CONTINUAL LEARNING: TOWARDS IMAGE CLASSIFICATION FROM SEQUENTIAL DATA

by

Jiangpeng He

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Electrical and Computer Engineering

West Lafayette, Indiana

August 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Fengqing M. Zhu

School of Electrical and Computer Engineering

Dr. Amy R. Reibman

School of Electrical and Computer Engineering

Dr. Stanley H. Chan

School of Electrical and Computer Engineering

Dr. Heather A. Eicher-Miller

School of Nutrition Science

Approved by:

Dr. Dimitrios Peroulis

To my dear wife, ***Chuqin*** *and my son,* ***Calix***

ACKNOWLEDGMENTS

I would like to thank my doctoral advisor Professor Fengqing M. Zhu for her support and guidance throughout my entire PhD life. Prof. Zhu taught me not only the knowledge to support my future career, but also the problem-solving ability to help me solve the obstacles in my future life. Besides, she also gave me many invaluable suggestions for my personal life. I really appreciate the help from Prof. Zhu and I am really proud and grateful to have this experience working with her. I would also like to thank Professor Amy R. Reibman for her help on my research to give me suggestions regarding my research projects. I would like to thank Professor Stanley H. Chan for his insightful feedback and suggestions on my research, especially, for the knowledge I learned from his Machine Learning course. I would like to thank Professor Heather A. Eicher-Miller for her help and support on my research as well. Professor Eicher-Miller gave me a lot of insights about nutrition science, which builds the foundation on many of my current and future research work.

Besides the help from my PhD committee, I would also like to thank all the former and current members of the Video and Image Processing Laboratory (VIPER) for their help and companionship, especially, Dr. Sri Kalyan Yarlagadda, Dr. Shaobo Fang, Dr. Qingshuang Chen, Dr. Di Chen, Sriram Baireddy, Emily Bartusiak, János Horváth, Ruiting Shao, Justin Yang, Enyu Cai, Jiaqi Guo, Changye Yang, Yue Han, Liming Wu, Alain Chen, Ziyue (Alan) Xiang, Runyu Mao, Zeman Shao, Xinyue Pan, Siddeshwar Raghavan, Zhihao Duan, Gautham Vinod, Yuning Huang, Weichen Xu, Yezhi Shen, Xiaoyu Ji.

Finally, I would like to thank my family, especially, my wife Chuqin Zhao, my son Calix He, and also the unconditional love and support from my mother and father. Their love and support make all of the impossible possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Though modern deep learning based approaches have achieved remarkable progress in computer vision community such as image classification using a static image dataset, it suffers from catastrophic forgetting when learning new classes incrementally in a phase-by-phase fashion, in which only data for new classes are provided at each learning phase. In this work we focus on continual learning with the objective of learning new tasks from sequentially available data without forgetting the learned knowledge. We study this problem from three perspectives including (1) continual learning in online scenario where each data is used only once for training (2) continual learning in unsupervised scenario where no class label is provided and (3) continual learning in real world applications. Specifically, for problem (1), we proposed a variant of knowledge distillation loss together with a two-step learning technique to efficiently maintain the learned knowledge and a novel candidates selection algorithm to reduce the prediction bias towards new classes. For problem (2), we introduced a new framework for unsupervised continual learning by using pseudo labels obtained from cluster assignments and an efficient out-of-distribution detector is designed to identify whether each new data belongs to new or learned classes in unsupervised scenario. For problem (3), we proposed a novel training regime targeted on food images using balanced training batch and a more efficient exemplar selection algorithm. Besides, we further proposed an exemplar-free continual learning approach to address the memory issue and privacy concerns caused by storing part of old data as exemplars.

In addition to the work related to continual learning, we study the image-based dietary assessment with the objective of determining what someone eats and how much energy is consumed during the course of a day by using food or eating scene images. Specifically, we proposed a multi-task framework for simultaneously classification and portion size estimation by future fusion and soft-parameter sharing between backbone networks. Besides, we introduce RGB-Distribution image by concatenating the RGB image with the energy distribution map as the fourth channel, which is then used for end-to-end multi-food recognition and portion size estimation.

.

1. INTRODUCTION

1.1 Continual Learning For Image Classification

One of the major open challenges towards artificial intelligence is learning new knowledge incrementally. For image classification task, instead of training a model to classify all classes in a static image datasets such as ImageNet [1] and CIFAR [2], the model needs to learn from sequentially available data where new classes are continuously added overtime. For image classification task, instead of training a model to classify all classes in static datasets such as ImageNet [1] and CIFAR [2], the model needs to learn from sequentially available data where new classes are continuously added overtime. However, existing deep learning based methods suffer from catastrophic forgetting [3], a phenomenon where the performance on the learned classes degrades dramatically as new classes are added due to the unavailability of the training data for learned classes. The objective of continual learning, also known as incremental learning and lifelong learning, is to learn new tasks from sequential data without forgetting the learned knowledge. A more general perspective of continual learning is the stability-plasticity dilemma [4] where the stability refers to preserving the knowledge for learned tasks and plasticity means the adaptation to the new knowledge. Therefore, the continual learning aims to strike the balance between stability and plasticity.

Continual learning has been studied under different learning scenarios. In general, it can be divided into (1) task-incremental, (2) class-incremental, and (3) domain-incremental as discussed in [5]. For task-incremental learning, the model learns disjoint classifier heads where each classifier corresponds to one independent task, which is also known as multi-head classifier [6]. In this case, the task-IDs are provided during both training and inference phases. For class-incremental learning, the model learns only one single classifier head used for all tasks, which is known as the single-head classifier [7]. In this configuration, the task-ID is not available during inference phase and the model needs to classify for all classes seen so far without using task-ID. For domain-incremental learning, the output label space keeps unchanged while the input data distributions are changing over time. Therefore, no new classes is added in this case and the model should adapt to the most recent distribution of input data. In addition, depending on whether each data is used more than once to update

the model, it can be categorized as (1) online learning that use each data once, and (2) offline learning with no epoch restriction. Finally, there are supervised and unsupervised continual learning depending on whether the class labels or human annotations are provided during the training phase. In this work, we focus on class-incremental setting to use one single-head classifier for all learned classes and study this problem under both online and unsupervised scenarios. Figure 1.1 illustrates an example of continual learning under class-incremental setting for food image classification.

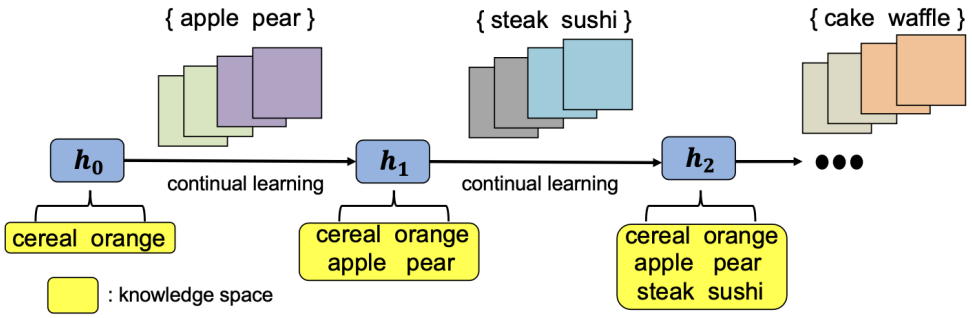


Figure 1.1. Continual learning for food image classification under class-incremental setting. The model h learns new food class sequentially overtime without accessing to already learned class data for each continual learning step. The updated model can classify all food classes seen so far.

1.1.1 Online Continual Learning

Continual learning methods under online and class-incremental setting use each data once to update the model and employs a single-head classifier [7] to test on all classes encountered so far during inference phase. This setting is more closer to real life learning environment where new classes come in as data streams with limited adaptation time and storage capacity allowed for processing [8]. For example, an on-device image recognition system should be able to update using each new captured image continually without forgetting the classes learned so far. Figure 1.2 illustrates the difference between online and offline continual learning under class-incremental setting.

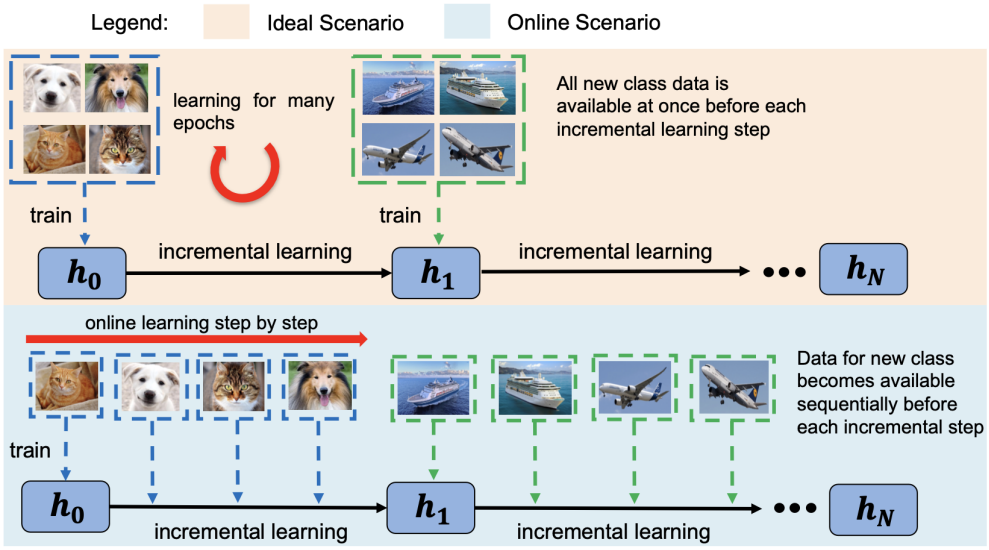


Figure 1.2. Main difference between existing class-incremental learning methods in ideal offline scenario and online scenario, h_{i} denotes the updated model after incremental step i and N is the total number of incremental steps.

Unfortunately, the catastrophic forgetting becomes more severe in online scenario due to limited run-time and data allowed to update the model and also the continual learning in online scenario is not well-studied compared with in offline setting. In this work, we address this problem from two perspectives: (i) adding stronger regularization term and (ii) reducing the output bias.

Knowledge distillation loss [9] is widely applied in offline continual learning to address catastrophic forgetting by regularizing the change of parameters related to learned classes. However, its effectiveness degrades in online scenario as each data is used only once, *i.e.*, we need stronger regularization for online scenario. In this work, we propose a modified knowledge distillation loss together with a two-step learning technique and achieved competitive performance compare with results in offline continual learning.

In addition, for class-incremental methods using a single-head classifier, the prediction result is always associated with the largest value of output logits. However, during continual learning, the output logits become biased towards new task due to the unavailability of old task data [10], *i.e.*, the output logits of new task are much larger than those of old tasks.

This results in the corresponding biased prediction on new tasks, which is a significant contributing factor for catastrophic forgetting.

1.1.2 Unsupervised Continual Learning

Unsupervised continual learning is an emerging future learning system, capable of learning a sequence of new tasks incrementally from unlabeled data. It requires neither static datasets nor human annotations compared with supervised offline learning. However, the problem becomes more challenging as we need to learn the knowledge from unlabeled sequential data while maintaining the knowledge for learned classes to address catastrophic forgetting. On the other hand, pseudo label [11] is widely applied in both semi-supervised and unsupervised learning to handle unlabeled data for downstream tasks, which is effective due to its simplicity, generality and ease of implementation. However, whether it is feasible for continual learning to rely on pseudo labels instead of human annotations is not well explored yet. In this work, we introduce a novel framework for unsupervised continual learning by using pseudo labels obtained from cluster assignments as shown in Figure 1.3. K-means [12] is adopted as our global clustering algorithm for illustration purpose and we propose to use the continual learning model (except the last fully connected layers) at every incremental step for feature extraction of unlabeled data to obtain pseudo label.

Besides, most existing unsupervised continual learning methods, especially those targeted on image classification, only work in a simplified scenario by assuming all new data belong to new tasks. We argue that if human annotation is not available as common in unsupervised scenario, we cannot know whether the unlabeled new data belongs to new or learned tasks. For example, an image recognition system should be able to distinguish new and learned classes at first instead of blindly treating all of them as new classes to perform unsupervised continual learning for update. Therefore, in order to make unsupervised continual learning work in practical problems, an out-of-distribution (OOD) detector should be required at the beginning of each incremental learning step to identify whether each data belongs to new or already learned tasks. However, the problem of OOD detection in continual learning still remains under-explored, *i.e.* none of the existing OOD detection methods target for continual

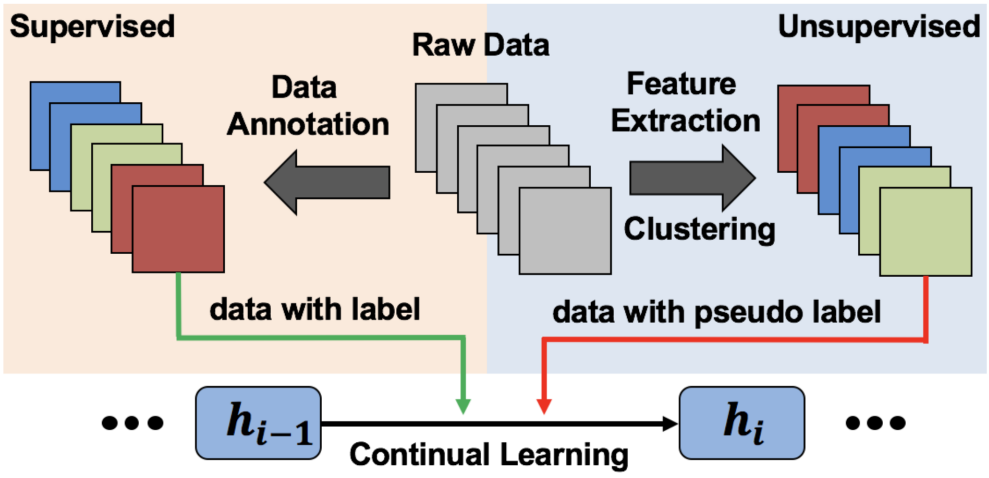


Figure 1.3. Supervised vs. unsupervised continual learning for learning a new task i, h refers to the model in different incremental steps. The supervised and our proposed pseudo label based unsupervised continual learning are illustrated by green and red arrows respectively.

learning. In this work, we further formulate the OOD detection in unsupervised continual learning scenario denoted as OOD-UCL and introduce the corresponding evaluation protocol. Then, we propose a novel OOD detection method to achieve improved performance in unsupervised continual learning scenario.

1.1.3 Application Based Continual Learning

Though recent continual learning work has achieved remarkable progress towards image classification task, most existing methods work only work for balanced dataset where each class has the same number of images, which rarely happens in real world. Therefore, instead of using common datasets such as CIFAR [2] or ImageNet[1] for experiments as in most existing methods, we focus on applying continual learning in real world challenging food dataset [13] containing 1,000 food classes with unbalanced number of data for each class. In addition, the food data exhibits higher intra-class variation [14] compared with commonly seen objects in real life due to different culinary culture and cooking style. In this work, we address this challenging problem of food image classification for continual learning by

introducing a novel training regime with balanced training batch of old and new classes data.

Besides, the success of most existing methods relies on storing part of learned task data or feature representation as exemplars for knowledge replay to address catastrophic forgetting. We argue that there are two major issues associated with using exemplars when applied in real life applications: (i) the performance is greatly relied on the size of stored exemplars while the storage consumption is a significant constraint in continual learning, (ii) storing exemplars may not always be feasible for certain applications due to privacy concerns such as medical or health research. In this work, we further propose a novel exemplar-free method by leveraging nearest-class-mean (NCM) classifier where the class mean is estimated during training phase on all data seen so far through online mean update criteria.

