions, which is critical for real-world applications [12]. Recent work [15] attempted to integrate continual learning with long-tailed recognition but used a multi-stage training process and did not focus on food images. In this work, we target long-tailed continual learning for visual food recognition, introducing a novel end-to-end framework to address both class imbalance and catastrophic forgetting simultaneously.

B. Long-tailed Recognition

Existing work on image recognition in long-tailed distributions can be categorized into two main groups: *re-weighting* and *re-sampling*. The major challenge is the imbalance between instance-rich (head) and instance-rare (tail) classes. **Re-weighting** methods balance the loss or gradients during training, with a class-level re-weighting loss like Balanced Softmax [35] and Label-Distribution-Aware Margin loss [36]. In addition, **re-sampling** based techniques construct a balanced training set by over-sampling tail classes or under-sampling head classes, but naive approaches [37], [38] can lead to overfitting or performance degradation. CMO [19] used CutMix [20] to cut and paste regions between tail and head class images, leveraging context from head classes to improve tail class generalization. Inspired by [39], we introduce a novel CAM-based CutMix, which combines the images seamlessly without losing semantic information, as detailed in Section IV.

C. Continual Learning

Continual learning, also known as incremental or lifelong learning, has been explored in scenarios like class-incremental, task-incremental, and domain-incremental learning [40]. This work focuses on class-incremental learning, which is key for real-world applications. It involves continuously learning new classes and classifying all previously seen classes during inference, without using task indexes or multi-head classifiers as in task-incremental learning [41]. Unlike domain-incremental learning, which handles domain shifts without new classes, class-incremental learning faces the challenge of catastrophic forgetting [13], where the model forgets previous knowledge due to the lack of data from learned classes. To address this, existing methods are mainly divided into *regularization-based* and *replay-based* approaches.

Regularization-based methods address forgetting by limiting changes to learned parameters while learning new classes. Initial work froze or constrained parameter updates [42], [43], limiting the model's ability to learn new data. Later, Li *et al.* [44] used knowledge distillation [16] to preserve learned knowledge by mimicking the teacher model's output logits. Feature-based distillation minimized representation discrepancies [45]. However, these methods are less effective on long-tailed data, where distillation can harm performance [17]. Direct feature-based distillation also faces challenges like feature space misalignment [18]. We address this problem by adding a prediction head to map the current representation space to the past for more efficient knowledge transfer.

Replay-based methods use a memory buffer to store exemplar data for knowledge replay during class-incremental learning. The herding algorithm [46] selects exemplars based on class mean vectors and is widely used [47]–[50]. However, these methods assume balanced training data and sufficient samples per class compared with the memory budget (*e.g.* 20 exemplars per class), which isn't the case in long-tailed scenarios. It can lead to class imbalance in the exemplar set and harm overall performance. We address this issue by constructing a balanced exemplar set by augmenting the tail class data with the proposed CAM-based CutMix, which augments tail class data, improving knowledge replay efficiency and generalization on tail classes for better overall performance.

III. Datasets

A. VFN186 Dataset

To encompass a broader range of food types, we expanded VFN [7] to include 186 food categories. Similar to VFN, we select an additional 112 commonly consumed foods by Americans based on What We Eat In America (WWEIA) [51] database. Specifically, we first match each of the 186 food types in VFN186 with one 8-digit USDA food code from the Food and Nutrient Database for Dietary Studies (FNDDS). Each 8-digit USDA food code represents a specific food item in the food supply. This expansion makes the dataset more comprehensive and scientifically robust for training practical models with broader applicability. Then we use a semi-automatic data collection system to crawl specific types of food images from the Google Image website based on food labels. Next, we employ a trained Faster R-CNN to remove noisy images. The remaining images were processed using an online crowdsourcing tool, where food items were boxed and labeled with their corresponding categories. Through this process, we expand the VFN dataset and created the VFN186 dataset, which includes 186 food types and 70230 images.

B. Long-tailed Food Datasets For Diabetes

The most recent work [12] introduced VFN-LT, a long-tailed version of the VFN [7] dataset for food recognition, where the data distribution reflects real-world food consumption frequencies [52] among healthy individuals aged 18 to 65 in the U.S. We processed the newly created VFN186 similarly, producing its long-tailed version, VFN186-LT. However, around 34.2 million U.S. individuals (10.5 % of the U.S. population) have diabetes [53], which can cause various health problems such as heart disease, vision loss, and kidney disease. Since diet plays a crucial role in diabetes management, accurate food recognition to support dietary choices is vital. Yet, no studies have specifically targeted this population. In this work, we address this gap by introducing two new benchmark long-tailed datasets, VFN186-INSULIN and VFN186-T2D, designed for dietary assessment among insulin takers and those with type 2 diabetes who do not take insulin.

For VFN186-Insulin and VFN186-T2D, food codes are labeled with their corresponding consumption frequency, as determined by Lin *et al* [52], using the nationally representative dietary data from the National Health and Nutrition Examination Survey (NHANES) collected from 2009–2016 among U.S adults aged from 20 to 65. The consumption frequency represents how often a particular food item was reported in one day within a specific age group in the U.S. population, reflecting the food's prominence relative to others. The study included **774** participants for insulin takers and **2,758** participants for those with type 2 diabetes who do not take insulin. Next, we reduce the number of training samples for food classes in the original VFN186 based on the matched consumption frequency. Specifically, given the consumption frequency f_i for the ith food class, we calculate the number of training samples by (1)

$$s_i = n_i \times \frac{f_i}{f_{max}} \tag{1}$$

where n_i is the original number of training samples in VFN186 for food class i and s_i denotes how many training images are retained for this class, which are randomly selected among original training data. f_{max} refers to the maximum matched consumption frequency among the 186 food types in VFN datasets.

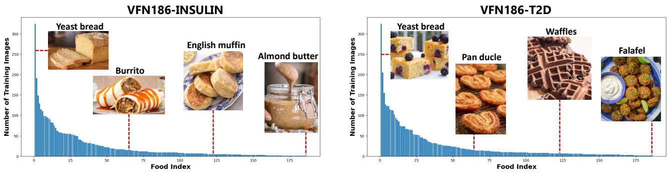

Fig. 1. The distribution of VFN186-INSULIN and VFN186-T2D shown in descending order based on the number of training samples.

Overall, VFN-INSULIN contains 4,179 training images from 186 food classes, and VFN-T2D contains 4,403 training images, with a maximum of 324 and a minimum of 1 image per class. The food type *Yeast bread* is highly consumed among represented adults and dominates the consumption frequency in both groups, resulting in an imbalance ratio $\rho = \frac{max_i\{s_i\}}{min_i\{s_i\}}$ of 324 in both datasets. However, the consumption frequencies of other food types vary between the two groups. Figure 1 shows the distribution of food types in VFN186-INSULIN and VFN186-T2D, ranked by the number of training samples per class.

IV. Method

In this work, we introduce a novel end-to-end long-tailed continual learning framework for visual food recognition. The overview of our method is shown in Figure 2. To address catastrophic forgetting, we leverage the teacher model learned from the last incremental step and perform feature-based knowledge distillation with an additional prediction head to enable efficient knowledge transfer. The exemplar set was selected based on a novel CAM-based data augmentation for tail classes. Finally, we replace the cross-entropy with the balanced softmax loss [35] based on the current training data distribution to learn class-balance visual representation. In this section, we first introduce the preliminaries for continual learning in the long-tailed distribution in Section IV-A and then illustrate the detail of each proposed component in Section IV-B, IV-C and IV-D, respectively.

A. Preliminaries

We focus on continual learning in class-incremental settings where the objective is to learn new classes incrementally and perform classification on all classes seen so far during the inference phase. Specifically, the continual learning in the class-incremental scenario can be formulated as applying an initial model h_0 to learn a sequence of N tasks denoted as $\mathcal{T} =$

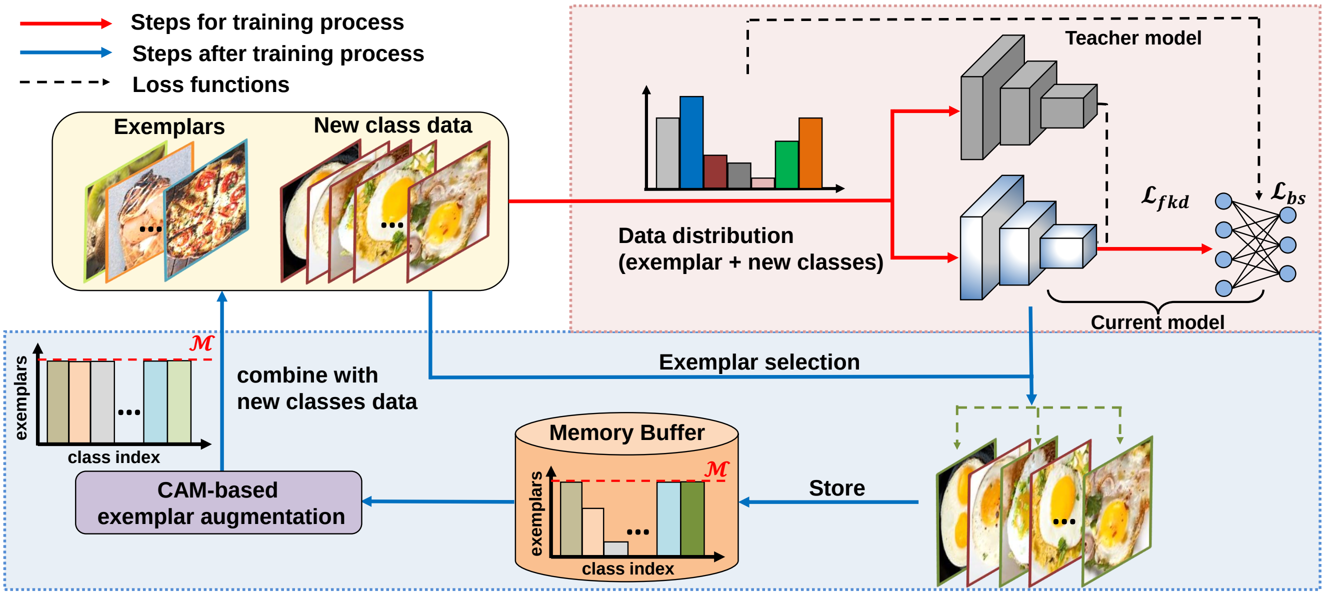

Fig. 2. The overview of our proposed framework. The red arrows show the training process with new class images and exemplars from previous classes. The blue arrows denote the steps after the training process where we construct a balanced exemplar set and store them in the memory buffer.