1.2 Image-Based Dietary Assessment

Dietary assessment is the process of determining what someone eats and how much energy is consumed during the course of a day. It provides valuable insights for mounting intervention programs for prevention of many chronic diseases. Modern deep learning techniques have achieved great success in image-based dietary assessment for food localization and classification [14]–[21], as well as food portion size estimation [20], [22]–[27]. However, existing methods only focus on one task at one time, which makes it challenging to integrate into a complete system for fast and streamlined process. In this work, we focus on designing end-to-end integrated food analysis system for both single-item and multi-item food images.

Specifically for single-item food images, we target on classification and portion size estimation tasks. In image-based dietary assessment, it is important to monitor and record what kind of food people eat for disease prevention by performing image classification. However, estimating an object's portion size is a challenging task. An object's portion size is defined as the numeric value that is directly related to the spatial quantity of the object in world coordinates. Examples may include an object's volume and weight, as $weight \propto volume$ ($weight = volume \times density$). In food portion size estimation, we want to estimate food energy ($food\ energy \propto food\ volume$, as $food\ energy = food\ volume \times unit\ volume\ energy$)

from an input image since energy intake is an important indicator for healthy eating. In this work, we introduce a multi-task framework by using L2-norm based soft parameter sharing to train the classification and portion estimation tasks simultaneously. We also propose the use of cross-domain feature adaptation together with normalization to further improve the performance of food portion size estimation.

For multi-item food images, we further include food localization to locate each individual food region for a given image with a bounding box. Pixels within the bounding box are assumed to represent a single food, which is the input to the food classification task. We proposed to improve the portion size estimation performance by incorporating the food localization results to obtain four-channel RGB-Distribution food images used for regression task, where the individual energy distribution map is generated by using conditional GAN [28].

Success of modern deep learning based methods also rely on the availability of data. The lack of good datasets have resulted limited progress end-to-end image-based dietary assessment system. Currently, there is no available food image dataset that includes both food category and corresponding portion size since it is difficult to obtain accurate food energy from the crowd based annotation on RGB images, unless these numeric values are recorded during image collection. In this work, we introduce an eating occasion datasets containing both food category and food portion size provided by registered dietitians. Figure 1.4 shows an example of multi-item eating occasion images.

1.3 Contributions Of This Thesis

In this thesis, we proposed new methods to address catastrophic forgetting targeted on online continual learning, unsupervised continual learning and application based continual learning. Besides, we designed an end-to-end integrated food analysis system and introduce novel portions size estimation method for image-based dietary assessment. The main contributions are listed as follows:

- Online Continual Learning
  - We introduce a modified cross-distillation loss together with a two-step learning technique to address catastrophic forgetting in online scenario.

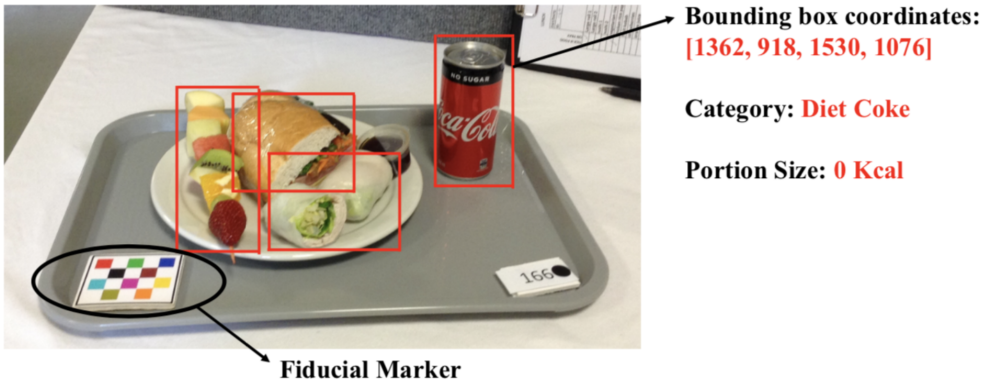


Figure 1.4. Example of an eating occasion image in our dataset. Each food item is manually cropped containing the groundtruth bounding box coordinates and food category. All food and beverages were pre-weighed. A fiducial marker [29] is used to calibrate the color and size of the input image

  - A continual learning framework is proposed, which is capable of lifelong learning and can be applied to a variety of real life online image classification problems where new data can belong to both new or learned class. We provide a simple yet effective method to mitigate concept drift by updating the exemplar set using the feature of each new observation of old classes.
  - Instead of using original data exemplars, we propose a simple yet effective method to store feature embeddings to reduce the memory burden and an online sampler is designed to select exemplars from sequentially available data stream through dynamic mean update criteria.
  - A novel candidates selection algorithm is introduced to mitigate forgetting in online scenario by reducing the output bias.

- Unsupervised Continual Learning

  - We explore a novel problem for continual learning using pseudo labels instead of human annotations, which is under-studied yet and a new benchmark evaluation protocol for unsupervised continual learning is introduced for future research work.

  - An unsupervised continual learning framework is proposed by using pseudo labels obtained from cluster assignments, which can be easily adapted by existing supervised continual learning techniques and we achieve competitive performance with supervised method but without human annotation.
  - We formulate the problem and proposed the corresponding evaluation protocol for out-of-distribution detection in unsupervised continual learning (OOD-UCL), which remains under-explored.
  - A novel method is introduced for OOD detection by correcting output bias and enhancing output confidence difference based on task discriminativeness.

- Application Based Continual Learning
  - To the best of our knowledge, we are the first to study online continual learning for food image classification task. We proposed a novel clustering based exemplar selection algorithm and a new online training regime to address catastrophic forgetting.
  - We proposed a novel exemplar-free online continual learning method by leveraging NCM classifier with class mean estimated on all data seen so far to reduce the memory burden and address privacy concerns in real life applications.

- Image-Based Dietary Assessment
  - We introduce a food image datasets collected from a nutrition study with the groundtruth food portion provided by registered dietitians.
  - A soft-parameter sharing multi-task framework is introduced for single-item food image analysis, which is capable of simultaneously food classification and portion size estimation.
  - We proposed to use four-channel RGB-Distribution food images and introduce an end-to-end food analysis system for multi-item food images by integrating localization, classification and portion size estimation.

1.4 Publications Resulting From This Thesis

1. **Jiangpeng He**, Runyu Mao, Zeman Shao, Fengqing Zhu, "Incremental Learning In Online Scenario," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2020, Virtual Conference.

2. **Jiangpeng He**, Zeman Shao, Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu, "Multi-Task Image-Based Dietary Assessment For Food Recognition And Portion Size Estimation," Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval, August 2020, Virtual Conference.

3. **Jiangpeng He**, Runyu Mao, Zeman Shao, Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu, "An End-to-End Food Image Analysis System," Electronic Imaging, January 2021, Virtual Conference.

4. **Jiangpeng He**, Fengqing Zhu, "Unsupervised Continual Learning Via Pseudo Labels," International Joint Conference on Artificial Intelligence, CSSL Workshop, August 2021, Virtual Conference.

5. **Jiangpeng He**, Fengqing Zhu, "Online Continual Learning For Visual Food Classification," Proceedings of the IEEE International Conference on Computer Vision, Large Fine Food AI Workshop, October 2021, Virtual Conference.

6. **Jiangpeng He**, Fengqing Zhu, "Online Continual Learning Via Candidates Voting," Proceedings of the IEEE Winter Conference on Applications of Computer Vision, January 2022, Hawaii.

7. **Jiangpeng He**, Fengqing Zhu, "Out-Of-Distribution Detection In Unsupervised Continual Learning," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Continual AI Workshop, June 2022, New Orleans.

8. **Jiangpeng He**, Fengqing Zhu, "Exemplar-Free Online Continual Learning," Proceedings of the IEEE International Conference on Image Processing, October 2022, Bordeaux, France.

1.5 Other Publications Not Related to This Thesis

1. **Jiangpeng He**, Kyle Ziga, Judy Bagchi, Fengqing Zhu, "CNN Based Parameter Optimization for Texture Synthesis," Electronic Imaging, January 2019, San Francisco.

2. Runyu Mao, **Jiangpeng He**, Zeman Shao, Sri Yarlagadda, Fengqing Zhu, "Visual Aware Hierarchy Based Food Recognition," Proceedings of International Conference on Pattern Recognition, Workshops and Challenges, January 2021, Virtual Conference.

3. Zeman Shao, Shaobo Fang, Runyu Mao, **Jiangpeng He**,Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu, "Towards Learning Food Portion From Monocular Images With Cross-Domain Feature Adaptation," Proceedings of IEEE 23rd International Workshop on Multimedia Signal Processing, October 2021, Virtual Conference.

4. Runyu Mao, **Jiangpeng He**, Luotao Lin, Zeman Shao, Heather Eicher-Miller, Fengqing Zhu, "Improving Dietary Assessment Via Integrated Hierarchy Food Classification," Proceedings of IEEE 23rd International Workshop on Multimedia Signal Processing, October 2021, Virtual Conference.

5. Zeman Shao, Yue Han, **Jiangpeng He**, Runyu Mao, Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu, "An Integrated System for Mobile Image-Based Dietary Assessment," Proceedings of ACM International Multimedia Conference, Workshop of AIxFood, October 2021, Virtual Conference.

6. Zeman Shao, **Jiangpeng He**, Ya-Yuan Yu, Luotao Lin, Alexandra Cowan, Heather Eicher-Miller, Fengqing Zhu, "Towards the Creation of a Nutrition and Food Group Based Image Database," Electronic Imaging, January 2022, Virtual Conference.

2. ONLINE CONTINUAL LEARNING

2.1 Continual Learning In Online Scenario

2.1.1 Overview

One of the major challenges of current deep learning based methods when applied to real life applications is learning new classes incrementally, where new classes are continuously added overtime. Furthermore, in most real life scenarios, new data comes in sequentially, which may contain both the data from new classes or new observations of old classes. Therefore, a practical vision system is expected to handle the data streams containing both new and old classes, and to process data sequentially in an online learning mode [30], which has similar constrains as in real life applications. For example, a food image recognition system designed to automate dietary assessment should be able to update using each new food image continually without forgetting the food categories already learned.

Most deep learning approaches trained on static datasets suffer from the following issues. First is catastrophic forgetting [3], a phenomenon where the performance on the old classes degrades dramatically as new classes are added due to the unavailability of the complete previous data. This problem become more severe in online scenario due to limited run-time and data allowed to update the model. The second issue arises in real life application where the data distribution of already learned classes may change in unforeseen ways [31], which is related to concept drift [32]. In this work, we aim to develop an incremental learning framework that can be deployed in a variety of image classification problems and work in the challenging online learning scenario.

A practical deep learning method for classification is characterized by (1) its ability to be trained using data streams including both new classes data and new observations of old classes, (2) good performance for both new and old classes on future data streams, (3) short run-time to update with constrained resources, and (4) capable of lifelong learning to handle multiple classes in an incremental fashion. Although progress has been made towards reaching these goals, none of the existing approaches for incremental learning satisfy all the above conditions. They assume the distribution of old classes data remain unchanged overtime and consider only new classes data for incoming data streams. As we mentioned earlier, data

distribution are likely to change in real life[31]. When concept drift happens, regardless the effort put into retaining the old classes knowledge, degradation in performance is inevitable. In addition, although these existing methods have achieved state-of-the-art results, none of them work in the challenging online scenario. They require offline training using all available new data for many epochs, making it impractical for many real life scenarios.

2.1.2 Related Work

Incremental learning remains one of the long-standing challenges for machine learning, yet it is very important to brain-like intelligence capable of continuously learning and knowledge accumulation through its lifetime.

Traditional methods. Prior to deep learning, SVM classifier [33] is commonly used. One representative work is [34], which learns the new decision boundary by using support vectors that are learned from old data together with new data. An alternative method is proposed in [35] by retaining the Karush-Kuhn-Tucker conditions instead of support vectors on old data and then update the solution using new data. Other techniques [36]–[38] use ensemble of weak classifiers and nearest neighbor classifier.

Deep learning based methods. These methods provide a joint learning of task-specific features and classifiers. Approaches such as [39], [40] are based on constraining or freezing the weights in order to retain the old tasks performance. In [39], the last fully connected layer is freezed which discourages change of shared parameters in the feature extraction layers. Inn [40] old tasks knowledge is retained by constraining the weights that are related to these tasks. However, constraining or freezing parameters also limits its adaptability to learn from new data. A combination of knowledge distillation loss [9] with standard cross-entropy loss is proposed to retain the old classes knowledge in [41], where old and new classes are separated in multi-class learning and distillation is used to retain old classes performance. However, performance is far from satisfactory when new classes are continuously added, particularly in the case when the new and old classes are closely related. Based on [41], auto encoder is used to retain the knowledge for old classes instead of using distillation loss in [42]. For all these methods, only new data is considered.

In [43] and [44], synthetic data is used to retain the knowledge for old classes by applying a deep generative model [45]. However, the performance of these methods are highly dependent on the reliability of the generative model, which struggles in more complex scenarios.

Rebuffi et al proposed iCaRL[46], an approach using a small number of exemplars from each old class to retain knowledge. An end-to-end incremental learning framework is proposed in [47] using exemplar set as well, along with data augmentation and balanced fine-tuning to alleviate the imbalance between the old and new classes. Incremental learning for large datasets was proposed in [10] in which a linear model is used to correct bias towards new classes in the fully connected layer. However, it is difficult to apply these methods to real life applications since they all require a long offline training time with many epochs at each incremental step to achieve a good performance. In addition, they assume the distribution of old classes remain unchanged and only update the classifiers using new classes data.

All in all, a modified cross-distillation loss along with a two-step learning technique is introduced to make incremental learning feasible in the challenging online learning scenario. Furthermore, our complete framework is capable of lifelong learning from scratch in online mode.

2.1.3 Problem Formulation

Online continual learning [30] is a subarea of continual learning that are additionally bounded by run-time and capability of lifelong learning with limited data compared to offline learning. However, these constraints are very much related to real life applications where new data comes in sequentially and is in conflict with the traditional assumption that complete data is available. A sequence of model $h_1, h_2, ..., h_t$ is generated on the given stream of data blocks $s_1, s_2, ..., s_t$ as shown in Figure 2.1. In this case, s_{i} is a block of new data with block size p, defined as the number of data used to update the model, which is similar to batch size as in offline learning mode. However, each new data is used only once to update the model instead of training the model using the new data with multiple epochs as in offline mode. $s_t = \{(\mathbf{x}_t^{(1)}, y_t^{(1)}), ..., (\mathbf{x}_t^{(p)}, y_t^{(p)})\} \in R^n \times \{1, ..., M\}$ where n is the data dimension and M is the total number of classes. The model $h_t : R^n \rightarrow \{1, ..., M\}$ depends solely on the

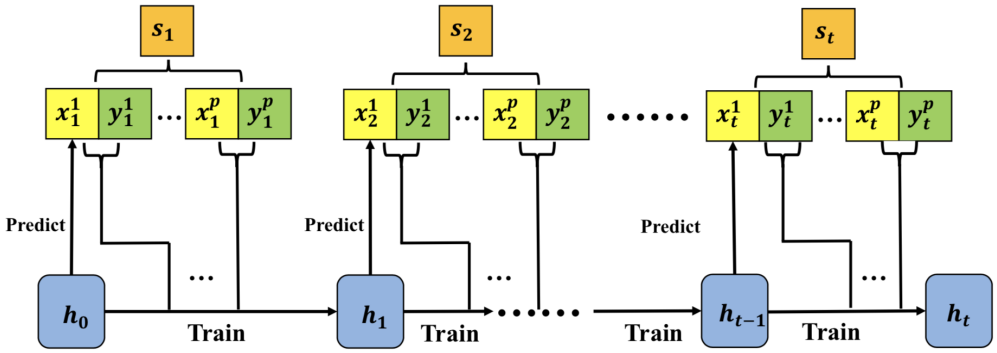


Figure 2.1. Continual learning in online scenario. A sequence of model $h_1, h_2, ..., h_t$ is generated using each block of new data with block size p, where $(\mathbf{x}_t^{\mathrm{i}}, y_t^{\mathrm{i}})$ indicate the i-th new data for the t-th block.

model h_{t-1} and the most recent block of new data s_t consisting of p examples with p being strictly limited, e.g. if we set $p = 16$ then we will predict for each new data and use a block of 16 new data to update the model.

Catastrophic forgetting is the main challenge faced by all incremental learning algorithms. Suppose a model h_{base} is initially trained on n classes and we update it with m new added classes to form the model h_{new}. Ideally, we hope h_{new} can predict all $n+m$ classes well, but in practice the performance on the n old classes drop dramatically due to the lack of old classes data when training the new classes. In this work, we propose a modified cross-distillation loss and a two-step learning technique to address this problem in online scenario.

Concept drift is another problem that happens in most real life applications. Concept [48] in classification problems is defined as the joint distribution $P(X, Y)$ where X is the input data and Y represents target variable. Suppose a model is trained on data streams by time t with joint distribution $P(X_t, Y_t)$, and let $P(X_n, Y_n)$ represent the joint distribution of old classes in future data streams. Concept drift happens when $P(X_t, Y_t) \neq P(X_n, Y_n)$. In this work, we do not measure concept drift quantitatively, but we provide a simple yet effective method to mitigate the problem by updating the exemplar set using the features of each new data in old classes.

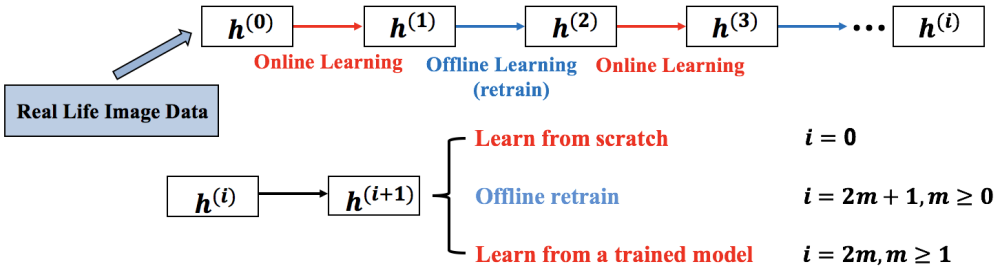


Figure 2.2. Proposed incremental learning framework. $h^{(i)}$ indicates the evolving model at i-th step.

2.1.4 Proposed Method

In this work, we propose an incremental learning framework as shown in Figure 2.2 that can be applied to any online scenario where data is available sequentially and the network is capable of lifelong learning. There are three parts in our framework: *learn from scratch*, *offline retraining* and *learn from a trained model*. Incremental learning in online scenario is implemented in 2.1.4 and lifelong learning can be achieved by alternating the last two parts after initial learning.

Learn from Scratch

This part serves as the starting point to learn new classes. In this case, we assume the network does not have any previous knowledge of incoming classes, which means there is no previous knowledge to be retained. Our goal is to build a model that can adapt to new classes fast with limited data, e.g. block size of 8 or 16.

Baseline. Suppose we have data with block size p belong to M classes: $\{s_1, ..., s_t\} \in R^n \times \{1, ..., M\}$. The baseline for the model to learn from sequential data can be thought as generating a sequence of model $\{h_1, ..., h_t\}$ using standard cross-entropy where h_t is updated from h_{t-1} by using block of new data s_t. Thus h_t is evolving from h_0 for a total of t updates by using the given data streams. Compared to traditional offline learning, the complete data is not available and we need to update the model for each block of new data to make it

dynamically fit to the data distribution used so far. So in the beginning, the performance on incoming data is poor due to data scarcity.

Online representation learning. A practical solution is to utilize representation learning when data is scarce at the beginning of the learning process. Nearest class Mean (NCM) classifier [46], [49] is a good choice where the test image is classified as the class with the closest class data mean. We use a pre-trained deep network to extract features by adding a representation layer before the last fully connected layer for each input data $\mathbf{x}_\mathrm{i}$ denoted as $\phi(\mathbf{x}_\mathrm{i})$. Thus the classifier can be expressed as

$$y^* = \underset{y \in \{1,\ldots,M\}}{\arg\min} \quad d(\phi(\mathbf{x}), \mu_y^\phi). \tag{2.1}$$

The class mean $\mu_y^\phi = \frac{1}{N_y} \sum_{\mathrm{i}:y_\mathrm{i}=\mathrm{i}} \phi(\mathbf{x}_\mathrm{i})$ and N_y denote the number of data in classes y. We assume that the highly non-linear nature of deep representations eliminates the need of a linear metric and allows to use Euclidean distance here

$$d_{xy}^\phi = (\phi(\mathbf{x}) - \mu_y^\phi)^T (\phi(\mathbf{x}) - \mu_y^\phi) \tag{2.2}$$

Our method: combining baseline with NCM classifier. NCM classifier behaves well when number of available data is limited since the class representation is based solely on the mean representation of the images belonging to that class. We apply NCM in the beginning and update using an online estimate of the class mean [50] for each new observation.

$$\mu_y^\phi \leftarrow \frac{n_{y\mathrm{i}}}{n_{y\mathrm{i}} + 1} \mu_y^\phi + \frac{1}{n_{y\mathrm{i}} + 1} \phi(\mathbf{x}_\mathrm{i}) \tag{2.3}$$

We use a simple strategy to switch from NCM to baseline classifier when accuracy for baseline surpass representation learning for s consecutive blocks of new data. Based on our empirical results, we set $s = 5$ in this work.

Offline Retraining

In order to achieve lifelong learning, we include an offline retraining part after each online incremental learning phase. By adding new classes or new data of existing class, both catastrophic forgetting and concept drift [32] become more severe. The simplest solution is to include a periodic offline retraining by using all available data up to this time instance.

Construct exemplar set. We use herding selection [51] to generate a sorted list of samples of one class based on the distance to the mean of that class. We then construct the exemplar set by using the first q samples in each class $\{E_1^{(y)}, ...E_q^{(y)}\}, y \in [1, ..., n]$ where q is manually specified. The exemplar set is commonly used to help retain the old classes' knowledge in incremental learning methods.

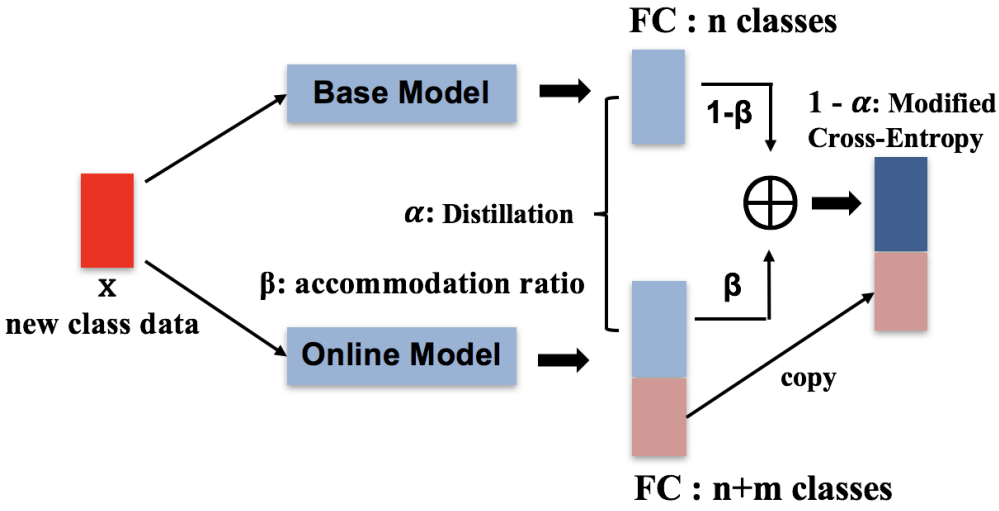


Figure 2.3. Modified Cross-Distillation Loss. It contains two losses: the distilling loss on old classes and the modified cross-entropy loss on all old and new classes.

Learn from a Trained Model

This is the last component of our proposed incremental learning framework. The goal here is to continue to learn from new data streams starting from a trained model. Different from existing incremental learning, we define new data containing both new classes data and new observations of old classes and we use each new data only once for training in online

scenario. In additional to addressing the catastrophic forgetting problem, we also need to consider concept drift for already learned classes due to the fact that data distribution in real life application may change over time in unforeseen ways [31].

Baseline: original cross-distillation loss. Cross-distillation loss function is commonly used in state-of-the-art incremental learning methods to retain the previous learned knowledge. In this case, we consider only new classes data for incoming data streams. Suppose the model is already trained on n classes, and there are m new classes added. Let $\{(\mathbf{x}_\mathrm{i}, y_\mathrm{i}), y_\mathrm{i} \in [n+1, ...n+m]\}$ denote new classes data. The output logits of the new classifier is denoted as $p^{(n+m)}(x) = (o^{(1)}, ..., o^{(n)}, o^{(n+1)}, ...o^{(n+m)})$, the recorded old classes classifier output logits is $\hat{p}^{(n)}(x) = (\hat{o}^{(1)}, ..., \hat{o}^{(n)})$. The knowledge distillation loss **kd** can be formulated as in Equation 2.4, where $\hat{p}_T^{(\mathrm{i})}$ and $p_T^{(\mathrm{i})}$ are the i-th distilled output logit as defined in Equation 2.5

$$L_D(x) = \sum_{\mathrm{i}=1}^{n} -\hat{p}_T^{(\mathrm{i})}(x) log[p_T^{(\mathrm{i})}(x)] \tag{2.4}$$

$$\hat{p}_T^{(\mathrm{i})} = \frac{\exp\left(\hat{o}^{(\mathrm{i})}/T\right)}{\sum_{\mathrm{j}=1}^{n} \exp\left(\hat{o}^{(\mathrm{j})}/T\right)} \ , \ p_T^{(\mathrm{i})} = \frac{\exp\left(o^{(\mathrm{i})}/T\right)}{\sum_{\mathrm{j}=1}^{n} \exp\left(o^{(\mathrm{j})}/T\right)} \tag{2.5}$$

T is the temperature scalar. When $T = 1$, the class with the highest score has the most influence. When $T > 1$, the remaining classes have a stronger influence, which forces the network to learn more fine grained knowledge from them. The cross entropy loss to learn new classes can be expressed as $L_C(x) = \sum_{\mathrm{i}=1}^{n+m} -\hat{y}^{(\mathrm{i})} log[p^{(\mathrm{i})}(x)]$ where $\hat{y}$ is the one-hot label for input data x. The overall cross-distillation loss function is formed as in Equation 2.6 by using a hyper-parameter α to tune the influence between two components.

$$L_{CD}(x) = \alpha L_D(x) + (1-\alpha) L_C(x) \tag{2.6}$$

Modified cross-distillation with accommodation ratio. Although cross-distillation loss forces the network to learn latent information from the distilled output logits, its ability to retain previous knowledge still remains limited. An intuitive way to make the network retain previous knowledge is to keep the output from the old classes' classifier as a part of the final classifier. Let output logits of the new classifier be denoted as $p^{(n+m)}(x) =$

$(o^{(1)}, ..., o^{(n)}, o^{(n+1)}, ...o^{(n+m)})$, the recorded old classes' classifier output logits is $\hat{p}^{(n)}(x) = (\hat{o}^{(1)}, ..., \hat{o}^{(n)})$. We use an accommodation ratio $0 \leq \beta \leq 1$ to combine the two classifier output as

$$\tilde{p}^{(\mathrm{i})} = \begin{cases} \beta p^{(\mathrm{i})} + (1-\beta)\hat{p}^{(\mathrm{i})} & 0 < \mathrm{i} \leq n \\ p^{(\mathrm{i})} & n < \mathrm{i} \leq n+m \end{cases} \tag{2.7}$$

When $\beta = 1$, the final output is the same as the new classifier and when $\beta = 0$, we replace the first n output units with the old classes classifier output. This can be thought as using the accommodation ratio β to tune the output units for old classes. As shown in Figure 2.3, the modified cross-distillation loss can be expressed by replacing the original cross-entropy loss part $L_C(x)$ with the new modified cross-entropy loss $\tilde{L}_C(x) = \sum_{\mathrm{i}=1}^{n+m} -\hat{y}^{(\mathrm{i})} log[\tilde{p}^{(\mathrm{i})}(x)]$ after applying the accommodation ratio as in Equation 2.8

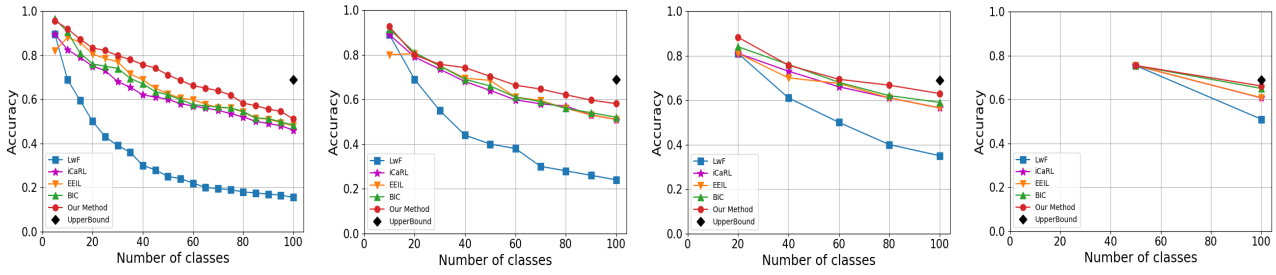
$$\tilde{L}_{CD}(x) = \alpha L_D(x) + (1-\alpha)\tilde{L}_C(x) \tag{2.8}$$

We empirically set $\beta = 0.5$, $T = 2$ and $\alpha = \frac{n}{n+m}$ in this work where n and m are the number of old and new classes. The modified cross-distillation loss push the network to learn more from old classes' output units since we add it directly as part of the final output.

Update exemplar set. As described in Section 4.1.1, we consider the new data streams containing both new classes data and new observations of old classes with unknown distribution. In this case, retaining previous knowledge is not sufficient since concept drift can happen to old classes and the model will still undergo performance degradation. One solution is to keep updating the network using the exemplars for old classes. The class mean of each old class $\{M^{(1)}, ..., M^{(n)},\ M^{(\mathrm{i})} \in R^n\}$ is calculated and recorded as described in Section 2.1.4 by constructing the exemplar set $\{(E_1^{(y)}, ...E_q^{(y)}), y \in [1, ..., n]\}$ using previous data streams. Let $\{(\mathbf{x}_\mathrm{i}, y_\mathrm{i}), y_\mathrm{i} \in [1, ..., n]\}$ denote the new observation of old classes. We follow the same online class mean update as described in Equation 2.3 to keep updating the class mean with all data seen so far. So when concept drift happens, e.g., the class mean changes toward the new data, we update the exemplar set to make new data more likely to be selected to update the model during two-step learning as described in next part.