$\{\mathcal{T}^1, ..., \mathcal{T}^N\}$ where each task $\mathcal{T}^i$ contains C_i non-overlapped new classes, which is also known as the incremental step size. During the learning phase of each new task, only the training data $D_i = \{\mathbf{x}_i^j, y_i^j\}$ of the current task is available where $\mathbf{x}_i^j$ and y_i^j denote the j-th input image and label, respectively. After each incremental learning step, the updated model h_i needs to classify $C_{1:i}$ classes encountered so far. The major challenge of continual learning is catastrophic forgetting [13] where the updated model h_i after learning the task $\mathcal{T}^i$ forgets the knowledge of previous tasks $\{\mathcal{T}^1, ..., \mathcal{T}^{i-1}\}$, resulting in significant performance degradation to classify $C_{1:i-1}$. In the conventional setup, the training data D_i for each task $\mathcal{T}^i$ is evenly distributed, containing $|D_i|/C_i$ samples per class. However, this assumption simplifies the real-world complexities, especially for food recognition where data is usually long-tailed distributed and exhibits imbalance among food classes. Formally, the training data D_i for each task in long-tailed continual learning is a class-imbalanced distributed with each class containing $(0, |D_i|)$ training samples. The entire training data D for all the N tasks $\mathcal{T}$ exhibits the long-tail distribution.

1) Knowledge distillation: Most existing work [44], [47]–[50] applies knowledge distillation [16] on output logits to maintain the performance on previously learned classes. Specifically, during the learning step of the task $\mathcal{T}^i$, a teacher model $h_t = h_{i-1}$ learned from the last task with fixed parameters is employed. The knowledge distillation aims to minimize the difference between the output logits of the current model $L = [o^1, o^2, ...o^{C_{1:i}}] \in \mathbb{R}^{C_{1:i}\times 1}$ and the outputs of the teacher model $\hat{L} = [\hat{o}^1, \hat{o}^2, ...\hat{o}^{C_{1:i-1}}] \in \mathbb{R}^{C_{1:i-1}\times 1}$ by

$$\mathcal{L}_{kd} = -\sum_{j=1}^{C_{1:i-1}} \hat{L}_T^{(j)} log(L_T^{(j)}) \tag{2}$$

where T is the temperature scalar to learn the hidden knowledge by softening the output distribution as

$$\hat{L}_T^{(j)} = \frac{\exp{(\hat{o}^{(j)}/T)}}{\sum_{k=1}^{C_{1:i-1}} \exp{(\hat{o}^{(k)}/T)}} \tag{3}$$

Finally, the knowledge distillation is integrated with cross-entropy during the training process by using a hyper-parameter α to learn new classes and maintain the learned knowledge.

$$\mathcal{L} = \alpha\mathcal{L}_{kd} + (1-\alpha)\mathcal{L}_{cn} \tag{4}$$

2) Exemplar replay: As one of the most commonly used strategies to address catastrophic forgetting, the exemplar replay-based methods [45], [47], [49] assume the availability of a reasonable memory budget to select a small fixed number of data as exemplars for each seen class and store them in memory buffer (also known as exemplar set). Specifically, after learning each task $\mathcal{T}^i$, the lower layers of updated model h_i are used to extract feature embeddings for the new classes training data $D_i = \{\mathbf{x}_i^j, y_i^j\}$. The Herding algorithm [46] is widely applied to select the most representative data for each new class based on the Euclidean distance between feature embedding and the class mean vector. Therefore, given a memory budget of $\mathcal{M}$ data per class (also known as memory capacity), a subset of $E_i \subseteq D_i$ is selected with $|E_i| = \mathcal{M} \times C_i$ and stored in the memory buffer. Finally, at the beginning of the next new task $\mathcal{T}^{i+1}$, all the exemplars in the memory buffer are combined with the new classes training data to construct $E_i + D_{i+1}$ for continual learning. In this work, we use Herding as the exemplar selection algorithm while other latest work [34] could also be applied.

B. Feature-based Knowledge Distillation

Despite the effectiveness of knowledge distillation in conventional continual learning setup as described in Section IV-A1, it is difficult to apply it to long-tailed distributions since the output logits of the teacher model can be

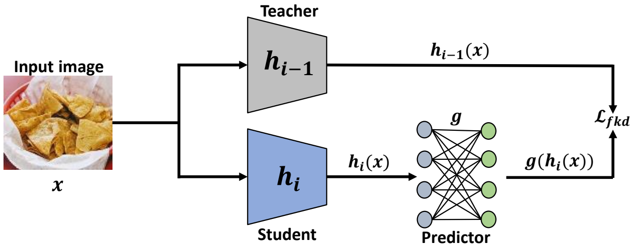

Fig. 3. The overview of proposed feature-based knowledge distillation by applying an additional predictor g.

heavily biased towards instance-rich classes [50]. Directly applying knowledge distillation as in (2) on biased output logits may even harm the overall performance [17]. Therefore, we explore feature-based knowledge distillation for better knowledge transfer in long-tailed continual learning. However, a key challenge is feature space misalignment of the challenges when applying feature-based distillation, where the representation of student and teacher models could mismatch in terms of both magnitude and direction [18]. This problem is also relevant in continual learning as the model evolves to incorporate new classes. To solve this, we introduce a simple yet effective method as shown in Figure 3. Specifically, instead of directly mimicking the feature from the teacher model, we apply an additional predictor g on the head of the continual learning model to map the current representation space to the past in the teacher model. g is a single-layer perceptron that performs domain mapping while preserving consistent dimensions. Specifically, it maps from $\mathbb{R}^{d\times 1}$ to $\mathbb{R}^{d\times 1}$, followed by a ReLU activation function. The dimensional consistency is crucial to ensure that the student model, with the added g, still outputs image features of the same size, i.e., $d \times 1$. Given an image $\mathbf{x}$, the predictor g takes the feature representation from the current model (*i.e.* student model) $h_i(\mathbf{x})$ as input and outputs the mapped feature $g(h_i(\mathbf{x}))$. g is a Then we distill the knowledge from the teacher model h_{i-1} by

$$\mathcal{L}_{fkd}(\mathbf{x})) = 1 - < g(h_i(\mathbf{x})), h_{i-1}(\mathbf{x})) > \quad (5)$$

where $<,>$ measures the cosine similarity. By applying the predictor, we give the student model more freedom to accommodate the previously learned representation into the current feature space, enabling more efficient knowledge distillation in long-tailed continual learning. The predictor g is removed after each incremental learning phase. Note that although we apply cosine embedding loss for knowledge distillation, our method can be integrated with other loss functions such as the Mean Squared Error (MSE) loss.

C. CAM-based Exemplar Augmentation

Existing exemplar replay-based methods assume each class should contain at least $\mathcal{M}$ images given $\mathcal{M}$ as the memory budget in Section IV-A2. However, most classes in long-tailed distribution may contain only a few training samples $n < \mathcal{M}$, which imposes two new challenges including (i) inefficiency of knowledge replay due to the insufficient training samples and (ii) intensification of the class-imbalance issue if we directly combine the stored exemplars with training data from new class due to the imbalanced nature of memory buffer. Therefore, we propose a novel data augmentation method in this work to construct a balanced exemplar set by augmenting the tail class images to address both aforementioned issues. The overview of the proposed data augmentation technique is illustrated in Figure 4. To address the issue of losing semantic information when performing data augmentation [12], [19] as described in Section II-B, we propose to use a class activation map (CAM) [54] to identify the most important region from instance-rare classes images and then preserve the semantic information by performing CutMix [20] to cut and paste the identified region into the images with rich context that are selected based on visual similarity. Specifically, we construct a class-balanced memory buffer before each new task $\mathcal{T}^{i+1}$ by augmenting stored images for food classes C_t with less than $\mathcal{M}$ exemplars through CutMix in conjunction with images selected from food classes C_h containing $\mathcal{M}$ exemplars. Given an input image $\mathbf{x}_t \in C_t$, we first select the most visually similar candidate $\mathbf{x}_h \in C_h$ by comparing the cosine similarity with h_i as feature extractor where $\mathbf{x}_h = \underset{\mathbf{x}_k \in C_h}{\operatorname{argmax}} < h_i(\mathbf{x}_t), h_i(\mathbf{x}_k) >$. The lower half of Figure 4 illustrates the procedure to identify the region to cut and paste into $\mathbf{x}_h$. Formally, given $\mathbf{x}_t \in \mathbb{R}^{c\times h\times w}$, the class-activation map $M(\mathbf{x}_t) \in \mathbb{R}^{h\times w}$ is calculated by

$$M(\mathbf{x}_t) = \sum_{k}^{d} v^k_{y_{\mathbf{x}_t}} h_i(\mathbf{x}_t) \quad (6)$$