Two-step learning. Unlike other incremental learning algorithms that mix new classes data with old classes exemplars, we first let the model learn from a block of new classes data and then a balanced learning step is followed. This two-step learning technique is deigned for online learning scenarios, where both update time and number of available data are limited. As shown in Figure 2.5, we use the modified cross-distillation loss in the first step to overcome catastrophic forgetting since all data in this block belongs to new classes. In the second step, we pair same number of old classes exemplars from the exemplar set with the new classes data. As we have balanced new and old classes, cross entropy loss is used to achieve balanced learning.

2.1.5 Experiment



(a) (b) (c) (d)

Figure 2.4. Incremental learning results on CIFAR-100 with split of (a) 5 classes, (b) 10 classes, (c) 20 classes and (d) 50 classes. The **Upper Bound** in last step is obtained by offline training a model using all training samples from all classes. (Best viewed in color)

Our experimental results consists of two main parts. In part one, we compare our modified cross-distillation loss and the two-step learning technique as introduced in Section 2.1.4 with current state-of-the-art incremental learning methods [10], [41], [46], [47]. We follow the iCaRL experiment benchmark protocol [46] to arrange classes and select exemplars, but in the more challenging online learning scenario as illustrated in Section 2.1.5. Our method is implemented on two public datasets: **CIFAR-100** [2] and **ImageNet-1000** (ILSVRC 2012) [1]. Part two is designed to test the performance of our complete framework. Since

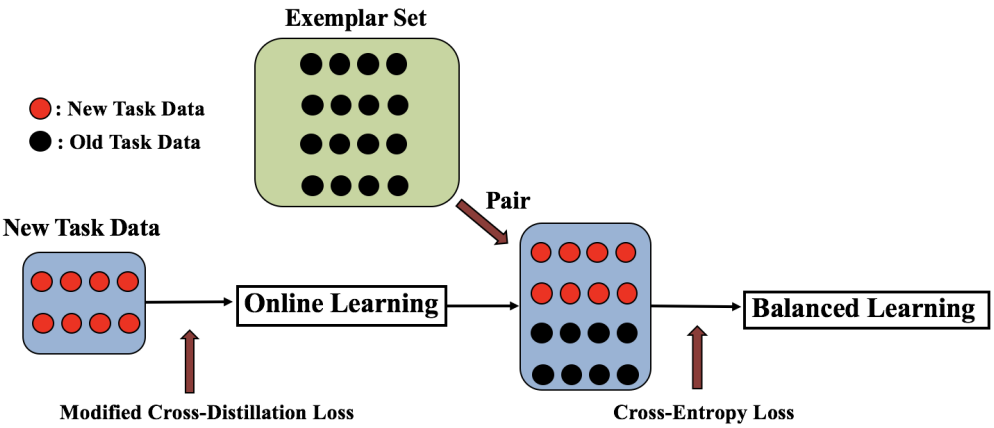


Figure 2.5. Two-Step Learning. Black dots correspond to old classes samples stored in exemplar set. Red dots correspond to samples from new classes.

our goal is to set up an incremental learning framework that can be applied to online learning scenario, we use **Food-101** [52] food image dataset to evaluate our methods. For each part of our proposed framework, we compare our results to baseline methods.

Datasets

We used three public datasets. Two common datasets: CIFAR-100 and ImageNet-1000 (ILSVRC 2012) and one food image dataset: Food-101.

Food-101 is the largest real-world food recognition dataset consisting of 1k images per food classes collected from *foodspotting.com*, comprising of 101 food classes. We divided 80% for training and 20% for testing for each class.

CIFAR-100 consists of 60k 32×32 RGB images for 100 common objects. The dataset is originally divided into 50K as training and 10k as testing.

ImageNet-1000 (ILSVRC 2012) ImageNet Large-Scale Visual Recognition Challenge 2012 (ILSVRC12) is an annual competition which uses a subset of ImageNet. This subset contains 1000 classes with more than 1k images per class. In total, there are about 1.2 million training data, 50k validation images, and 150k testing images.

Data pre-processing For Food-101, we performed image resize and center crop. As for CIFAR-100, random cropping and horizontal flip was applied following the original im-

plementation [53]. For ImageNet, we follow the steps in VGG pre-processing [54], including random cropping, horizontal flip, image resize and mean subtraction.

Implementation Detail

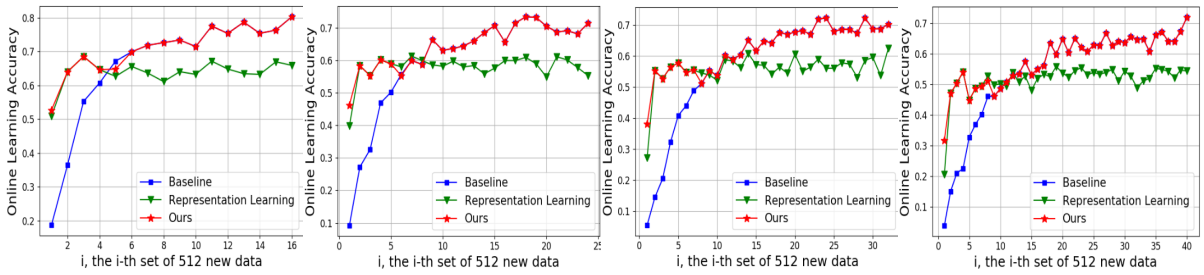
Our implementation is based on Pytorch [55]. For experiment part one, we follow the same experiment setting as current state-of-the-art incremental learning methods, a standard 18-layer ResNet for ImageNet-1000 and a 32-layer ResNet for CIFAR-100. For experiment part two, we applied a 18-layer ResNet to Food-101. The ResNet implementation follows the setting suggested in [53]. We use stochastic gradient descent with learning rate of 0.1, weight decay of 0.0001 and momentum of 0.9.

Selection of block size p in online learning scenario. Different from offline learning scenario, where we select a batch size to maximize overall performance after many epochs. In online learning scenario, we need to select block size p based on real life applications. More specifically, a large block size causes slow update since we have to wait until enough data arrives to update the model. On the other hand, using a very small block size, e.g., update with each new observation, although very fast, is not suitable for deep neural network due to strong bias towards new data. Therefore, we design a pretest using the first 128 new data for each experiment to repeatedly update the model by varying block size $p \in \{1, 2, 4, 8, 16, 32, 64\}$. Thus the optimal block size is chosen which gives the highest accuracy on these 128 new data. We do not consider $p > 64$ as such a large block size is not practical for real life applications.

Table 2.1. Online learning from scratch on Food-101 with (a) Online accuracy and (b) Testing accuracy. The **Upper Bound** is obtained by offline training a model using all training samples from all given classes. (Best result marked in bold)

Method	20	30	40	50
Baseline	62.81%	56.53%	54.35%	51.39%
Representation Learning	60.21%	55.32%	53.68%	51.26%
Ours	**70.90%**	**64.32%**	**62.31%**	**57.83%**

	Testing	Upper Bound
20	78.77%	84.17%
30	73.28%	80.95%
40	71.42%	77.82%
50	67.54%	74.46%



(a) (b) (c) (d)

Figure 2.6. Starting from scratch on Food-101 with number of new classes (a) 20 classes (b) 30 classes (c) 40 classes and (d) 50 classes. Baseline and our method are illustrated in Section 2.1.4 (Best viewed in color)

Evaluation of Modified Cross-Distillation Loss and Two-Step Learning

In this part, we compared our modified cross-distillation loss and two-step learning technique with the current state-of-the-art methods [10], [46], [47]. We consider the online setting that new classes data comes sequentially and we predict each new data at first and then use a block of new data to update the model. For each incremental step, we compare our accuracy obtained in online scenario with state-of-the-art results in offline mode. We constructed the exemplar set for both CIFAR and ImageNet with the same number of samples as in [10], [46], [47] for fair comparison.

Table 2.2. Online learning from a trained model on Food-101 with **baseline method using original cross-distillation loss** in the left of $\rightarrow$ and **our proposed method** in the right (best result marked in bold), ([.5]) shows the **Upper Bound** results.

	Online Accuracy		Test Accuracy	
Incremental Step	new	old	new	old
20	54.35% $\rightarrow$ **64.78%**	22.83% $\rightarrow$ **61.01%**	**70.97%** $\rightarrow$ 64.00%	41.77% $\rightarrow$ **70.32%** (84.17%)
30	52.62% $\rightarrow$ **62.25%**	22.41% $\rightarrow$ **60.00%**	**71.56%** $\rightarrow$ 61.87%	42.25% $\rightarrow$ **69.90%** (80.95%)
40	46.30% $\rightarrow$ **61.53%**	20.53% $\rightarrow$ **53.43%**	**66.62%** $\rightarrow$ 56.31%	40.82% $\rightarrow$ **65.65%** (77.82%)
50	43.49% $\rightarrow$ **56.76%**	19.47% $\rightarrow$ **51.71%**	**63.32%** $\rightarrow$ 54.20%	36.81% $\rightarrow$ **63.92%** (74.46%)

CIFAR-100. We divided 100 classes into splits of 5, 10, 20, and 50 in random order. Therefore, we have incremental training steps for 20, 10, 5, and 2, respectively. The optimal block size is set as $p = 8$. We ran the experiment for four trials and each time with a random

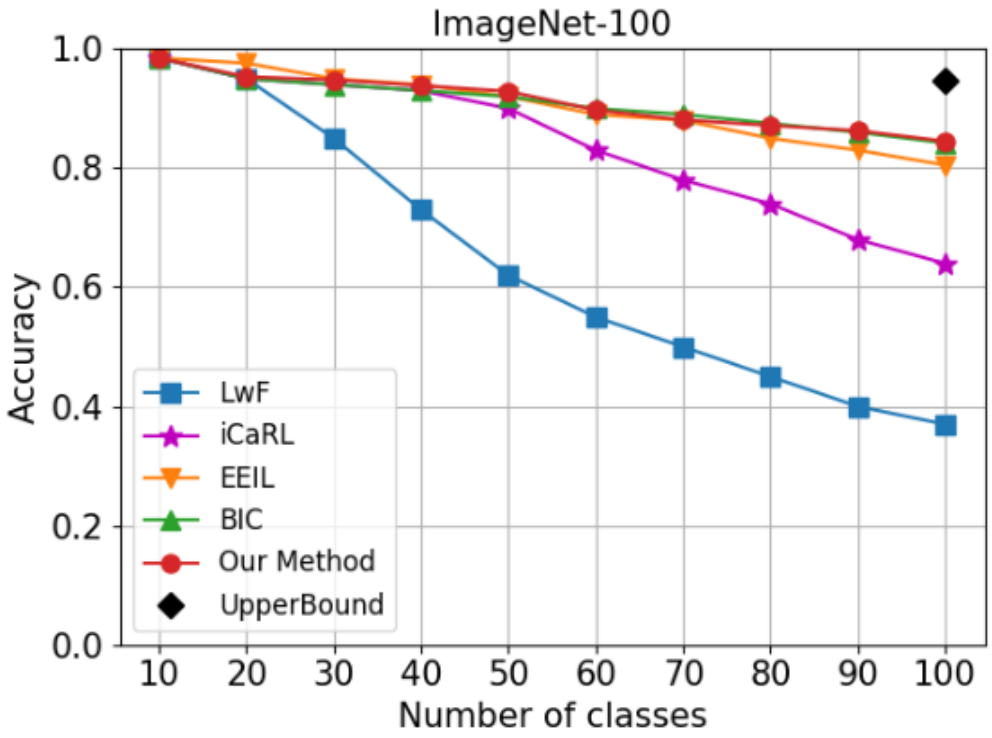


Figure 2.7. Incremental learning results on ImageNet-100 with split of 10 classes. The **Upper Bound** in last step is obtained by offline training a model using all training samples from all classes. (Best viewed in color)

order for the 100 classes. The average accuracy is shown in Figure 2.4. Our method shows the best accuracy for all incremental learning steps even in the challenging online learning scenario.

ImageNet-1000. As 1000-class is too large and impractical for online scenario, so we randomly selected 100 classes from the 1000 classes to construct a subset of the original dataset, which is referred to as ImageNet-100. We then divided the 100 classes into 10 classes split so we have an incremental step of 10. The optimal block size is set as $p = 16$. We ran this for four trials as before and we recorded the average accuracy in each step as shown in Figure 2.7. Although the performance of EEIL [47] surpass our method in the second step, we attain the best performance as more classes are added.

Evaluation of Our Complete Framework

We used a food image dataset **Food-101** [52] to evaluate performance of our proposed incremental learning framework.

Benchmark protocol of online incremental learning. Until now, there is no benchmark protocol on how to evaluate an online incremental learning method. In addition to address catastrophic forgetting [3] as in offline incremental learning, we also need to consider concept drift [32] in online scenario. We propose the following evaluation procedure: for a given multi-class classification dataset, the classes should be randomly arranged. For each class, the training data should be further split into new training data and old training data. The former is used when a class is introduced to the model for the first time. The later is considered when the model has seen the class before, which is used to simulate real life applications and test the ability of the method to handle new observations of old classes. After each online learning phase, the updated model is evaluated on test data containing all classes already been trained so far. There is no over-fitting since the test data is never used to update the model. In addition to the overall test accuracy, we should separately examine the accuracy for new classes and accuracy for old classes data. We also suggest to use online accuracy, which is the accuracy for data in training set before they are used to update the model, to represent the classification performance of future data stream. In general, online accuracy shows the model's ability to adapt to future data stream and online accuracy for old classes indicates the model's ability to handle new observations of old classes.

Results on Food-101

Although there are three separate components of the proposed incremental learning framework, we only test the component described in 2.1.4 once and then alternate between the two components described in 2.1.4 and 2.1.4. In addition, the offline retraining part in 2.1.4 is inapplicable with online incremental learning. So in this experiment, we test for one cycle of our proposed framework starting from scratch then learning from a trained model provided by offline retraining. We use half training data per class as new classes data and the other half as new observations of old classes. We divided the Food-101 dataset into

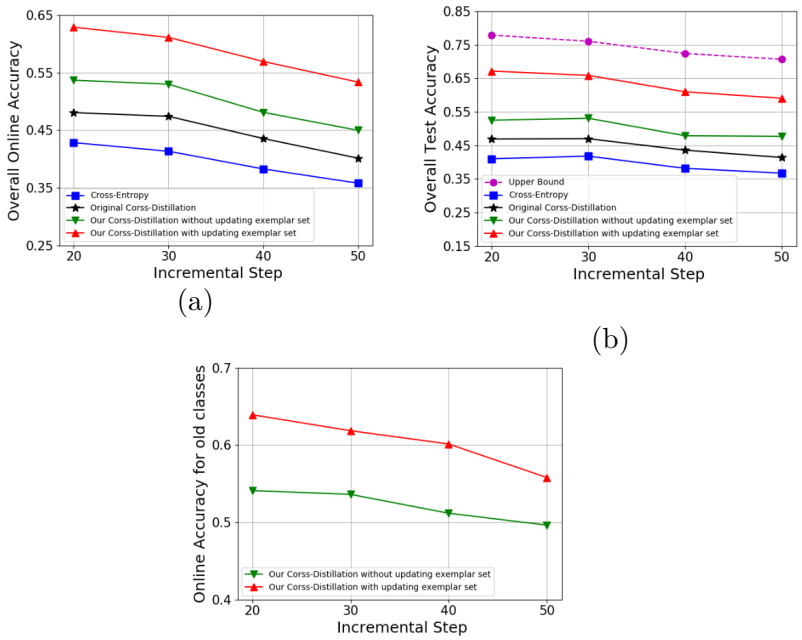


Figure 2.8. Ablation study on Food-101 dataset with (a) overall online accuracy (b) overall test accuracy (c) online accuracy for old classes. (Best viewed in color)

split of 20, 30, 40, 50 classes randomly and performed the one incremental step learning with step size of 20, 30, 40, and 50, respectively. In addition, we construct exemplar set with only 10 samples per class to simulate real life applications instead of including more samples per class.