where $v_{y_{\mathbf{x}_t}} \in \mathbb{R}^d$ refers to the weight vector in the classifier of the current model corresponding to the seen class $y_{\mathbf{x}_t} \in C_{1:i}$. The value of CAM ranges from $[0,1]$ and a higher value indicates the more discriminative class-specific region. Therefore, we apply a random threshold $\sigma \in (0,1)$ to select the region $M(\mathbf{x}_t)^T \in \mathbb{R}^{h\times w}$ where

$$M(\mathbf{x}_t)^T = \begin{cases} M(\mathbf{x}_t) & M(\mathbf{x}_t) \geq \sigma \\ 0 & M(\mathbf{x}_t) < \sigma \end{cases} \quad (7)$$

without losing the semantic information of the input image. Finally, we apply CutMix to generate a synthetic image $\tilde{\mathbf{x}}_t$ by

$$\tilde{\mathbf{x}}_t = (1 - S(\mathbf{x}_t)^T) \odot \mathbf{x}_h + S(\mathbf{x}_t)^T \odot \mathbf{x}_t \quad (8)$$

where $\odot$ refers to element-wise multiplication and $S(\mathbf{x}_t)^T$ denotes the binary mask obtained from $M(\mathbf{x}_t)^T$ that $\mathbf{1}$ indicates the region with $M(\mathbf{x}_t)^T > 0$. The class label $\tilde{y}_t$ of the synthetic image is calculated by the area of the replaced region in $\mathbf{x}_h$ as

$$\tilde{y}_t = \frac{1 - A_r}{A} y_h + \frac{A_r}{A} y_t \quad (9)$$

where A_r and A denote the area of the replaced region and the total area of $\mathbf{x}_h$, and y_h and y_t are the original class labels of $\mathbf{x}_h$ and $\mathbf{x}_t$. The exemplar augmentation is performed at the beginning of each new task and the augmented images are not stored in the memory buffer. Note that the Grad-CAM [55], which can be regarded as the generalization of CAM [54], could also be applied in our method.

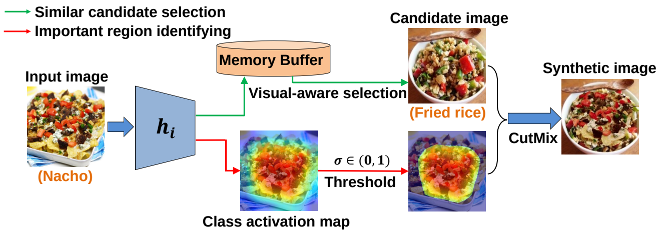

Fig. 4. The overview of proposed CAM-based data augmentation technique. The green arrow describes the selection of the most visually similar candidate image and the red arrow illustrates the steps to obtain the most important region of the input image to perform CutMix [20].

D. Integrated Loss

While the exemplar augmentation mitigates the class-imbalance issue by constructing a balanced memory buffer, the number of available training data between new classes and the stored classes may still vary a lot during the training phase due to the limited memory budget. Existing work [15] addresses this problem by decoupling the training process into two stages to first learn a feature extractor and then fine-tune the classifier using a class-balanced sampler. In this work, we propose to use Balanced Softmax (BS) [35] by extending it into a long-tailed continual learning scenario without requiring a decoupled training process. Specifically, during the training phase of the new task $\mathcal{T}^i$, a distribution vector $v_d \in \mathbb{R}^{C_{1:i}}$ is generated by counting the number of training data of each food class for input images in the current task. Recall $L \in \mathbb{R}^{C_{1:i}}$ is the output logits from current model h_i, the distribution vector is then used as the prior information when calculating the loss as shown in (10)

$$\mathcal{L}_{bs} = \sum_{k=1}^{C_{1:i}} -y^k log[\Phi(\bar{v}_d^k + L^k)] \tag{10}$$

where $\bar{v}_d = v_d / sum(v_d)$ is the normalized distribution vector and $\Phi()$ denotes the $Softmax$ function. Therefore, the larger value in the distribution vector achieves smaller gradients when we compute the cross-entropy using the adjusted logits $v_t + L$ and vice versa. This addresses the class-imbalance issue and enables the end-to-end training pipeline.

The overall training loss function is the weighted sum of feature-based knowledge distillation as described in (5) and the balanced softmax $\mathcal{L}_{bs}$, which can be expressed as

$$\mathcal{L} = \mathcal{L}_{bs} + \lambda \mathcal{L}_{fkd} \tag{11}$$

where λ is the adaptive ratio to tune the two losses. In this work, as the number of training data D_i may vary a lot for each task $\mathcal{T}^i$, we propose to calculate $\lambda = \sqrt{|D_i|/|D_{1:i}|}$ as the ratio of training data for the current task and the learned tasks. Therefore, the ratio λ increases when there are more training data from new classes.

V. Experiment

In this section, we evaluate our proposed long-tailed continual learning framework for visual food recognition as illustrated in Section IV. Specifically, we first introduce the experimental setup including the split of datasets and implementation detail described in Section V-A and V-B. Then we compare our method with existing work in Section V-C and conduct an ablation study to show the effectiveness of each individual component in Section V-D. Finally, we discuss potential techniques that can boost the performance of real-world food-related applications in Section V-E.

A. Datasets

We conduct experiments on long-tailed food image datasets: (1) Food101-LT [12], (2) VFN-LT [12], (3) VFN186-LT, (4) VFN186-INSULIN and (5) VFN186-T2D, with the latter four introduced in Section III.

Food101-LT is the long-tailed version of Food-101 [10], created using the *Pareto distribution* [59] with the power ratio of $\alpha = 6$. We randomly partition the 101 food classes into 5, 10, and 20 tasks for continual learning, where each task introduces 20, 10, and 5 new classes, respectively, except the first task with one extra class. The test set is kept as balanced with 125 images per class.

VFN-LT is a long-tailed version of VFN [7] based on food consumption frequency of healthy people. The 74 food classes are split into 7 tasks, with the first task containing 14 new classes and the remaining tasks containing 10 new classes. The test set has 25 images per class.

VFN186-LT, VFN186-INSULIN and **VFN186-T2D** are long-tailed versions of VFN186. VFN186-LT is created similarly to VFN-LT, while VFN186-INSULIN and VFN186-T2D are based on food consumption frequencies of insulin takers and individuals with type 2 diabetes without insulin. We divide 186 food classes into $N = 9$ tasks with the first task containing 26 new classes and the rest containing 20. To facilitate an equatable analysis, we use the same testing data in VFN186-LT, VFN186-INSULIN and VFN186-T2D, which is balanced with 25 samples per class, totaling 4,650 images.

B. Implementation Detail

Our implementation of neural networks are based on the Pytorch framework and we apply the ResNet-18 from scratch as the backbone for all experiments. The ResNet implementation follows the setting suggested in [9]. We train each new task for 90 epochs with the learning rate starting from 0.1 and decreasing with a ratio of $1/10$ for every 30 epochs. The batch size is set to 128 and we apply the stochastic gradient descent (SGD) optimizer with a weight decay of 0.0001. The memory budget is set to $\mathcal{M} = 20$ to store at most 20 images per food class in the memory buffer.

Evaluation protocol: We use Top-1 classification accuracy to evaluate the model after each task $\mathcal{T}^i$ on test data covering previously seen classes $C_{1:i}$. Besides, we report the average accuracy A_M, calculated by averaging the accuracy after each task, which shows the overall performance across the continual learning procedure. Each experiment is run five times and the average performance is presented.

TABLE I

RESULTS ON FOOD101-LT, VFN-LT, VFN186-LT, VFN186-INSULIN, AND VFN186-T2D BY COMPARING WITH EXISTING CONTINUAL LEARNING METHODS IN TERMS OF AVERAGE ACCURACY (A_M). BEST RESULTS ARE MARKED IN BOLD.

Datasets	**Food101-LT**			**VFN-LT**	**VFN186-LT**	**VFN186-INSULIN**	**VFN186-T2D**
Number of tasks	$N = 5$	$N = 10$	$N = 20$	$N = 7$	$N = 9$	$N = 9$	$N = 9$
LwF [44]	10.02	5.86	0.83	4.85	11.60	10.99	11.12
EWC [43]	5.05	3.70	0.83	6.31	11.09	10.40	4.12
iCaRL [47]	12.42	12.46	11.04	18.76	8.76	7.23	7.56
EEIL [49]	12.68	10.57	6.98	20.77	11.70	10.83	11.49
LwM [56]	10.82	7.22	2.45	12.32	11.04	10.37	10.88
IL2M [17]	11.45	10.97	6.81	18.68	11.23	10.52	11.30
BiC [50]	16.72	12.39	10.38	20.89	9.09	9.27	11.08
LUCIR [45]	16.94	10.17	6.74	23.37	14.54	11.73	12.97
PODNet [57]	12.22	10.46	8.99	18.21	13.40	11.87	12.56
EEIL-2stage [15]	14.96	13.29	9.76	22.98	12.86	11.74	12.69
LUCIR-2stage [15]	18.90	13.03	10.85	24.26	15.80	13.64	14.88
PODNet-2stage [15]	17.89	11.12	10.28	25.58	16.00	13.77	15.11
DGR [58]	23.08	20.35	16.43	26.11	19.58	17.66	17.90
Ours	**27.52**	**25.12**	**21.72**	**27.53**	**22.21**	**20.61**	**21.23**

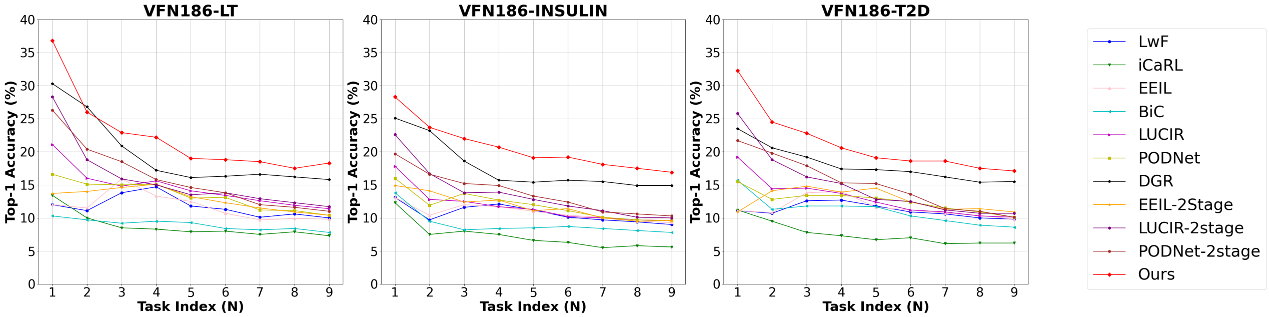

Fig. 5. Results on VFN186-LT, VFN186-INSULIN and VFN186-T2D with the number of tasks $N = 9$. Each marker represents the Top-1 classification accuracy evaluated on all classes seen so far after learning each task.

C. Comparisons With Existing Methods

Table I summarizes the average accuracy A_M on Food101-LT, VFN-LT, VFN186-LT, VFN186-INSULIN and VFN186-T2D. Our method shows significant improvements, particularly on Food101, with different numbers of tasks $N \in \{5, 10, 20\}$, achieving approximately a 5% increase in accuracy. Moreover, there are significant enhancements on VFN186-LT, VFN186-INSULIN, and VFN186-T2D, which feature more imbalanced distributions as discussed in Section III. For example, on three long-tailed VFN186, we achieve about a 7% increase over the 2-stage framework even without requiring a decoupled training process and a 3% improvement compared to DGR. However, tends often decreases as the total number of tasks N increases. Therefore, we need to address the catastrophic forgetting to maintain the learned knowledge at each learning phase of new tasks after the first task.

Figure 5 shows top-1 classification accuracy across all seen classes after each task. Our method achieves promising performance at each stage of the new task. Interestingly, in long-tailed scenarios, unlike conventional continual learning where accuracy typically declines over time, we observe improvements after learning new tasks sometimes. For example, on VFE186-INSULIN, accuracy increases for most methods after the third task. This occurs because the number of training samples varies significantly among different tasks in long-tailed continual learning where the model gains better knowledge for tasks with a larger number of training images. However, it also imposes new challenges in handling class-imbalance across different tasks and hyper-parameter tuning (*e.g.* the knowledge distillation factor in Equation 4).

In this work, we address catastrophic forgetting through feature-based knowledge distillation and exemplar augmentation, while also alleviating class-imbalance by integrating balanced softmax with an adaptive ratio to adjust the impact of each loss term, as illustrated in Equation 11, and strike a balance of stability-plasticity.

D. Ablation Study

In this section, we evaluate the effectiveness of each individual component in our proposed framework including (i) the feature-based knowledge distillation ($\mathcal{L}_{fkd}$), (ii) the cam-based data augmentation (**CAM-CutMix**) and (iii) the integration of balanced softmax with adaptive ratio ($\mathcal{L}_{bs}$). Formally, we consider the *baseline* method as using an imbalanced memory buffer ($\mathcal{M} = 20$) with cross-entropy loss and integrating each of the components mentioned above to conduct

TABLE II
ABLATION STUDY ON FOOD101-LT AND VFN-LT IN TERMS OF AVERAGE ACCURACY A_M.

			Food101-LT			VFN-LT
$\mathcal{L}_{fkd}$	**CAM-CutMix**	$\mathcal{L}_{bs}$	$N = 5$	$N = 10$	$N = 20$	$N = 7$
			5.90	8.79	10.55	12.21
✓			17.42	15.83	13.96	22.53
	✓		13.27	12.99	11.64	16.73
		✓	16.52	14.20	12.02	22.96
✓	✓		19.31	17.26	15.49	24.18
✓	✓	✓	**21.83**	**19.25**	**17.43**	**29.33**

TABLE III
ABLATION STUDY OF DIFFERENT DATA AUGMENTATION METHODS ON FOOD101-LT ($N = 10$) AND VFN-LT WITH AVERAGE ACCURACY A_M.

	Food101-LT ($N = 10$)	VFN-LT
CMO [19]	17.28	25.93
VM-CMO [12]	16.47	26.41
SnapMix [39]	18.31	27.62
D-Mixup [33]	15.93	25.14
CAM-CutMix (Ours)	**19.25**	**29.33**

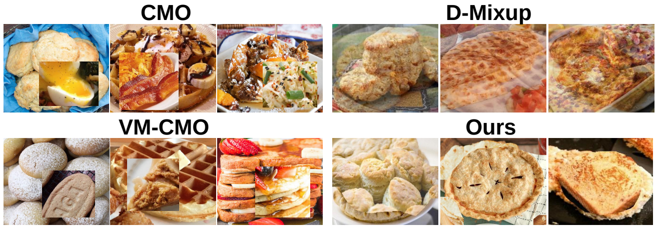

Fig. 6. Examples of augmented food images on VFN-LT using CMO [19], VM-CMO [12], D-Mixup [33] and our proposed CAM-CutMix.

experiments. The results in terms of average accuracy A_M are summarized in Table II. We observe consistent performance improvements compared with *baseline* by adding our proposed techniques. Specifically, the feature-based knowledge distillation $\mathcal{L}_{fkd}$ achieves the largest improvements on the Food101-LT dataset, demonstrating that catastrophic forgetting is a crucial issue and the integration with CAM-CutMix can achieve higher accuracy. On the other hand, as VFN-LT exhibits more severe class-imbalance problems due to a higher imbalance ratio, the balanced softmax $\mathcal{L}_{bs}$ term has the most significant impact, resulting in the largest performance improvements. By integrating all three components, our proposed framework obtains the best classification accuracy on these two datasets.

We also evaluate our proposed CAM-CutMix by replacing it with existing data augmentation based methods including the CutMix [20] based approaches: (a) the original CutMix used in **CMO** [19], (b) Visual-Multi CutMix (**VM-CMO**) [12], (c) **SnapMix** [39] and the Mixup [60] based approach: (d) **D-Mixup** [33]. We conduct experiments on VFN-LT and Food101-LT with $N = 10$ as shown in Table III. Generally, the CutMix-based approaches work better in long-tailed continual learning scenarios than D-Mixup, which is usually applied in multi-label recognition scenarios. In addition, the SnapMix achieves a slightly better performance than CMO and VM-CMO as it also considers the class-activation map (CAM) when generating mixed labels. Our method achieves the best performance as it not only preserves the most important regions based on CAM but also enables seamless CutMix, rather than relying on a randomly generated bounding box. The example augmented food images using VFN-LT are shown in Figure 6. Note that we do not visualize SnapMix [39] as it has the same synthetic image as in CMO [19] but with a different mixed label.

E. Discussions

Despite the performance improvements our framework demonstrates compared to existing methods as shown in Table I, the deployment in real-world applications remains challenging based on the current classification accuracy. Therefore, in this section, we discuss potential techniques that could be applied to boost the performance including (1) increasing the memory buffer capacity to store more exemplar images for knowledge replay and (2) performing transfer learning by pre-training the backbone on large-scale image datasets.

1) Memory buffer capacity: As one of the most efficient techniques to address catastrophic forgetting, the performance of knowledge replay greatly relies on the capacity of the memory buffer (*i.e.* how many exemplar images can be stored). In this part, we evaluate the long-tailed continual learning performance by varying the memory buffer capacity $\mathcal{M} \in \{10, 20, 30, 40, 50, 100\}$. Table IV shows the results in terms of average accuracy A_M on Food101-LT ($N = 10$) and VFN-LT where we observe consistent performance improvements by increasing the memory capacity. However, the memory buffer capacity is a significant constraint for continual learning in real-world applications as it requires larger memory storage and also poses challenges related to privacy concerns when storing original images as exemplars. Additionally, we observe a performance bottleneck where increasing memory capacity does not substantially boost performance. This is predominantly due to dual challenges of catastrophic forgetting and class-imbalance problems that arise in the long-tailed continual learning scenario.

TABLE IV
AVERAGE ACCURACY (A_M) ON FOOD101-LT ($N = 10$) AND VFN-LT BY VARYING THE MEMORY BUFFER CAPACITY $\mathcal{M} \in \{10, 20, 30, 40, 50, 100\}$.

Buffer Size	**10**	**20**	**30**	**40**	**50**	**100**
Food101-LT	16.96	19.69	20.71	22.15	23.14	24.30
VFN-LT	25.71	29.33	30.84	31.25	31.89	32.55

2) Transfer learning: Applying the deep models pre-trained on large-scale image datasets as the backbone is a common strategy to enhance performance in many vision tasks [11], [61]. In this part, we evaluate the efficacy by leveraging a variety of pre-trained models for long-tailed continual learning in visual food recognition. We consider different network structures with various depth such as *ResNet-50* [9], *MobileNet* [62], *EfficientNet* [63] *Vision Transformers (ViT)* [64] and its variants *DeiT* [65] and *Swin* [66] transformers. In addition, we leverage ImageNet-1K [67] and ImageNet-21K [68] as the pre-training datasets. ImageNet-1K contains 1,000 classes of general objects, which is the subset of full ImageNet-21K that contains 21,841 classes with over 14,197,122 training images. The VFN-LT results in average accuracy A_M are shown in Table V. We observe over 20% performance improvements by using pre-trained models on large-scale datasets compared to our results in Table I with a model from scratch. It manifests that pre-training enhances the backbone network's feature extraction capabilities, thereby yielding the most discriminative features essential for downstream tasks. In addition, pre-training on larger-scale datasets with more images and classes makes higher accuracy. However, there is a trade-off between the computation complexity and the performance where the increase of model parameters would require longer training time and higher computation capability, which may not be practical for specific real-world applications with limited resources. Note that we intentionally refrain from utilizing food datasets for pre-training in this part to prevent potential overlap with any food class in VFN [7], though there may be a more substantial performance enhancement if pre-trained on large-scale food datasets such as Food2K [11].