Learn from scratch. In this part, we evaluate our method that combines baseline and representation learning. Optimal block size is set as $p = 16$. Result of online accuracy compared to baseline and representation learning is shown in Table 2.1. Our method achieved the best online accuracy in all incremental learning steps. Similarly, test accuracy compared to upper bound is shown in Table 2.1. We also calculated the accuracy of each 512 incoming new data as shown in Figure 2.6. We observed that the representation learning works well at the beginning when data is scarce and the baseline achieved higher accuracy as the number of new data increases. Thus by combining the two methods and automatically switch from one to the other, we attain a higher overall online accuracy.

Learn from a trained model. In this part, we perform a one incremental step experiment following our proposed benchmark protocol described in Section 2.1.5 and the result is shown in Table 2.2. Compared to the baseline, our method improved the online learning accuracy for both new and old classes, which shows that our model can adapt quickly to future data stream including both new classes data or new observations of old classes. In addition, we significantly improved the test accuracy compared to the baseline method. However, the trade off is slightly lower accuracy for the new classes test accuracy compared to the baseline due to the use of the accommodation ratio in our method. Since it is difficult for the model to perform well on new classes without losing knowledge from the old classes, the accommodation ratio can be manually tuned to balance between the new classes and the old classes depending on the application scenario. A higher accommodation ratio leads to higher accuracy on new classes by trading off accuracy on old classes. For this experiment, we simply use $\beta = 0.5$.

Ablation study. We analyzed different components of our method to demonstrate their impacts. We first show the influence of different loss functions including cross-entropy, cross-distillation, and our modified cross-distillation. We then analyzed the impact of updating the exemplar set to mitigate concept drift. As shown in Figure 2.8a and 2.8b, even without updating exemplar set, our modified cross-distillation loss outperformed the other two (black and blue lines) for all incremental steps. By updating the exemplar set, we were able to achieve a higher overall online and test accuracy. Furthermore, Figure 2.8c illustrates improvement of online accuracy for old classes by updating the exemplar set. Since we do not deliberately select any new data from old classes to update the model during the incremental learning step, as the data distribution changes, our method was able to automatically update the exemplar set by using the current class mean calculated by all data in old classes seen so far. Thus through the proposed two-step learning which pairs each new data with an exemplar, we can achieve a higher online accuracy for future data streams.

2.2 Online Continual Learning Via Candidates Voting

2.2.1 Overview

Continual learning, a promising future learning strategy, is able to learn from a sequence of tasks incrementally using less computation and memory resource compared with retraining from scratch whenever observing a new task. However, it suffers from catastrophic forgetting [3], in which the model quickly forgets already learned knowledge due to the unavailability of old data. Existing methods address this problem under different scenarios including (1) *task-incremental* vs. *class-incremental* depending on whether task index is available and (2) *offline* vs. *online* depending on how many passes are allowed to use each new data. In general, *online class-incremental* methods use each data once to update the model and employs a single-head classifier [7] to test on all classes encountered so far during inference. This setting is more closer to real life learning environment where new classes come in as data streams with limited adaptation time and storage capacity allowed for processing [8]. Unfortunately, class-incremental learning in online scenario is not well-studied compared with offline setting. In addition, existing online methods [21], [56]–[59] all require original data from each learned task as exemplars, which restricts their deployment for certain applications (*e.g.*, healthcare and medial research) with memory constraints or privacy concerns. Therefore, an effective online continual learning method is needed to address the above challenges to improve the performance of online methods.

Motivated by the observation that the model is still able to maintain its discriminability for classes within each task [60] despite the bias issue, *i.e.*, the correct class label can be drawn from the candidate prediction given by each learned task during inference, we further propose to treat the class label associated with the largest output logit for each learned task as a candidate and the final prediction is based on the weighted votes of all selected candidates. Figure 2.9 illustrates the main difference between our method and others to make prediction based on the output of a single-head classifier.

To achieve this goal, there are two associated questions we need to address: (1) How to obtain the largest logits as candidates from the output of each learned task using a single-head classifier without knowing the task index? (2) How to generate the weight for each

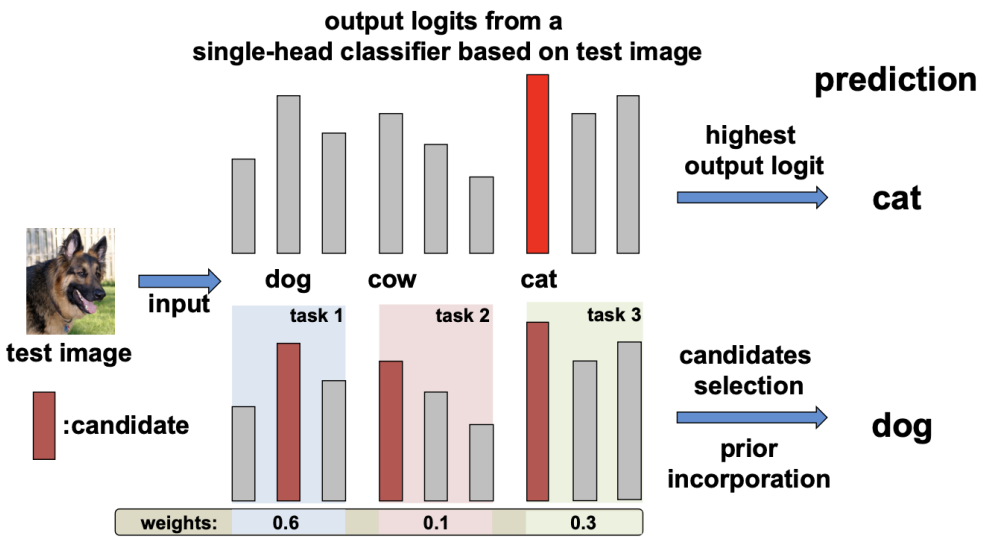


Figure 2.9. Illustration of the difference between our proposed method and other methods to make prediction based on output of a single-head classifier. With single-head classifier, the output is associated with the largest value of the output logits. In contrast, our method makes prediction by first selecting candidates from each learned task and then incorporating the corresponding weights.

selected candidate to determine the final prediction? In this work, we address both problems by leveraging exemplar set [46], where a small number of old task data is stored for replay during continual learning. However, different from existing methods [21], [56]–[59] which use original data as exemplar, we apply a feature extractor and store only feature embeddings, which is more memory-efficient and privacy-preserving. We argue that the task index can be stored together with selected exemplars while learning each new task. Therefore, during inference phase, we can directly obtain the output logits for each learned task from the single-head classifier based on stored task index in the exemplar set and extract the largest output logits. We refer to this as the **candidates selection** process. In addition, we design a probabilistic neural networks [61] leveraging all stored feature embeddings to generate the probability distribution of learned task that the input test data belongs to, and use it as the weights to decide the final prediction. We denote this step as **prior incorporation**.

2.2.2 Related Work

In this part, we study the continual learning under online and class-incremental setting, where the model observes each data once and perform classification within all seen classes during inference phase. We review existing continual learning works related to our method in two categories including (1) Regularization-based and (2) Replay-based methods.

Regularization-based methods restrict the impact of learning new tasks on the parameters that are important for learned tasks. Representative methods include freezing part of layers [40], [62] and using distillation loss or its variants [21], [41], [46], [47], [63]–[66]. However, they also limit the model's ability to learn new task and can even harm the performance if the teacher model used by distillation [9] is not learned on large balanced data [67]. Our method applies a fixed backbone model that is pre-trained on large scale datasets to extract feature embeddings of new data as input and uses cross-entropy to learn a discriminative classifier for each new task. Therefore, even though we freeze the parameters for learned tasks in the classifier, it has minimum impact on extracted features to learn new task. Recent studies [10], [60] also found that the bias of model weights towards new classes is one of the reasons for catastrophic forgetting. Therefore, Wu *et al.* [10] proposed to correct the weights by applying an additional linear model. Then Weight Aligning is proposed in [60] to directly correct the biased weights in the FC layer without requiring additional parameters. However, none of these methods are designed for online scenario where each data is only allowed to use once for training. In this work we propose to tackle this problem from a novel perspective by selecting candidates for each learned task and then use the weighted score for final prediction, which effectively addresses catastrophic forgetting in online case.

Replay-based methods are shown to be effective for maintaining learned knowledge by either using the original data as exemplars [46], [47], [56]–[59], [68]–[73] or synthetic data and statistics [43], [44], [74], [75]. However, using original data may not be feasible for certain applications due to privacy concerns and also it may require large storage depending on the size of input data. In addition, using synthetic data or data statistic require training a generative model [45] during learning phase, which is not feasible in online scenario. Therefore, we propose to use feature embeddings as exemplars for rehearsal to mitigate forgetting in

online case. Besides, we also utilize the stored feature to (1) generate binary masks for each learned task to select candidates and (2) provide prior information as weights to obtain final prediction. We argue that both information are valuable to explore, particularly under the online continual learning context when available resource is limited.

Among these methods, only a few are studied for online mode [56]–[59], [68], [71]–[73] with even less work under class-incremental setting [57]–[59], [71], which is more challenging but also worth investigating as it closely relates to applications in real world scenario.

2.2.3 Proposed Method

The overview of our method is illustrated in Figure 2.10, including a learning phase to learn new task from a data stream and an inference phase to test for all tasks seen so far. Our method applies a fixed backbone network to extract feature embedding as input, which is more discriminative, memory-efficient and also privacy-preserving compared with using original data. We freeze the parameters in the classifier after learning each new task to maximally maintain its discriminability. We emphasize that our method still uses a single-head classifier but restricts the update of parameters corresponding to all learned tasks.

Learning Phase

The upper half of Figure 2.10 shows the learning phase in online scenario where we train the classifier by pairing each extracted feature embedding of the new data with one exemplar randomly selected from exemplar set into the training batch. Cross-entropy is used as the classification loss to update the model, which generates a more discriminative classifier as no regularization term on learned tasks is used. It also does not require additional memory to store the output logits compared with using knowledge distillation loss [9].

Online sampler: There are two necessary conditions we need to satisfy when designing the online sampler for our method: (1) it should be able to select exemplars from sequentially available data in online scenario, (2) the selected exemplars should near the class mean as we will leverage stored features to provide prior information using distance-based metric during inference phase, which is described later in Section 2.2.3. However, none of the existing

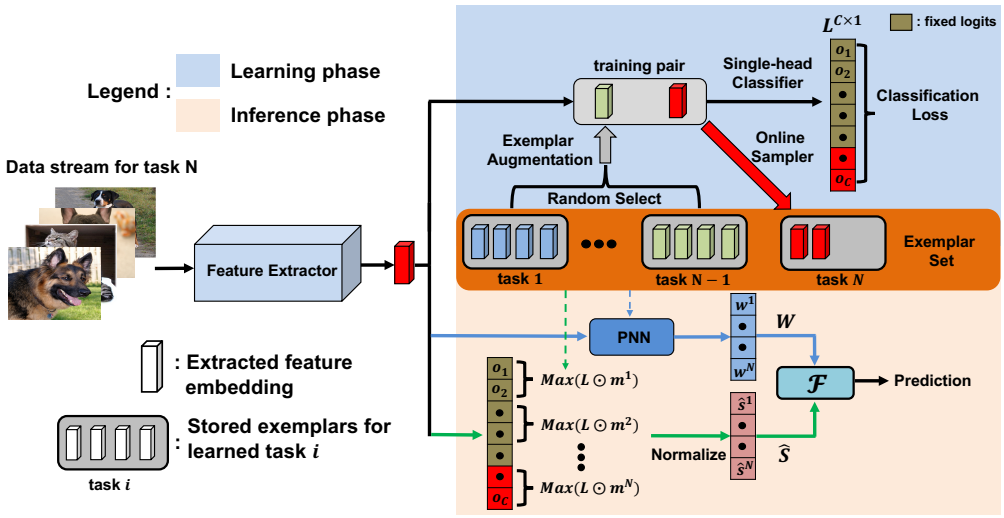


Figure 2.10. Overview of our proposed online continual learning method to learn a new task N.The upper half shows the learning phase where we pair the extracted feature of new data with an exemplar to train the single-head classifier. L denotes the output logits for all classes C seen so far. The parameters for each learned task in the classifier are fixed to maximally maintain its discriminability and an online sampler is designed to select exemplars for current task N. The lower half shows the inference phase where the candidates selection and prior incorporation are denoted by green and blue arrows, respectively. The output logits for each learned task is obtained using element-wise product on classifier output L and binary mask $\{m^{\mathrm{i}}, \mathrm{i} = 1, 2, ...N\}$ generated from exemplar set and we treat the highest logits for each task as candidates. A probabilistic neural network (PNN) is designed using all stored exemplars to provide the prior information of which task index the input data belongs to during inference, which can be regarded as weights for selected candidates to obtain the final prediction using our proposed function $\mathcal{F}$. (Best viewed in color)

exemplar selection algorithms satisfy both conditions. In addition, although Herding [51] is widely applied to select exemplars based on class mean, it only works in offline scenario assuming the data from new task is all available. Therefore, we propose to use an online dynamic class mean update criteria [76] for exemplar selection, which does not require knowing the total number of data beforehand as shown in Equation 4.6.

$$\mathbf{v}_{mean} = \frac{n}{n+1}\mathbf{v}_{mean} + \frac{1}{n+1}\mathbf{v}_n \tag{2.9}$$

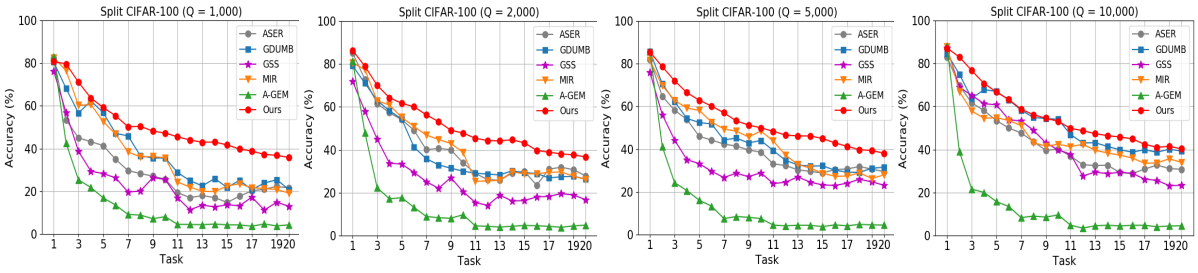
where n refers to the number of data seen so far in this class and $\mathbf{v}_n$ denotes a new observation. For the exemplar selection process of a new task N, where $q = \frac{Q}{|class|}$ denotes the number of allowable exemplars per class given total capacity Q and $f_m^{(y_i)}$ is the mean vector for total n_{y_i} data seen so far for class label y_i. The exemplar set can be expressed as $E = \{(\mathbf{v}_1, y_1)^1, (\mathbf{v}_2, y_2)^1, ..., (\mathbf{v}_1, y_1)^N, (\mathbf{v}_2, y_2)^N, ...\}$, where $(\mathbf{v}_j, y_j)^k$ denotes the j-th stored exemplar for the k-th learned task and $k \in \{1, 2, ..., N\}$. Each stored exemplar contains extracted feature $\mathbf{v}$, class label y and task index k.