TABLE V

Average accuracy (A_M) on VFN-LT by leveraging pre-trained models.

Model	**MobileNet**	**ResNet**	**EfficientNet**	**ViT**	**DeiT**	**Swin**
Parameters (10M)	1.8	2.5	5.4	8.2	8.5	8.8
ImageNet-1K	54.99	56.73	61.78	59.75	61.41	63.17
ImageNet-21K	56.64	58.92	63.83	64.79	65.01	71.18

VI. Conclusion

In this work, we focus on visual food recognition in long-tailed continual learning. We create an expanded dataset VFN186 and its three benchmark long-tailed food image datasets that exhibit the real-life food consumption frequency. The proposed end-to-end framework combines effective feature-based knowledge distillation and a novel data augmentation module, capable of learning new food classes in long-tailed data distribution without forgetting the learned knowledge. Our method outperforms existing approaches on all mentioned datasets. Future work includes developing an exemplar-free framework to tackle issues related to large memory buffers and privacy concerns with stored food images.

References

[1] CJ Boushey, M Spoden, FM Zhu, EJ Delp, and DA Kerr, "New mobile methods for dietary assessment: review of image-assisted and image-based dietary assessment methods," *Proceedings of the Nutrition Society*, vol. 76, no. 3, pp. 283–294, 2017.

[2] Fengqing Zhu, Marc Bosch, Insoo Woo, SungYe Kim, Carol J Boushey, David S Ebert, and Edward J Delp, "The use of mobile devices in aiding dietary assessment and evaluation," *IEEE journal of selected topics in signal processing*, vol. 4, no. 4, pp. 756–766, 2010.

[3] Jiangpeng He, Runyu Mao, Zeman Shao, Janine L Wright, Deborah A Kerr, Carol J Boushey, and Fengqing Zhu, "An end-to-end food image analysis system," *Electronic Imaging*, vol. 2021, no. 8, pp. 285–1, 2021.

[4] Jiangpeng He, Zeman Shao, Janine Wright, Deborah Kerr, Carol Boushey, and Fengqing Zhu, "Multi-task image-based dietary assessment for food recognition and portion size estimation," *2020 IEEE Conference on Multimedia Information Processing and Retrieval*, pp. 49–54, 2020.

[5] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain, "A survey on food computing," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–36, 2019.

[6] Jianing Qiu, Frank P.-W. Lo, Yingnan Sun, Siyao Wang, and Benny P. L. Lo, "Mining discriminative food regions for accurate food recognition," *British Machine Vision Conference*, 2019.

[7] Runyu Mao, Jiangpeng He, Zeman Shao, Sri Kalyan Yarlagadda, and Fengqing Zhu, "Visual aware hierarchy based food recognition," *Proceedings of the International Conference on Pattern Recognition Workshop*, pp. 571–598, 2021.

[8] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith, "Learning to make better mistakes: Semantics-aware visual food recognition," *Proceedings of the 24th ACM international conference on Multimedia*, pp. 172–176, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[10] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool, "Food-101 – mining discriminative components with random forests," *Proceedings of the European Conference on Computer Vision*, 2014.

[11] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang, "Large scale visual food recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[12] Jiangpeng He, Luotao Lin, Heather Eicher-Miller, and Fengqing Zhu, "Long-tailed food classification," *arXiv preprint arXiv:2210.14748*, 2022.

[13] Michael McCloskey and Neal J Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," vol. 24, pp. 109–165. Elsevier, 1989.

[14] Jiangpeng He, *Continual Learning: Towards Image Classification From Sequential Data*, Ph.D. thesis, Purdue University, West Lafayette, IN, 2022.

[15] Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng, "Long-tailed class incremental learning," *European Conference on Computer Vision*, pp. 495–512, 2022.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean, "Distilling the knowledge in a neural network," *Proceedings of the NIPS Deep Learning and Representation Learning Workshop*, 2015.

[17] Eden Belouadah and Adrian Popescu, "Il2m: Class incremental learning with dual memory," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 583–592, 2019.

[18] Guo-Hua Wang, Yifan Ge, and Jianxin Wu, "Distilling knowledge by mimicking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8183–8195, 2021.

[19] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi, "The majority can help the minority: Context-rich minority oversampling for long-tailed classification," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6887–6896, 2022.

[20] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

[21] Keigo Kitamura, Toshihiko Yamasaki, and Kiyoharu Aizawa, "Food log by analyzing food images," *Proceedings of the 16th ACM international conference on Multimedia*, pp. 999–1000, 2008.

[22] Zeman Shao, Yue Han, Jiangpeng He, Runyu Mao, Janine Wright, Deborah Kerr, Carol Jo Boushey, and Fengqing Zhu, "An integrated system for mobile image-based dietary assessment," *Proceedings of the 3rd Workshop on AIxFood*, p. 19–23, 2021.

[23] S. Fang, Z. Shao, D. A. Kerr, C. J. Boushey, and F. Zhu, "An end-to-end image-based automatic food energy estimation technique based on learned energy distribution images: Protocol and methodology," *Nutrients*, vol. 11, no. 4, pp. 877, 2019.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[25] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, Honolulu, HI.

[27] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2014.

[28] Xin Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, “Recipe recognition with large multimodal food dataset,” *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–6, June 2015.

[29] Jingjing Chen and Chong-Wah Ngo, “Deep-based ingredient recognition for cooking recipe retrieval,” *Proceedings of the 24th ACM international conference on Multimedia*, pp. 32–41, 2016.

[30] Jingjing Chen, Bin Zhu, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang, “A study of multi-task and region-wise deep learning for food ingredient recognition,” *IEEE Transactions on Image Processing*, vol. 30, pp. 1514–1526, 2020.

[31] Javier Ródenas, Bhalaji Nagarajan, Marc Bolaños, and Petia Radeva, “Learning multi-subset of classes for fine-grained food recognition,” *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management*, pp. 17–26, 2022.

[32] Shuqiang Jiang, Weiqing Min, Yongqiang Lyu, and Linhu Liu, “Few-shot food recognition via multi-view representation learning,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3, pp. 1–20, 2020.

[33] Jixiang Gao, Jingjing Chen, Huazhu Fu, and Yu-Gang Jiang, “Dynamic mixup for multi-label long-tailed food ingredient recognition,” *IEEE Transactions on Multimedia*, 2022.

[34] Jiangpeng He and Fengqing Zhu, “Online continual learning for visual food classification,” *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 2337–2346, 2021.

[35] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al., “Balanced meta-softmax for long-tailed visual recognition,” *Advances in neural information processing systems*, vol. 33, pp. 4175–4186, 2020.

[36] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma, “Learning imbalanced datasets with label-distribution-aware margin loss,” *Advances in neural information processing systems*, vol. 32, 2019.

[37] Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano, “Experimental perspectives on learning from imbalanced data,” *Proceedings of the 24th international conference on Machine learning*, pp. 935–942, 2007.

[38] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural networks*, vol. 106, pp. 249–259, 2018.

[39] Shaoli Huang, Xinchao Wang, and Dacheng Tao, “Snapmix: Semantically proportional mixing for augmenting fine-grained data,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 1628–1636, 2021.

[40] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira, “Re-evaluating continual learning scenarios: A categorization and case for strong baselines,” *arXiv preprint arXiv:1810.12488*, 2018.

[41] Davide Maltoni and Vincenzo Lomonaco, “Continuous learning in single-incremental-task scenarios,” *Neural Networks*, vol. 116, pp. 56–73, 2019.

[42] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim, “Less-forgetting learning in deep neural networks,” *arXiv preprint arXiv:1607.00122*, 2016.

[43] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., “Overcoming catastrophic forgetting in neural networks,” *The National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[44] Zhizhong Li and Derek Hoiem, “Learning without forgetting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[45] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin, “Learning a unified classifier incrementally via rebalancing,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

[46] Max Welling, “Herding dynamical weights to learn,” *Proceedings of the International Conference on Machine Learning*, pp. 1121–1128, 2009.

[47] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert, “iCaRL: Incremental classifier and representation learning,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[48] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu, “Incremental learning in online scenario,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13926–13935, 2020.

[49] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari, “End-to-end incremental learning,” *Proceedings of the European Conference on Computer Vision*, 2018.

[50] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu, “Large scale incremental learning,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[51] U.S. Department of Agriculture and Agricultural Research Service, “What we eat in america, nhanes 2015-2016,” 2018, Accessed: 2024-08-10.

[52] Luotao Lin, Fengqing Zhu, Edward J Delp, and Heather A Eicher-Miller, “Differences in dietary intake exist among us adults by diabetic status using nhanes 2009–2016,” *Nutrients*, vol. 14, no. 16, pp. 3284, 2022.

[53] Centers for Disease Control and Prevention, “National diabetes statistics report 2020,” https://www.cdc.gov/diabetes/pdfs/data/statistics/national-diabetes-statistics-report.pdf.

[54] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, “Learning deep features for discriminative localization,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[55] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 618–626, 2017.

[56] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa, “Learning without memorizing,” *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5138–5146, 2019.

[57] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle, “Podnet: Pooled outputs distillation for small-tasks incremental learning,” *Proceedings of the European Conference on Computer Vision*, pp. 86–102, 2020.

[58] Jiangpeng He, “Gradient reweighting: Towards imbalanced class-incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16668–16677.

[59] Stuart A. Klugman, Harry H. Panjer, and Gordon E. Willmot, *Loss Models: From Data to Decisions*, John Wiley & Sons, 4th edition, 2012.

[60] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, “Mixup: Beyond empirical risk minimization,” *International Conference on Learning Representations*, 2018.

[61] Dan Hendrycks, Kimin Lee, and Mantas Mazeika, “Using pre-training can improve model robustness and uncertainty,” *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 2712–2721, 2019.

[62] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[63] Mingxing Tan and Quoc Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” *International conference on machine learning*, pp. 6105–6114, 2019.

[64] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *International Conference on Learning Representations*, 2021.

[65] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, “Training data-efficient image transformers & distillation through attention,” *International conference on machine learning*, pp. 10347–10357, 2021.

[66] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[67] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[68] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

Long-Tailed Continual Learning For Visual Food Recognition
Supplementary Material

VII. Extended Illustration of Datasets

A. Long-tailed Distribution

Long-tailed distribution is a scenario where New classes with imbalanced distribution arrive sequentially over time at each incremental learning phase. In a real-world food classification scenario, there are still two major challenges when applying food recognition method including (i) how to update the model when new food classes appear sequentially overtime, and (ii) how to address the severe class-imbalance issue since the real-life food images are usually in long-tailed distribution as shown in [12] where a minority of foods classes (*i.e. instance-rich or head classes*) are consumed more frequently than the remaining majority food classes (*i.e. instance-rare or tail classes*). The food recognition performance could drop dramatically without considering both two challenges.

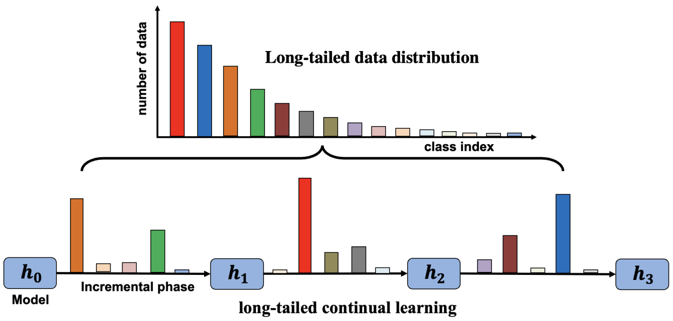

Fig. 7. The overview of long-tailed continual learning. The updated model should be able to learn new classes continuously and perform classification on all the classes seen so far.

As shown in Figure 7, an ideal food recognition system in the real world should be able to learn new foods incrementally in long-tailed distribution with class-imbalance training samples at each incremental phase.

B. The Selection of VFN186 Food Classes

The VFN186 dataset consists of 186 (these food classes of original VFN are included in 186 classes of VFN186) most frequently consumed foods (exclude drink and beverages) selected based on What We Eat In America (WWEIA) [1]. Following the similar process in [12], we first manually match each of the 186 food types in VFN [7] with one 8-digit USDA food codes from the Food and Nutrient Database for Dietary Studies (FNDDS)[2]. Each 8-digit USDA food code represents a specific food item reported and consumed in the food supply. Table VI shows matched food codes from FNDDS in the VFN dataset. Next, food codes were labeled with their corresponding consumption frequency as determined by Lin *et al* [52] using the nationally representative dietary data collected through the National Health and Nutrition Examination Survey (NHANES) [3] from 2009–2016 among U.S adults aged from 20 to 65.

VIII. Additional Experimental Results

In this part, we first present additional metric results for models across long-tailed datasets and then show the visualization of the top-1 classification accuracy. Finally, we compare our method with existing continual learning models in terms of the computation efficiency. All the experiment setting follows the same implementation setups in Section V-B.

A. Other Metric for Continual Learning Models

We report the last step accuracy A_L across five long-tailed datasets as seen in Figure VII. A_L is defined as the classification accuracy on the entire test set after learning all N tasks, which shows the model's ability to learn new knowledge while retaining previously acquired information. Table VII summarizes the results on Food101-LT, VFN-LT, VFN-INSULIN and VFN-T2D in terms of the last step accuracy A_L. We observe noticeable improvements by comparing with existing work on Food101-LT, VFN186-LT and VFN186-T2D. We achieve around 5% increase of A_L on VFN186-T2D compared with the latest released framework DGR **??**. On food101-LT, we outperform others with the number of tasks $N \in \{5, 10, 20\}$ for about 4%. Also, similar to the metric A_M, A_L will drop as the total number of tasks N increases since we need to maintain the learned knowledge at each learning phase of new tasks except for the first task.

B. Visualization for Models across all Datasets

We plot the results of top-1 classification accuracy evaluated on all the classes seen so far after learning each new task on three long-tailed datasets. Figure 8 shows the results of top-1 classification accuracy evaluated on all the classes seen so far after learning each new task. Our method achieves promising performance at each learning phase of the new task. Moreover, counterintuitive improvement is quite evident especially on Food101-LT with $N = 5$ after learning the third task. Our method's strong ability to learn new knowledge while retaining previously acquired information is well demonstrated across these datasets.

C. Computational Complexity

Regarding computational complexity, we recorded the time required for each model to be trained from scratch, including the time needed for testing. As seen in Figure VIII-C, there is a 15-minute reduction compared to the recently released

[1] https://data.nal.usda.gov/dataset/what-we-eat-america-wweia-database

[2] https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/food-surveys-research-group/docs/fndds-download-databases/