Exemplar augmentation in feature space: Although exemplars help to remember learned tasks by knowledge reply during continual learning, the model performance greatly depends on the size of the exemplar set, *i.e.,* the larger the better, which is challenging given a limited memory budget particularly in online scenario. Therefore, we also study the exemplar augmentation techniques in this work to help improve the performance without requiring additional storage. Since we store feature embedding as exemplar, common data augmentation methods that are typically applied to image data such as rotation, flip and random crop cannot be used directly in feature space. Therefore, we adopt random perturbation for feature augmentation [77].

Random perturbation: We generate pseudo feature exemplar by adding a random vector P drawn from a Gaussian distribution with zero mean and per-element standard deviation σ as shown in Equation 2.10

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i + \alpha_r P, \quad P \sim N(0, \sigma_i) \tag{2.10}$$

where $\mathbf{v}_i$ refers to the stored feature in exemplar set, and $\tilde{\mathbf{v}}_i$ denotes the augmented feature. α_r is a constant which controls the scale of noise, and is set to $\alpha_r = 1$ in our implementation. We emphasize that we do not need to store augmented feature in exemplar set and the exemplar augmentation is randomly implemented when pairing the extracted feature of new data.



(a) (b) (c) (d)

Figure 2.11. Results on Split CIFAR-100 by comparing with existing online methods with different exemplar size Q. The accuracy is measured after learning of each task on all tasks seen so far. (Best viewed in color)

Table 2.3. Average accuracy and Last step accuracy on Split CIFAR-10 and CORE-50. Best results marked in bold.

Datasets	**Split CIFAR-10**								**CORE-50**							
Size of exemplar set	$Q = 1,000$		$Q = 2,000$		$Q = 5,000$		$Q = 10,000$		$Q = 1,000$		$Q = 2,000$		$Q = 5,000$		$Q = 10,000$	
Accuracy(%)	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
A-GEM [56]	43.0	17.5	59.1	38.3	74.0	59.0	74.7	62.5	20.7	8.4	21.9	10.3	22.9	11.5	24.6	12.0
MIR [59]	67.3	52.2	80.2	66.2	83.4	74.8	86.0	78.4	33.9	21.1	37.1	24.5	38.1	27.7	41.1	31.8
GSS [71]	70.3	56.7	73.6	56.3	79.3	64.4	79.7	67.1	27.8	17.8	31.0	18.9	31.8	21.1	33.6	22.6
ASER [57]	63.4	46.4	78.2	59.3	83.3	73.1	86.5	79.3	24.3	12.2	30.8	17.4	32.5	18.5	34.1	21.8
GDUMB [58]	73.8	57.7	83.8	72.4	85.3	75.9	87.7	82.3	41.2	23.6	48.4	32.7	54.3	41.6	56.1	45.5
Ours	**76.0**	**62.9**	**84.9**	**74.1**	**86.1**	**77.0**	**88.3**	**82.7**	**45.1**	**26.5**	**50.7**	**34.5**	**56.3**	**43.1**	**57.5**	**46.2**

Inference Phase

The lower half of Figure 2.10 shows inference phase, which comprises of two key components: candidates selection and prior incorporation. The stored exemplars along with their task indexes are used to generate binary mask to obtain the corresponding output logits for each learned task during inference. We extract the highest output as candidates and a variant of probabilistic neural network (PNN) [61] using all stored exemplars is designed to provide prior information as weights for selected candidates to vote for final prediction, which will be described in detail below.

Candidates selection: We denote $L = \{o^1, o^2, ..., o^C\}$ as the output logits from the single-head classifier where C refers to the total number of seen classes belonging to N learned tasks so far. During inference phase, the exemplar set generates a binary mask $m^k \in \{0, 1\}^C$ for task k by assigning the i-th entry m^k_{i} as 1 if class label i belongs to task k

and as 0 if not, so we have $\sum_{\mathrm{i}=1}^{C} m_{\mathrm{i}}^{k} = C^{k}$, where C^{k} is the number of classes belonging to task k. Thus, the candidate output logit from each learned task is selected by

$$s^{k} = Max\{L \odot m^{k}\}, \quad k = 1, 2, ..., N \tag{2.11}$$

where $\odot$ refers to element-wise product. We then perform normalization step for the extracted candidate logits by using the corresponding norm of weight vectors in classifier. Specifically, for each selected candidate s^{k}, let $W^{k} \in \mathcal{R}^{d_m \times 1}$ and $|W^{k}|$ denotes the weight vector in classifier and its norm respectively where d_m is the input dimension. Then we normalize each candidate with

$$\hat{s}^{k} = \frac{1}{|W^{k}|} \frac{s^{k} - Min\{s^{1}, ...s^{N}\}}{\epsilon_n + \sum_{\mathrm{j}=1}^{N}(s^{\mathrm{j}} - Min\{s^{1}, ...s^{N}\})}$$

where ϵ_n is for regularization and larger $\hat{s}$ can reflect higher probability as prediction. Finally, the normalized selected candidates for N learned tasks can be expressed as $\hat{S} = \{\hat{s}^{1}, \hat{s}^{2}, ..., \hat{s}^{N}\}$ with corresponding extracted candidate class labels $Y = \{y^{1}, y^{2}, ..., y^{N}\}$.

Prior incorporation: We apply PNN to generate prior probability distribution of which learned task index the test data belongs to. PNN computes class conditional probabilities using all stored features in the exemplar set. Specifically, it calculates the probability that an input feature vector $\mathbf{f}$ belongs to task k as formulated in Equation 2.12 below.

$$\begin{aligned} P(k|\mathbf{f}) &= \frac{\alpha^{k}}{\sum_{\mathrm{i}=1}^{N} \alpha^{\mathrm{i}}} \\ \alpha^{k} &= (\epsilon_r + Min_{\mathrm{j}}||\mathbf{f} - \mathbf{v}_{\mathrm{j}}^{k}||_2))^{-1} \end{aligned} \tag{2.12}$$

where $\epsilon_r > 0$ is used for regularization and $\mathbf{v}_{\mathrm{j}}^{k}$ denotes the j-th stored feature in exemplar set for learned task k.

The output of PNN is a N dimension prior vector $W = (w^{1}, w^{2}, ..., w^{N})$ and we use it as the weights to combine with the normalized candidates $\hat{S}$ to get final predicted class label $\hat{y}$ using Equation 2.13.

$$\hat{y} = \underset{y^{\mathrm{i}} \in Y}{argmax}(\hat{s}^{\mathrm{i}} + \mathrm{e}^{(\gamma - 1)} \times w^{\mathrm{i}}) \tag{2.13}$$

where $\gamma = \frac{Max(W)-Min(W)}{\beta}$ is a dynamic hyper-parameter used for incorporation determined by calculating difference between maximum and minimum value in prior vector. $\beta \in (0, 1)$ is a normalization constant. In this work, we show the effectiveness of our method by using a fixed $\beta = 0.5$ for all experiments.

2.2.4 Experiment

To show the effectiveness of our proposed approach, we compare with both the state-of-the-art *online methods* following experiment setting similar in [56], [68], and *offline continual learning methods* as well under benchmark protocol [46] by varying the incremental step size, which are illustrated in Section 2.2.4 and Section 2.2.4, respectively. In Section 2.2.4, we conduct ablation experiments to validate each component of our propose method. Finally, we study the weight bias problem in online scenario and analyze the storage consumption in Section 2.2.4.

Evaluation Metrics

We focus on continual learning under class-incremental setting as illustrated in Section 4.2.2. During inference, the model is evaluated to classify all classes seen so far. We use commonly applied evaluation metrics such as average accuracy (*Avg*) and last step accuracy (*Last*) in this section where *Avg* is calculated by averaging all the accuracy obtained after learning of each task, which shows the overall performance for the entire continual learning procedure. The *Last* accuracy shows the performance after the continual learning for all seen classes. No task index is provided during inference and we ran each experiment five times and report the average Top-1 classification results.

Compare With Online Methods

We compare our method with existing *replay-based* online approaches including A-GEM [56], GSS [71], MIR [59], ASER [57] and GDUMB [58].

Dataset: We use Split CIFAR-10 [78], Split CIFAR-100 [79] and CORE-50 [80] for evaluation in this part.

- **Split CIFAR-10** splits CIFAR-10 dataset [2] into 5 tasks with each contains 2 disjoint classes. Each class contains 6,000, 32×32 RGB images with originally divided 5,000 for training and 1,000 for testing.
- **Split CIFAR-100** contains 20 tasks with non-overlapping classes constructed using CIFAR-100 [2]. Each task contains 2,500 training images and 500 test images corresponding to 5 classes.
- **CORE-50** is another benchmark dataset for continual learning. For class incremental setting, it is divided into 9 tasks and has a total of 50 classes with 10 classes in the first task and 5 classes in the subsequent 8 tasks. Each class has around 2,400, 128×128 RGB training images and 900 testing images.

Implementation detail: A small version of ResNet-18 (reduced ResNet-18) [56], [68] pretrained on ImageNet [1] is applied as the backbone model for all the compared methods. The ResNet implementation follows the setting as suggested in [53]. We emphasize that only our method freeze the parameters in backbone network while others do not. We apply SGD optimizer with a mini-batch size of 10 and a fixed learning rate of 0.1. We vary the size of exemplar set for $Q \in \{1000, 2000, 5000, 10000\}$ for comparisons.

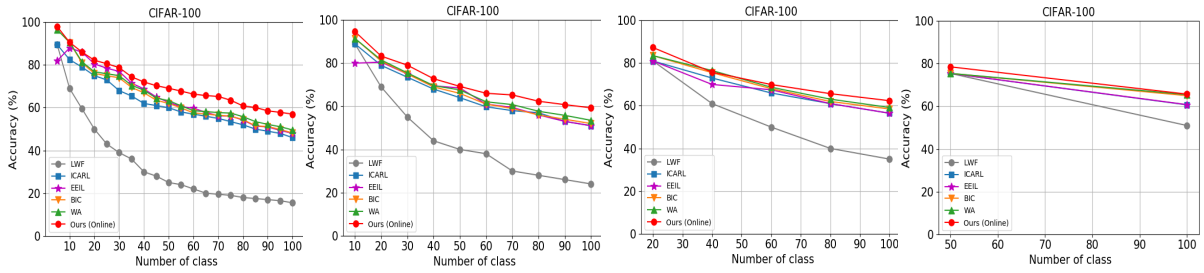


Figure 2.12. Results on CIFAR-100 by comparing with offline approaches with step size (a) 5, (b) 10, (c) 20 and (d) 50. Note that only our method is implemented in online. (Best viewed in color)

Results on Benchmark Datasets

The average accuracy (*Avg*) and last step accuracy *Last* on Split CIFAR-10 and CORE-50 are summarized in Table 2.3. Given different exemplar size Q, our method outperforms existing online approaches, especially when Q is smaller by a larger margin, *i.e.*, our method performs better even with limited storage capacity. The reason is that our approach does not solely rely on exemplars to retain old knowledge but maintains the classifier's discriminability for each learned task and makes the prediction through candidates selection and prior incorporation. In addition, our method includes the exemplar augmentation step, which is more effective given limited number of exemplars as analyzed in Section 2.2.4. In addition, Figure 2.11 visualizes the results for continual learning of 20 tasks on Split CIFAR-100. The model is evaluated after learning each task on test data belonging to all classes seen far. Our method achieves the best performance for each step and we observe that A-GEM [56] does not work well under class-incremental setting, which only use stored exemplars to restrict the update of corresponding parameters while others perform knowledge replay by combining with new class data.

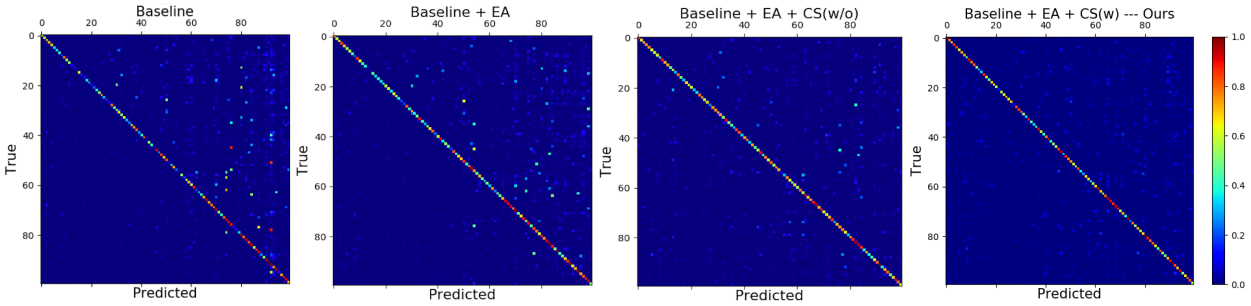


Figure 2.13. Confusion matrices on Split CIFAR-100 for different variants in ablation study. (Best viewed in color)

Compare With Offline Methods

While focusing on online continual learning, we also compare our method with offline continual learning approaches that use each data multiple times to update the model. Although it is widely acknowledged that performance in the online scenario is worse than offline

as discussed in [58], [68] due to the limited number of available new data and each data is observed only once by the model, we show that our method implemented in online scenario is also effective to achieve comparable performance with state-of-the-arts offline approaches including LWF [41], ICARL [46], EEIL [47], BIC [10] and WA [60] following the benchmark protocol similar in [46].

Datasets: We use CIFAR-100 [2] for evaluation and arrange it into splits of 5, 10, 20, and 50 non-overlapped classes, resulting in 20, 10, 5, and 2 tasks, respectively.

Implementation detail: For experiments on CIFAR-100, we apply ResNet-50 [53] pretrained on ImageNet [1] as the backbone model. We apply SGD optimization with mini-batch size of 10 and a fixed learning rate of 0.1 for our method implemented in online scenario. For all the experiments, we arrange classes using identical random seed [46] and use fixed size of exemplar set as $Q = 2,000$.

Results on CIFAR-100

We implement our proposed method in online scenario to use each data only once for training (except for the first task, which is learned in offline under this protocol), while all the compared existing methods are implemented in offline for all tasks. The results on CIFAR-100 for each incremental step are shown in Figure 2.12. Our method still achieves the best results for all incremental step sizes particularly for smaller step size. One of the reasons is that the weight bias problem becomes more severe with smaller incremental step size (more incremental steps) especially in offline case where the model is updated multiple times for each step, which is analyzed in Section 2.2.4. However, this problem is alleviated in online scenario by our proposed learning strategies to pair each new data with an exemplar as described in Section 2.2.3. Furthermore, our method for inference further mitigate the bias problem by selecting candidates and incorporating prior information using stored exemplars, which is illustrated later in Section 2.2.4.

Table 2.4. Average accuracy (%) for ablation study on Split CIFAR-10, Split CIFAR-100 and CORE-50. Best results (except upper-bound) are marked in bold.

Method	**CIFAR-10**	**CIFAR-100**	**CORE-50**
Baseline	56.2	16.7	19.8
Baseline + EA	58.9	20.1	22.4
Baseline + EA + CS(w/o)	81.7	49.6	43.9
Baseline + EA + CS(w) - Ours	**84.9**	**52.0**	**50.7**
Upper-bound	92.2	70.7	67.9

Table 2.5. Performance of exemplar augmentation step for the exemplar size $Q \in \{1000, 5000, 10000\}$. Average accuracy (%) and the corresponding improvements compared with baseline are reported. Highest improvements are marked in bold for each dataset.