[3] https://www.cdc.gov/nchs/nhanes/index.html

TABLE VI
FOOD TYPES WITH MATCHED FOOD CODES AND WWEIA CATEGORY DESCRIPTIONS

Food Type	Food Code	Main food description from FNDDS	WWEIA Category description	Food Type	Food Code	Main food description from FNDDS	WWEIA Category description
Almond butter	42200500	Almond butter	Nuts and seeds	Green beans	75205030	Green beans, NS as to form, cooked	String beans
Almonds	42100100	Almonds, NFS	Nuts and seeds	Green peas	75120000	Green peas, raw	Other starchy vegetables
Apple	63101000	Apple, raw	Apples	Green salad	75114000	Mixed salad greens, raw	Lettuce and lettuce salads
Applesauce	63101110	Applesauce, regular	Apples	Grilled salmon	26137110	Salmon, cooked, NS as to cooking method	Fish
Asparagus	75202011	Asparagus, fresh, cooked, no added fat	Other vegetables and combinations	Grits	56201050	Grits, regular or quick, made with water, NS as to fat	Grits and other cooked cereals
Avocado	63105010	Avocado, raw	Other vegetables and combinations	Ground beef	63409010	Guacamole, NFS	Dips, gravies, other sauces
Bacon	22600200	Pork bacon, NS as to fresh, smoked or cured, cooked	Bacon	Guacamole	63409015	Guacamole with tomatoes	Dips, gravies, other sauces
Bagels	51180010	Bagel	Bagels and English muffins	Gyoza	58121620	Dumpling, vegetable	Turnovers and other grain-based items
Baked potato	71100100	Potato, baked, NFS	White potatoes, baked or boiled	Ham	22311000	Ham, smoked or cured, cooked, NS as to fat eaten	Cold cuts and cured meats
Bananas	63107010	Banana, raw	Bananas	Hashbrowns	71404000	Potato, hash brown, NFS	French fries and other fried white potatoes
BBQ meat	27120030	Ham or pork with barbecue sauce	Meat mixed dishes	Ice cream	13110000	Ice cream, NFS	Ice cream and frozen dairy desserts
Beans	41100990	Beans, NFS	Beans, peas, legumes	Jello	91501010	Gelatin dessert	Gelatins, ices, sorbets
Bean mixed dishes	58160110	Beans and white rice	Bean, pea, legume dishes	Kale	72119190	Kale, raw	Other dark green vegetables
Beef curry	27116100	Beef curry	Meat mixed dishes	Kiwi	63126500	Kiwi fruit, raw	Other fruits and fruit salads
Beef liver	25110140	Beef liver, fried	Liver and organ meats	Lamb chop	23101020	Lamb chop, NS as to cut, cooked, lean only eaten	Lamb, goat, game
Biscuits	52101000	Biscuit, NFS	Biscuits, muffins, quick breads	Lasagna	58130011	Lasagna with meat	Pasta mixed dishes, excludes macaroni and cheese
Blackberries	63201010	Blackberries, raw	Blueberries and other berries	Lettuce	75113000	Lettuce, raw	Lettuce and lettuce salads
Blueberries	63203010	Blueberries, raw	Blueberries and other berries	Lobster	26311160	Lobster, steamed or boiled	Shellfish
Boiled egg	31103010	Egg, whole, boiled or poached	Eggs and omelets	M&M	91746100	M&M's Milk Chocolate Candies	Candy containing chocolate
Bread stuffing	51182010	Bread stuffing	Turnovers and other grain-based items	Macaroni or noodles with cheese	58145110	Macaroni or noodles with cheese	Macaroni and cheese
Breaded fish	26100140	Fish, NS as to type, coated, fried, made with oil	Fish	Mango	63129010	Mango, raw	Mango and papaya
Broccoli	72201100	Broccoli, raw	Broccoli	Mashed potatoes	71501000	Potato, mashed, NFS	Mashed potatoes and white potato mixtures
Brownies	53204000	Cookie, brownie, NS as to icing	Cookies and brownies	Meat loaf	27214100	Meat loaf made with beef	Meat mixed dishes
Burgers	27510155	Cheeseburger, NFS	Burgers (single code)	Melons	63109010	Cantaloupe, raw	Melons
Burrito	58100160	Burrito with meat, beans, and rice	Burritos and tacos	Muffins	52301000	Muffin, NFS	Biscuits, muffins, quick breads
Cabbage	75105000	Cabbage, red, raw	Cabbage	Mushroom	75115000	Mushrooms, raw	Other vegetables and combinations
Cakes or cupcakes	53100100	Cake or cupcake, NS as to type	Cakes and pies	Nachos	58104180	Nachos with meat, cheese, and sour cream	Nachos
Candy	91745020	Hard candy	Candy not containing chocolate	Nectarine	63131010	Nectarine, raw	Peaches and nectarines
Carrots	73101010	Carrots, raw	Carrots	Noodles	56112000	Noodles, cooked	Pasta, noodles, cooked grains
Cashews	42104110	Cashews, unsalted	Nuts and seeds	Nutrition bars	53729000	Nutrition bar or meal replacement bar, NFS	Nutrition bars
Cauliflower	75107000	Cauliflower, raw	Other vegetables and combinations	Omelet	32129990	Egg omelet or scrambled egg, NS as to fat	Eggs and omelets
Celery	75109000	Celery, raw	Other vegetables and combinations	Onion	75117020	Onions, raw	Onions
Cereal	57100100	Cereal, ready-to-eat, NFS	Ready-to-eat cereal, higher sugar (¿21.2g/100g)	Orange	61119010	Orange, raw	Citrus fruits
Cereal bars	53712100	Cereal or Granola bar, NFS	Cereal bars	Orange chicken	27146350	Orange chicken	Stir-fry and soy-based sauce mixtures
Ceviche	27151030	Ceviche	Seafood mixed dishes	Other sandwiches	27500050	Sandwich, NFS	Other sandwiches (single code)
Cheese	14010000	Cheese, NFS	Cheese	Pakora	75440400	Pakora	Fried vegetables
Cheese corn snack	54401055	Cheese flavored corn snacks	Tortilla, corn, other chips	Pan dulce	51161250	Pan Dulce, no topping	Doughnuts, sweet rolls, pastries
Cheese sandwich	14640000	Cheese sandwich, NFS	Cheese sandwiches (single code)	Pancakes	55101000	Pancakes, plain	Pancakes, waffles, French toast
Cheesecake	53104500	Cheesecake	Cakes and pies	Papaya	63133010	Papaya, raw	Mango and papaya
Cherry	63115010	Cherries, raw	Other fruits and fruit salads	Pasta mixed dishes	56130000	Pasta, cooked	Pasta, noodles, cooked grains
Chicken breast	24120120	Chicken breast, NS as to cooking method, skin not eaten	Chicken, whole pieces	Peach	63135010	Peach, raw	Peaches and nectarines
Chicken curry	27146150	Chicken curry	Poultry mixed dishes	Peanut butter	42202000	Peanut butter	Nuts and seeds
Chicken nugget	24198729	Chicken nuggets, NFS	Chicken patties, nuggets and tenders	Peanut butter and jelly sandwiches	42302010	Peanut butter and jelly sandwich, NFS	Peanut butter and jelly sandwiches (single code)
Chicken or turkey salad	27446200	Chicken or turkey salad, made with mayonnaise	Poultry mixed dishes	Peanuts	42111100	Peanuts, roasted, salted	Nuts and seeds
Chicken tender	24198739	Chicken tenders or strips, NFS	Chicken patties, nuggets and tenders	Pear	63137010	Pear, raw	Pears
Chicken thigh	24150210	Chicken thigh, NS as to cooking method, skin eaten	Chicken, whole pieces	Pepper	75122100	Pepper, raw, NFS	Other vegetables and combinations
Chicken wing	24160110	Chicken wing, NS as to cooking method	Chicken, whole pieces	Pho	28310330	Pho	Soups
Chicken or turkey sandwiches	27540210	Chicken fillet wrap sandwich, fried, from fast food	Chicken/turkey sandwiches (single code)	Pies	53300100	Pie, NFS	Cakes and pies
Chocolate	91705300	Chocolate, sweet or dark	Candy containing chocolate	Pineapple	63141010	Pineapple, raw	Pineapple
Chow mein or chop suey	27343910	Chicken or turkey chow mein or chop suey with noodles	Fried rice and lo/chow mein	Pinto beans	41104020	Pinto beans, from dried, no added fat	Beans, peas, legumes
Cinnamon buns	51160100	Roll, sweet, cinnamon bun, no frosting	Doughnuts, sweet rolls, pastries	Pistachio nuts	42114140	Pistachio nuts, salted	Nuts and seeds
Coleslaw	75141000	Cabbage salad or coleslaw, made with coleslaw dressing	Coleslaw, non-lettuce salads	Pizza	58106230	Pizza, cheese, from restaurant or fast food, thick crust	Pizza
Collards	72107227	Collards, fresh, cooked with oil	Other dark green vegetables	Plantain	71900100	Plantain, cooked, no added fat	Other starchy vegetables
Cookies	53201000	Cookie, NFS	Cookies and brownies	Plum	63143010	Plum, raw	Other fruits and fruit salads
Corn	75216111	Corn, fresh, cooked, no added fat	Corn	Pomegranate	63145010	Pomegranate, raw	Other fruits and fruit salads
Corn dog	27560300	Corn dog, frankfurter or hot dog with cornbread coating	Frankfurter sandwiches (single code)	Popcorn	54403081	Popcorn, ready-to-eat packaged, plain	Popcorn
Cottage cheese	14200100	Cheese, cottage, NFS	Cottage/ricotta cheese	Popsicle	91611000	Popsicle	Gelatins, ices, sorbets
Crab	26305110	Crab, cooked, NS as to cooking method	Shellfish	Pork chop	22101000	Pork chop, NS as to cooking method, NS as to fat eaten	Pork
Crab cake	27250040	Crab cake	Seafood mixed dishes	Pork rib	22701000	Pork, spareribs, cooked, NS as to fat eaten	Pork
Crackers or saltines	54304000	Crackers, cheese	Crackers, excludes saltines	Potato chips	71200100	Potato chips, plain	Potato chips
Croissants	51166000	Croissant	Doughnuts, sweet rolls, pastries	Potato salad	71601010	Potato salad with egg, made with mayonnaise	Mashed potatoes and white potato mixtures
Cucumber	75111000	Cucumber, raw	Other vegetables and combinations	Pretzels or snack mix	54408017	Pretzels, hard, plain, salted	Pretzels/snack mix
Doughnuts	53521110	Doughnut, yeast type	Doughnuts, sweet rolls, pastries	Pudding	13230110	Pudding, flavors other than chocolate, ready-to-eat	Pudding
Dried fruits	62101050	Fruit mixture, dried	Dried fruits	Quesadilla	58104710	Quesadilla, just cheese, meatless	Other Mexican mixed dishes
Dumpling	58112510	Dumpling, steamed, filled with meat, poultry, or seafood	Egg rolls, dumplings, sushi	Quick breads	52201000	Cornbread, prepared from mix	Biscuits, muffins, quick breads
Edamame	41420020	Edamame, cooked	Beans, peas, legumes	Quinoa	56204005	Quinoa, no added fat	Pasta, noodles, cooked grains
Egg roll	58110110	Egg roll, meatless	Egg rolls, dumplings, sushi	Radish	75125000	Radish, raw	Other vegetables and combinations
Egg or breakfast sandwiches	27560670	Sausage and cheese on English muffin	Egg/breakfast sandwiches (single code)	Raspberries	63219000	Raspberries, raw	Blueberries and other berries
English muffin	51186010	Muffin, English	Bagels and English muffins	Sashimi	26137100	Salmon, raw	Fish
Falafel	41209000	Falafel	Bean, pea, legume dishes	Sausages	25221400	Sausage, NFS	Sausages
Fish sandwiches	27550000	Fish sandwich, fried, from fast food	Seafood sandwiches (single code)	Shrimp	27150110	Shrimp cocktail	Shellfish
Frankfurter sandwich	25210110	Frankfurter or hot dog, NFS	Frankfurters	Smoked salmon	26137190	Salmon, smoked	Fish
French fries	71400990	Potato, french fries, NFS	French fries and other fried white potatoes	Soup	58400000	Soup, NFS	Soups
French toast	55301000	French toast, plain	Pancakes, waffles, French toast	Spanish rice	58163420	Spanish rice, no added fat	Rice mixed dishes
Fried egg	31105005	Egg, whole, fried, NS as to fat	Eggs and omelets	Spinach	72125100	Spinach, raw	Spinach
Fried mushrooms	75414030	Fried mushrooms	Fried vegetables	Squid	26213140	Squid, coated, fried	Shellfish
Fried okra	75414500	Fried okra	Fried vegetables	Steak	21101000	Beef steak, NS as to cooking method, NS as to fat eaten	Beef, excludes ground
Fried rice	58150310	Rice, fried, NFS	Fried rice and lo/chow mein	Stew beef	27211200	Beef stew with potatoes, gravy	Meat mixed dishes
Fried sweet potato	73410400	Sweet potato fries, fast food / restaurant	Fried vegetables	Strawberries	63223020	Strawberries, raw	Strawberries
Fruit salad	63311000	Fruit salad, fresh or raw, excluding citrus fruits, no dressing	Other fruits and fruit salads	Summer squash	75233027	Summer squash, yellow or green, fresh, cooked with oil	Other vegetables and combinations
Graham crackers	54102010	Graham crackers	Cookies and brownies	Sushi	58151100	Sushi, NFS	Egg rolls, dumplings, sushi
Grapes	63123000	Grapes, raw	Grapes	Sweet potato	73401000	Sweet potato, NFS	Other red and orange vegetables
Taco or tostada	58101320	Taco or tostada with meat	Burritos and tacos	Turkey	24201000	Turkey, NFS	Turkey, duck, other poultry
Tamale	58103120	Tamale with meat	Other Mexican mixed dishes	Turnover	58126150	Turnover, meat- and cheese-filled, tomato-based sauce	Turnovers and other grain-based items
Tangerine	61125010	Tangerine, raw	Citrus fruits	Vegetable chips	71220000	Vegetable chips	Tortilla, corn, other chips
Tilapia	26158013	Tilapia, baked or broiled, no added fat	Fish	Vegetable curry	75440600	Vegetable curry	Vegetable dishes
Tofu	41420010	Soybean curd	Processed soy products	Waffles	55200200	Waffle, plain, from fast food / restaurant	Pancakes, waffles, French toast
Tomatoes	74101000	Tomatoes, raw	Tomatoes	Walnuts	42116000	Walnuts, excluding honey roasted	Nuts and seeds
Tortilla and corn chips	54401075	Tortilla chips, plain	Tortilla, corn, other chips	White and brown rice	56205000	Rice, cooked, NFS	Rice
Tortillas	52215100	Tortilla, corn	Tortillas	Whole chicken	24100000	Chicken, NS as to part and cooking method, NS as to skin eaten	Chicken, whole pieces
Trail mix	42110160	Mixed nuts, without peanuts, unsalted	Nuts and seeds	Wonton soup	58408010	Wonton soup	Soups
Tuna salad	27450060	Tuna salad, made with mayonnaise	Seafood mixed dishes	Yeast breads	51000100	Bread, NS as to major flour	Yeast breads
Tuna sandwiches	27550720	Tuna salad sandwich, on bread	Seafood sandwiches (single code)	Yogurt	11434010	Yogurt, NS as to type of milk or flavor	Yogurt, regular

DGR [58] method, which requires 84 minutes, but it achieves superior classification performance, making it both time-efficient and effective. iCaRL [47] stands out as the fastest but at the cost of lower classification accuracy, particularly in handling datasets of long-tailed distribution. Three two-stage methods show robust classification accuracy but at the expense of significantly higher training times, which are all over three times to our approach's. Our method strikes an optimal balance between computational efficiency and classification accuracy, outperforming all other methods when considering both aspects, which makes it particularly suitable for real-world applications.

TABLE VII
RESULTS ON FOOD101-LT, VFN-LT, VFN186-LT, VFN186-INSULIN, AND VFN186-T2D BY COMPARING WITH EXISTING CONTINUAL LEARNING METHODS IN TERMS OF LAST STEP ACCURACY (A_L). BEST RESULTS ARE MARKED IN BOLD.

Datasets	**Food101-LT**			**VFN-LT**	**VFN186-LT**	**VFN186-INSULIN**	**VFN186-T2D**
Number of tasks	$N = 5$	$N = 10$	$N = 20$	$N = 7$	$N = 9$	$N = 9$	$N = 9$
LwF [44]	8.62	5.86	0.83	1.02	10.03	8.99	9.78
EWC [43]	4.29	3.70	0.83	1.58	8.95	8.85	2.75
iCaRL [47]	11.48	12.46	11.04	11.84	7.34	5.64	6.19
EEIL [49]	12.88	10.57	6.98	16.87	9.80	9.27	9.84
LwM [56]	10.62	7.22	2.45	7.41	9.57	9.31	9.14
IL2M [17]	12.55	10.97	6.81	14.87	9.51	9.34	10.15
BiC [50]	14.94	12.39	10.38	15.40	7.82	7.83	8.58
LUCIR [45]	13.87	10.17	6.74	16.31	11.40	9.62	10.14
PODNet [57]	12.04	10.46	8.99	14.75	10.44	6.09	10.08
EEIL-2stage [15]	14.16	13.29	9.76	17.61	10.44	9.59	10.91
LUCIR-2stage [15]	16.34	13.03	10.85	17.33	11.75	9.98	10.74
PODNet-2stage [15]	13.94	11.12	10.28	17.14	10.96	10.30	10.08
DGR [58]	26.70	26.90	21.13	26.14	22.39	20.34	19.26
Ours	**30.98**	**29.89**	**24.2**	**28.36**	**25.58**	**22.86**	**24.14**

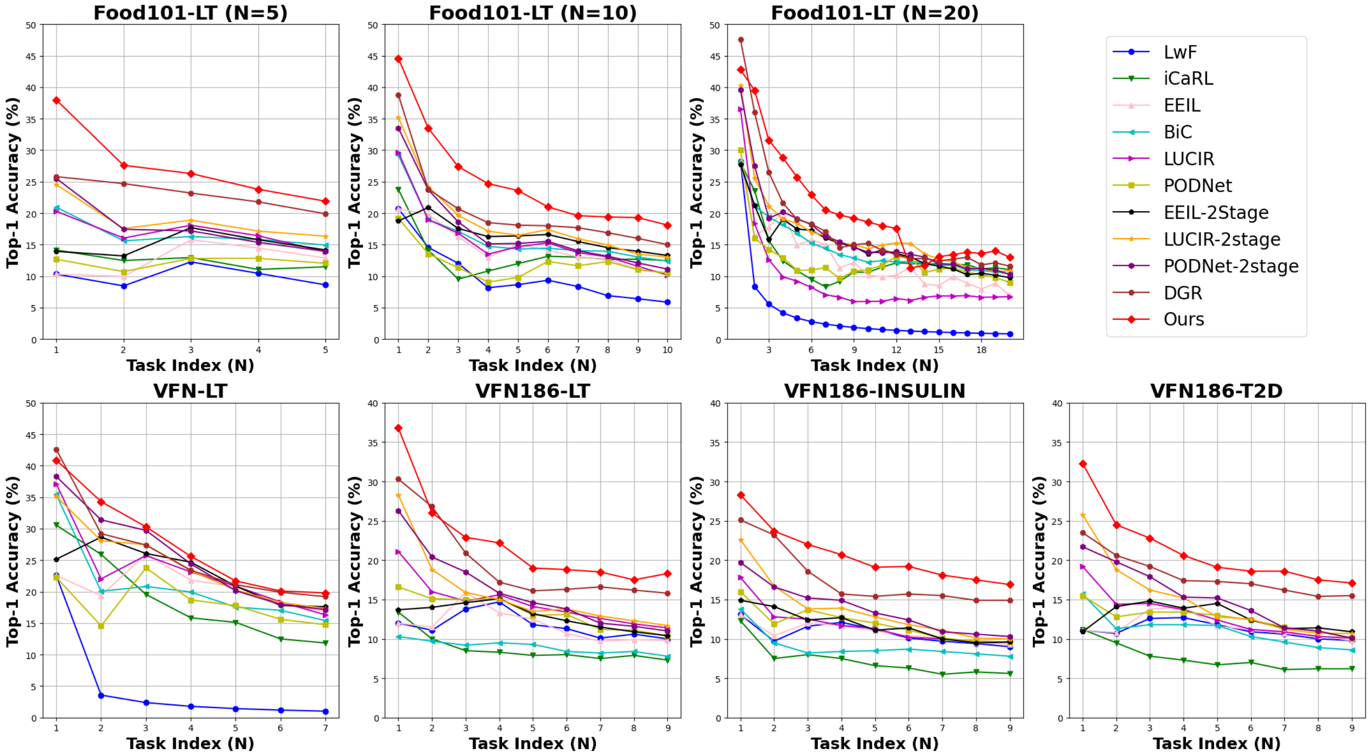

Fig. 8. Results on Food101-LT, VFN-LT, VFN186-LT, VFN186-INSULIN and VFN186-T2D with different number of tasks N. Each marker represents the Top-1 classification accuracy evaluated on all classes seen so far after learning each task.

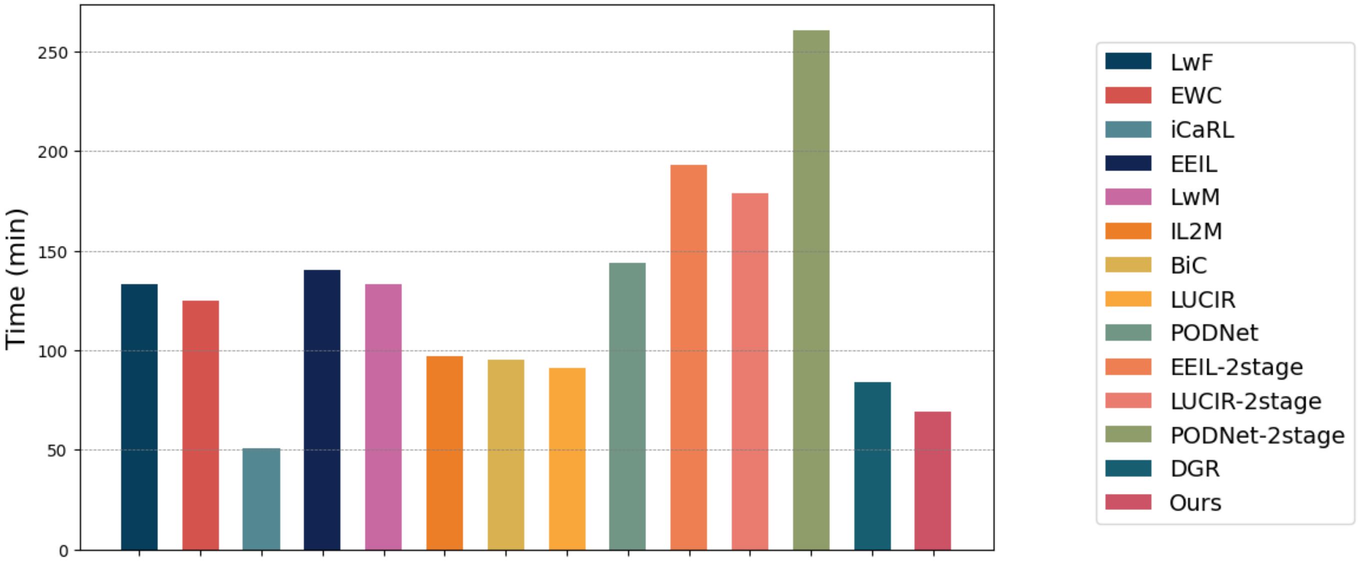

Fig. 9. Running time (min) comparison on VFN186-LT for different models.