Method	**CIFAR-10**	**CIFAR-100**	**CORE-50**
Baseline (Q=1,000)	46.6	13.9	17.2
Baseline + EA	49.8 (**+3.2**)	18.5 (**+4.6**)	20.6 (**+3.4**)
Baseline (Q=5,000)	54.9	23.8	25.4
Baseline + EA	56.2 (+1.3)	25.4 (+1.6)	26.9 (+1.5)
Baseline (Q=10,000)	57.2	26.8	31.4
Baseline + EA	58.1 (+0.9)	27.4 (+0.6)	31.9 (+0.5)

Ablation Study

We also conduct ablation study to analyze the effectiveness of each component in our proposed method including *exemplar augmentation in feature space* (EA) and *candidates selection with prior incorporation* (CS) as illustrated in Section 2.2.3 and 2.2.3, respectively. Specifically, we consider the following variants of our method.

Baseline: remove both CS and EA from our method while keeping exemplar set

Baseline + EA: perform exemplar augmentation

Baseline + EA + CS(w/o): select candidates using stored exemplar but without prior incorporation, which completely trusts the result of PNN by assigning the class of the closest store example as final prediction

Baseline + EA + CS(w): Our proposed method with prior incorporation using Equation 2.13 We also include **Upper-bound** for comparison, which is obtained by training a model in non-incremental setting using all training samples from all classes together. We fix the size of exemplar set for $Q = 2,000$ and the average accuracy are summarized in Table 2.4.

We observe large improvements by adding candidates selection step and our proposed prior incorporation method outperforms directly using PNN output as prediction. The main reason is that the stored feature embeddings extracted by a fixed pre-trained model may not be discriminative enough to make decision especially when there exists obvious distribution difference between the training and testing data as in CORE-50 [80], where the data are collected in distinct sessions (such as indoor or outdoor). Therefore, our proposed prior incorporation step mitigate this problem and achieves the best performance. In addition, we also provide confusion matrices as shown in Figure 2.13 to analyze the results in detail where the **Baseline** tends to predict new classes more frequently and ours is able to treat new classes and old classes more fairly. Finally, we analyze the exemplar augmentation (EA) by varying exemplar size Q and results are summarized in Table 2.5. Our EA works more efficiently given limited storage capacity, which is one of the most significant constraints to apply continual learning in real world applications.

Weight Bias And Storage Consumption

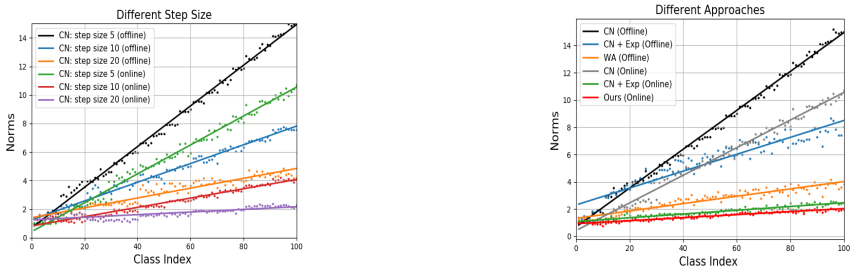
In this section, we implement additional experiments to show the advantages of our proposed method in online scenario including the analysis of norms of weight vectors in classifier and the comparisons of storage consumption.

Norms of weight vectors: One of the main reasons for catastrophic forgetting is the weights in trained model's FC layer are heavily biased towards new classes, which is already discussed in offline mode [10], [60] but lacks sufficient study in online scenario. Therefore, we provide analysis for the impact on biased weights in online and offline scenarios by (1) varying incremental step size and (2) with or without using exemplar set (Exp). For generality, we consider **CN** and **CN + Exp** as two baseline methods using regular cross entropy for continual learning without and with exemplars, respectively. We use CIFAR-100 with step size 5, 10 and 20 for experiments. We train 70 epochs in offline as in [46], [47] and 1 epoch in online scenario for each learning step. Results are shown in Figure 2.14. Each dot corresponds to the norm of the weight vectors in FC layer for each class. For better

visualization, we fit the dots using linear least square to show the trend of each method when new classes are added sequentially.

We observe that the weight bias problem is getting more severe when the number of incremental steps increases, especially in offline case since we repeatedly update model using only new class data. The overall performance in online scenario is much better than offline as each data is used only once for training.

Next, we show that using exemplars is effective to correct biased weights in both online and offline scenario as indicated by **CN+EXP** compared to **CN**. We additionally compare baseline methods with our methods **Ours** and applying Weight Aligning [60] denoted as **WA** for bias correction. The performance of using exemplars in online scenario is even better than applying WA in offline case and our proposed strategy further alleviate this problem. Both analysis explain the larger gains we achieved for smaller step size on CIFAR-100 as discussed in Section 2.2.4. The comparison between online and offline results also show the potential to address catastrophic forgetting in online scenario with the benefit of reduced weight bias problem.



(a) (b)

Figure 2.14. Norms of the weight vectors for (a) the impact of different step size 5, 10, and 20. (b) Impact of different methods using step size 5. The solid line is obtained by linear least square to show the trend for each case.

Storage consumption: Storage requirement poses significant constrains for continual learning in online mode. If we can store all data seen so far without considering storage requirement in real world scenario, then we can easily update the model using all available data. Therefore, we compare the storage consumption of our method with existing

approaches to show the significant reduction in storage requirement. Let S denote the image size, C denote the number of total classes seen so far, Q refers to the number of data stored in exemplar set for each class and D denotes the dimension of extracted feature embedding. (1) For methods using original data as exemplars [10], [21], [46], [47], [56], [58]–[60], [68], [71]–[73], the storage requirement for storing data in exemplar set is $O(3 \times S^2 \times Q \times C)$. (2) For methods which store statistics of old classes and conduct pseudo rehearsal [74], [75], the total cost is $O(D^2 \times C)$ (3) For our method that store feature embeddings as exemplars, the total storage is $O(D \times C \times Q)$. Therefore, as $Q \ll D < 3 \times S^2$, our method requires the least storage while still achieving the best performance.

3. UNSUPERVISED CONTINUAL LEARNING

3.1 Unsupervised Continual Learning Via Pseudo Labels

3.1.1 Overview

The success of many deep learning techniques rely on the following two assumptions: 1) training data is identically and independently distributed (*i.i.d.*), which rarely happens if new data and tasks arrive sequentially over time, 2) labels for the training data are available, which requires additional data annotation by human effort, and can be noisy as well. Continual learning has been proposed to tackle issue #1, which aims at learning new tasks incrementally without forgetting the knowledge on all tasks seen so far. Unsupervised learning focuses on addressing issue #2 to learn visual representations used for downstream tasks directly from unlabeled data. However, unsupervised continual learning, which is expected to tackle both issues mentioned above, has not been well studied [81]. Therefore, we introduce a simple yet effective method in this work that can be adapted by existing supervised continual learning approaches in unsupervised setting where no class label is required during the learning phase. We focuses on image classification task under the class-incremental setting [5] and the objective is to learn from unlabeled data for each incremental step while providing semantic meaningful clusters on all classes seen so far during inference.

Current continual learning approaches can be generally summarized into three categories including (1) *Regularization based*, (2) *Bias-correction based* and (3) *Rehearsal based*. Our proposed method can be directly embedded into existing supervised approaches in category (1) and (2) with an additional step to extract features of unlabeled data and perform clustering to obtain pseudo label. However, for methods in (3), selecting exemplars from learned tasks when class label is not provided in unsupervised scenario is still an unsolved and challenging step. In this work, we tackle this issue by sampling the unlabeled data from the centroid of each generated cluster as exemplars to incorporate with *Rehearsal based* approaches.

In this work, we adopt K-means [12] as our global clustering algorithm for illustration purpose and we propose to use the continual learning model (except the last fully connected layers) at every incremental step for feature extraction of unlabeled data to obtain pseudo

label. The exemplars used for *Rehearsal based* approaches are selected after applying k-means from each generated cluster based on the distance to cluster centroid without requiring the class labels. Note that we are not proposing new approach to address catastrophic forgetting for continual learning in this work, but instead we test the effectiveness of using pseudo labels to make existing supervised methods feasible in unsupervised setting. Therefore, we incorporate our method with existing representative supervised approaches from all three categories mentioned above including LWF [41], ICARL [46], EEIL [47], LUCIR [64], WA [60] and ILIO [21]. We show promising performance in unsupervised scenario on both CIFAR-100 [2] and ImageNet (ILSVRC) [1] datasets compared with results in supervised case that do require the ground truth for continual learning.

3.1.2 Related Work

The major challenge for continual learning is catastrophic forgetting [3] where the model quickly forgets already learned knowledge due to the unavailability of old data during the learning phase of new tasks. Many effective techniques have been proposed to address catastrophic forgetting in supervised scenario, which can be divided into three main categories: (1) *Regularization based* methods aim to retain old knowledge by constraining the change of parameters that are important for old tasks. Knowledge distillation loss [9] is one of the representatives, which was first applied in [41] to transfer knowledge using soft target distribution from teacher model to student model. Later the variants of distillation loss proposed in [21], [64] are shown to be more effective by using stronger constraints. (2) *Bias-correction based* strategy aims to maintain the model performance by correcting the biased parameters towards new tasks in the classifier. Wu *et al.* [10] proposed to apply an additional linear layer with a validation sets after each incremental step. Weight Aligning (WA) is proposed in [60] to directly correct the biased weights in the FC layer, which does not require extra parameters compared with previous one. (3) *Rehearsal based* methods [46], [47] use partial data from old tasks to periodically remind model of already learned knowledge to mitigate forgetting. However, all these methods require class label for the continual learning process, which limits their applications in real world. Therefore, in this work we propose to

use pseudo label obtained from cluster assignments to make existing supervised approaches feasible in unsupervised mode.

For unsupervised learning, many approaches have been proposed to learn visual representation using deep models with no supervision. *Clustering* is one type of unsupervised learning methods that has been extensively studied in computer vision problems [82], [83], which requires little domain knowledge from unlabeled data compared with self-supervised learning [84]. Caron *et al.* [82] proposed to iteratively cluster features and update model with subsequently assigned pseudo labels obtained by applying standard clustering algorithm such as K-means [12]. The most recent work [83] propose to perform clustering and model update simultaneously to address the model's instability during training phase. However, all these existing methods only work on static datasets and are not capable of learning new knowledge incrementally. In addition, the idea of using pseudo label is also rarely explored under continual learning context where the learning environment changes a lot since we need to address catastrophic forgetting as well besides learning visual representation from unlabeled data. In this work, we propose to use the fixed pseudo label for unsupervised continual learning. We also show that iteratively perform clustering to update pseudo labels will result in performance degradation under continual learning context.

3.1.3 Proposed Method

In this work, we propose a simple yet effective method for unsupervised continual learning using pseudo label obtained based on cluster assignments. The updated model after learning each task is evaluated to provide semantic meaningful clusters on all classes seen so far.

For illustration purpose, we adopt k-means as our global clustering algorithm to generate cluster assignments and obtain pseudo label. Then, we demonstrate how to easily incorporate our method with existing supervised approaches in Section 3.1.3.

Clustering: Obtain Pseudo Label

Clustering is one of the most common methods for unsupervised learning, which requires little domain knowledge compared with self-supervised techniques. We focus on using a

general clustering method such as K-means [12], while we also provide the experimental results using other clustering methods as illustrated in *Appendix*, which indicates that the choice is not critical for continual learning performance in our setting. Specifically, K-means algorithm learns a centroid matrix $\boldsymbol{C}$ together with cluster assignments $\tilde{a}_k$ for each input data $\mathbf{x}_k$ by iteratively minimizing $\frac{1}{N}\sum_{k=1}^{N}||\mathbf{h}_{fe}(\mathbf{x}_k) - \boldsymbol{C}\tilde{a}_k||_2^2$, where $\mathbf{h}_{fe}$ refers to the feature extractor. Let m and n represent the number of learned classes and new added classes respectively, then we have $\tilde{a}_k \in \{1, 2, ..., n\}$ and the pseudo label $\tilde{Y}$ for continual learning is obtained by $\{\tilde{y}_k = \tilde{a}_k + m | k = 1, 2, ..\}$ and $\tilde{y}_k \in \{m+1, m+2, ..., m+n\}$.

Learning visual representation from unlabeled data using pseudo label is proposed in [82], which iteratively performs clustering and updating the feature extractor. However, they are not capable of learning new classes incrementally and the learning environment changes under continual learning context as we need to maintain the learned knowledge as well as learning from new tasks. Therefore, in this work we propose to apply the model, $\mathbf{h}_{fe} = \mathbf{h}_{\mathrm{i}-1}$, obtained after incremental step $\mathrm{i}-1$ (except the last fully connected layer) as the feature extractor for incremental step i to extract feature embeddings on all unlabeled data belonging to the new task. Next, we apply k-means based on extracted features to generate cluster assignments and use the fixed pseudo label $\tilde{Y}$ to learn from new task during the entire incremental learning step i. We show in our experiments later that alternatively performing clustering and use pseudo label to update the model as in [82] will result in performance degradation which is discussed in Section 3.1.5. Note that we assume $\mathbf{h}_1$ is obtained from $\mathcal{T}^1$ in supervised mode, so in this work we mainly focus on how to incrementally learn new classes from unlabeled data while maintaining performance on all old classes seen so far.

Incorporating into Supervised Approaches

The obtained pseudo label $\tilde{Y}$ can be easily incorporated with *Regularization-based* methods using knowledge distillation loss or its variants. The distillation loss is formulated by Equation 3.1

$$L_D = \frac{1}{N}\sum_{k=1}^{N}\sum_{r=1}^{m} -\hat{p}_T^{(r)}(\mathbf{x}_k) log[p_T^{(r)}(\mathbf{x}_k)] \tag{3.1}$$

$$\hat{p}_T^{(r)} = \frac{\exp\left(\hat{o}^{(r)}/T\right)}{\sum_{\mathrm{j}=1}^{m} \exp\left(\hat{o}^{(\mathrm{j})}/T\right)} \ , \ p_T^{(r)} = \frac{\exp\left(o^{(r)}/T\right)}{\sum_{\mathrm{j}=1}^{m} \exp\left(o^{(\mathrm{j})}/T\right)}$$

where $\hat{o}^{m\times 1}$ and $o^{m\times 1}$ denote the output logits of student and teacher models respectively for the m learned classes. T is the temperature scalar used to soften the probability distribution. The cross entropy loss to learn the added n new classes can be expressed as

$$L_C = \frac{1}{N} \sum_{k=1}^{N} \sum_{r=1}^{n+m} -\tilde{y}_k^{(r)} log[p^{(r)}(\mathbf{x}_k)] \tag{3.2}$$

where $\tilde{y}_k \in \tilde{Y}$ is the obtained pseudo label for data $\mathbf{x}_k$ instead of the ground truth labels in supervised case. Then the cross-distillation loss combining cross entropy L_C and distillation L_D is formulated in Equation 3.3 with a hyper-parameter $\alpha = \frac{m}{m+n}$ to tune the influence between two terms.

$$L_{CD}(\mathbf{x}) = \alpha L_D(\mathbf{x}) + (1-\alpha) L_C(\mathbf{x}) \tag{3.3}$$

Herding dynamic algorithm [51] is widely applied for *Rehearsal based* methods to select exemplars based on class mean in supervised case. However, since no class label is provided in unsupervised scenario, we instead propose to select exemplars based on cluster mean. The exemplar set Q stores the data and pseudo label pair denoted as $(\mathbf{x}_k, \tilde{y}_k)$.

The incorporation with *Bias-correction based* methods is the most straightforward. BIC [10] applies an additional linear model for bias correction after each incremental step using a small validation set containing balanced old and new class data. In our unsupervised scenario, both the training and validation set used to estimate bias can be constructed using obtained pseudo label instead of the ground truth. The most recent work WA [60] calculates the norms of weights vectors in FC layer for old and new class respectively and use the ratio to correct bias without requiring extra parameters. Thus our method can be directly embedded with it by an addition step to obtain pseudo label as illustrated in Section 3.1.3.

We emphasize that we are not introducing new method to address catastrophic forgetting, but rather investigating whether it is possible to use pseudo labels instead of ground truth labels for continual learning. We show in Section 4.1.4 that our proposed method works effectively with existing approaches from all categories mentioned above.

3.1.4 Experimental Results

In this section, we evaluate our proposed method from two perspectives. 1) We incorporate with existing approaches and compare results obtained in unsupervised and supervised cases to show the ability of using pseudo labels for unsupervised continual learning to provide semantic meaningful clusters for all classes seen so far. 2) We analyze the effectiveness of each component in our proposed method including the exemplar selection and the choice of feature extractor in unsupervised scenario. These experimental results are presented and discussed in Sections 3.1.5 and 3.1.5, respectively.

Benchmark Experimental Protocol

Although different benchmark experimental protocols are used in supervised case [21], [46], [64], there is no agreed protocol for evaluation of unsupervised continual learning methods. In addition, various learning environments may happen when class label is not available so it is impossible to use one protocol to evaluate upon all potential scenarios. Thus, our proposed new protocol focuses on class-incremental learning setting and aims to evaluate the ability of unsupervised methods to learn from unlabeled data while maintaining the learned knowledge during continual learning. Specifically, the following assumptions are made: (1) all the new data belong to new class, (2) the number of new added class (step size) is fixed and known beforehand, (3) no class label is provided for learning (except for the initial step) and (4) the updated model should be able to provide semantic meaningful clusters for all classes seen so far during inference. Our protocol is introduced based on current research progress for supervised class-incremental learning and three benchmark datasets are considered including (i) CIFAR-100 [2] with step size 5, 10, 20, 50 (ii) ImageNet-1000 (ILSVRC) [1] with step size 100 and (iii) ImageNet-100 (100 classes subset of ImageNet-1000) with step size 10. Top-1 and Top-5 ACC are used for CIFAR-100 and ImageNet, respectively.

Evaluation Metrics

We evaluate our method using cluster accuracy (ACC), which is widely applied in unsupervised setting [82], [85] when class label is not provided. We first find the most represented class label for each cluster using Hungarian matching algorithm [86], and then calculate the accuracy as $\frac{N_c}{N}$ where N is the total number of data and N_c is the number of correctly classified data. Note that the classification accuracy used in supervised setting is consistent with cluster accuracy and is widely used for performance comparison in unsupervised case as in [85]. In this work, ACC is used to evaluate the model's ability to provide semantic meaningful clusters.

3.1.5 Implementation Detail

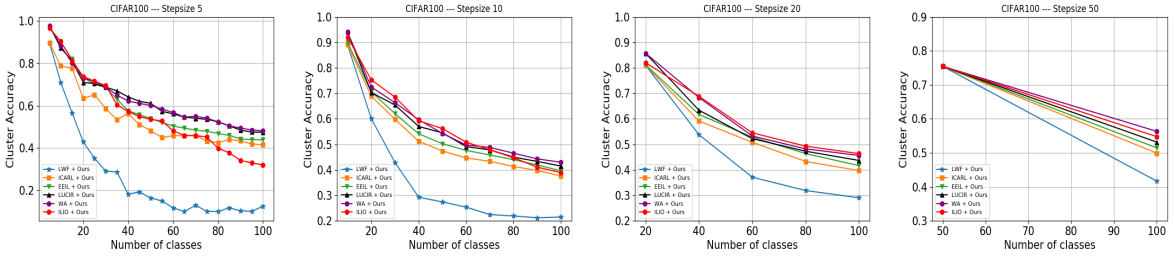
Our implementation is based on Pytorch [55] and we use ResNet-32 for CIFAR-100 and ResNet-18 for ImageNet. The ResNet implementation follows the setting as suggested in [53]. The setting of incorporated existing approaches follows their own repositories. We select $q = 20$ exemplars per cluster to construct exemplar set and arrange classes using identical random seed (1993) with benchmark supervised experiment protocol [46]. We ran five times for each experiment and the average performance is reported.

Incorporating with Supervised Approaches

In this part, our method is evaluated when incorporated into existing supervised approaches including **LWF** [41], **ICARL** [46], **EEIL** [47], **LUCIR** [64], **WA** [60] and **ILIO** [21], which are representative methods from all *Regularization based*, *Bias-correction based* and *Rehearsal based* categories as described in Section 4.2.2. Note that **ILIO** is implemented in online scenario where each data is used only once to update model while others are implemented in offline. We embed the pseudo label to evaluate the performance of selected approaches in unsupervised mode. *E.g.* **ICARL + Ours** denotes the implementation of **ICARL** in unsupervised mode by incorporating with our proposed method. Table 3.1 summarizes results in terms of last step ACC (Last) and average ACC (Avg) calculated

Datasets	CIFAR-100								ImageNet			
Step size	5		10		20		50		10		100	
ACC	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
LWF (w/)	0.299	0.155	0.393	0.240	0.465	0.352	0.512	0.512	0.602	0.391	0.528	0.374
LWF+Ours (w/o, Δ)	-0.071	-0.029	-0.091	-0.025	-0.086	-0.062	-0.095	-0.095	**-0.033**	-0.053	-0.211	-0.174
ICARL (w/)	0.606	0.461	0.626	0.518	0.641	0.565	0.607	0.607	0.821	0.644	0.608	0.440
ICARL+Ours (w/o, Δ)	-0.084	-0.045	-0.135	-0.142	-0.158	-0.174	-0.108	-0.108	**-0.043**	-0.047	-0.197	-0.015
EEIL (w/)	0.643	0.482	0.638	0.517	0.637	0.565	0.603	0.603	0.893	0.805	0.696	0.520
EEIL+Ours (w/o, Δ)	-0.071	-0.043	-0.131	-0.121	-0.131	-0.148	-0.088	-0.088	**-0.040**	-0.064	-0.199	-0.154
LUCIR (w/)	0.623	0.478	0.631	0.521	0.647	0.589	0.642	0.642	0.898	0.835	0.834	0.751
LUCIR+Ours (w/o, Δ)	**-0.015**	-0.003	-0.104	-0.106	-0.131	-0.152	-0.111	-0.111	**-0.037**	-0.083	-0.293	-0.342
WA (w/)	0.643	0.496	0.649	0.535	0.669	0.592	0.655	0.655	0.905	0.841	0.859	0.811
WA+Ours (w/o, Δ)	**-0.034**	-0.014	-0.110	-0.106	-0.121	-0.136	-0.092	-0.092	**-0.037**	-0.056	-0.295	-0.376
ILIO (w/)	0.664	0.515	0.676	0.564	0.681	0.621	0.652	0.652	0.903	0.845	0.696	0.601
ILIO+Ours (w/o, Δ)	-0.123	-0.194	-0.140	-0.175	-0.134	-0.157	-0.106	-0.106	-0.057	-0.118	-0.178	-0.212

by averaging ACC for all incremental steps, which shows overall performance for the entire continual learning procedure. We also report the performance difference $\Delta = w/ - w/o$ and observe only small degradation by comparing unsupervised results with supervised results. In addition, we calculate the average accuracy drop by $Avg(\Delta) = Avg(w/) - Avg(w/o)$ for each incremental step corresponds to each method. The $Avg(\Delta)$ ranges from $[0.015, 0.295]$ with an average of 0.114. Our method can work well with but not limited to these selected representative methods and we achieve competitive performance in unsupervised scenario without requiring human annotated labels during continual learning phase. Figure 3.1 shows cluster accuracy for each incremental step on CIFAR-100.



(a) (b) (c) (d)

Figure 3.1. Results on CIFAR-100 with step size 5, 10, 20, and 50 by incorporating our method with existing approaches to realize continual learning in unsupervised scenario. (Best viewed in color)

Ablation Study

We conduct extensive experiments to **1)** analyze the unsupervised exemplar selection step as described in Section 3.1.3 by varying the number of exemplars per class and compare the results with random selection. **2)** Study the impacts of different methods that can be used to extract feature for clustering to obtain pseudo label during continual learning. For both experiments, we first construct our baseline method denoted as **Ours**, which uses distillation loss as described in Equation 3.3 and exemplars from learned tasks.

For part **1)**, we vary the target number of exemplars per class $q \in \{10, 20, 50, 100\}$ and compare the results with random exemplar selection from each generated cluster, denoted as **Random**. The results on CIFAR-100 are shown in Figure 3.2. We observe that the overall performance will be improved by increasing q even using randomly selected exemplars. In addition, our proposed method, which selects exemplars based on cluster mean, outperforms **Random** by a larger margin when q becomes larger.

For part **2)**, we compare our method using the updated model from last incremental step as feature extractor with i) **Scratch**: apply a scratch model with the same network architecture as feature extractor, ii) **PCA**: directly apply PCA algorithm [87] on input images to obtain feature embeddings for clustering, iii) **Fixed Feature Extractor (FFE)**: use model $\mathbf{h}_1$ as described in Section 3.1.3 as the fixed feature extractor for the entire continual learning process, iv) **Updated Pseudo Label (UPL-K)**: iteratively update model and perform clustering within each incremental step as proposed in [82], where K indicates how frequently we update the pseudo label *e.g.* UPL - 10 means we update pseudo label for every 10 epochs. All these variants are modified based on our baseline method. Results are summarized in Table 3.2. The scratch method provides lower bound performance and FFE outperforms PCA by a large margin, showing the advanced ability of using deep models to extract more discriminative feature for clustering. Note that we did not perform PCA on ImageNet-1000 as it takes quite a long time for computation. Comparing UPL-K with $K = 0, 10, 20, 30$ ($K = 0$ is Ours), we observe that if the updating frequency increases (K decreases), the overall performance degrades. As discussed in Section 3.1.3, different from unsupervised representation learning that uses a model from scratch, in continual learning we

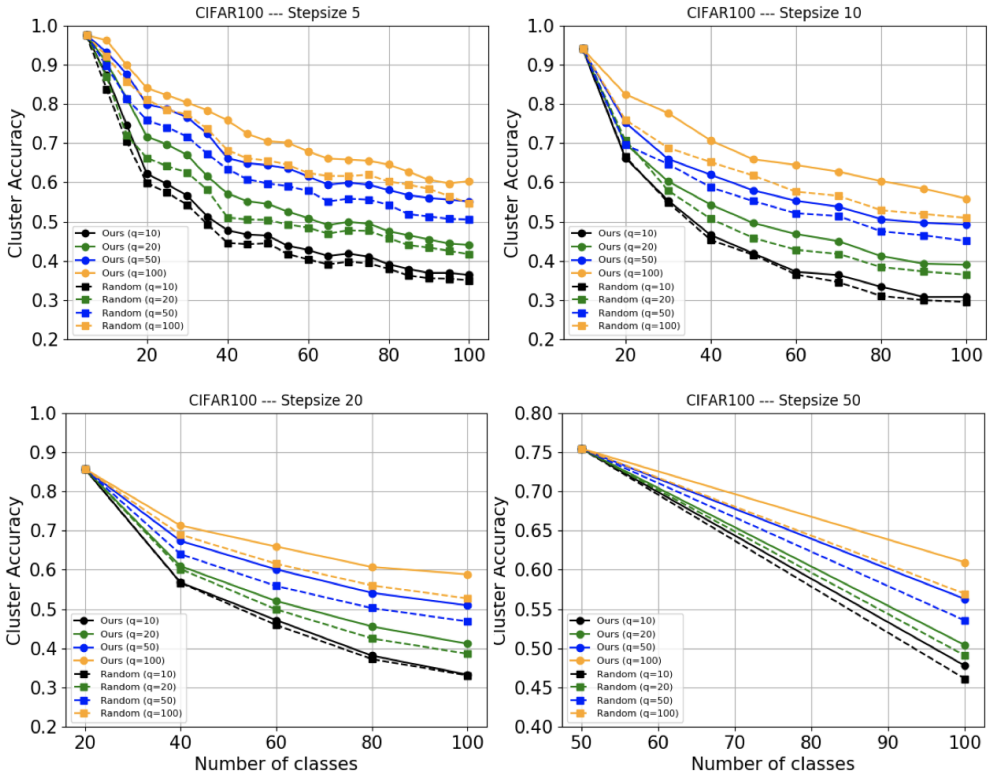


Figure 3.2. Results on CIFAR-100 by varying target exemplar size $q \in \{10, 20, 50, 100\}$ and comparison with random selection.

also need to preserve the learned knowledge for all classes seen so far and update pseudo label repeatedly will accelerate the catastrophic forgetting, resulting in the performance drop.

3.2 Out-Of-Distribution Detection In Unsupervised Continual Learning

3.2.1 Overview

Unsupervised continual learning is an emerging future learning system, capable of learning new tasks incrementally from unlabeled data. It requires neither static datasets nor human annotations compared with supervised offline learning. Existing methods study this problem under the assumption that all new data belongs to new tasks. We argue that if human annotation is not available as common in unsupervised scenario, we cannot know whether the unlabeled new data belongs to new or learned tasks. For example, an image-based mobile food recognition system should be able to distinguish new and learned food

Table 3.2. Ablation study for different approaches to obtain pseudo labels on CIFAR-100 and ImageNet in terms of average ACC (Avg) and last step ACC (Last). The best results are marked in bold.

Datasets	CIFAR-100								ImageNet			
Step size	5		10		20		50		10		100	
ACC	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
Scratch	0.106	0.038	0.095	0.015	0.122	0.038	0.226	0.226	0.282	0.158	0.069	0.023
PCA	0.156	0.085	0.143	0.061	0.171	0.083	0.287	0.287	0.308	0.175	/	/
FFE	0.459	0.338	0.399	0.281	0.401	0.323	0.392	0.392	0.757	0.620	0.405	0.275
UPL-10	0.498	0.376	0.415	0.293	0.430	0.320	0.401	0.401	0.797	0.653	0.446	0.294
UPL-20	0.523	0.394	0.422	0.296	0.445	0.339	0.413	0.413	0.816	0.699	0.458	0.311
UPL-30	0.513	0.383	0.435	0.324	0.459	0.364	0.433	0.433	0.832	0.705	0.460	0.332
Ours	**0.558**	**0.426**	**0.482**	**0.368**	**0.486**	**0.397**	**0.495**	**0.495**	**0.849**	**0.722**	**0.471**	**0.342**

images first instead of blindly treating all of them as new food classes to perform unsupervised continual learning for update. Therefore, in order to make unsupervised continual learning work in practical problems, an out-of-distribution (OOD) detector should be required at the beginning of each incremental learning step to identify whether each data belongs to new or already learned tasks. However, the problem of OOD detection in continual learning still remains under-explored, *i.e.* none of the existing OOD detection methods target for continual learning.

The goal of OOD detection for image classification is to detect novel classes data. However, it becomes more challenging under continual learning scenario due to (1) the training data of learned tasks becomes unavailable; (2) we also need to address catastrophic forgetting problem [3]. Most existing methods cannot be applied here because they either require all training data for already learned tasks to train an OOD detector [88]–[90], or they need to modify the training procedure and objectives [91]–[94], which may sacrifice the classification accuracy. Therefore, we focus on "post-hoc" methods [95] that can be directly applied on any trained classification models to perform OOD detection based on the output confidence, which has been widely adopted in real-world environments to avoid the need to access training data.

The central idea of "post-hoc" methods to perform OOD detection is to assign in-distribution (ID) data with higher confidence value $Conf_{in}$ than the OOD data $Conf_{out}$ based on the output vector where the confidence $Conf$ is defined as the maximum of softmax output [96], [97] or the energy score [98]. The detection performance greatly depends on the

difference value of output confidence between ID and OOD data $\mathcal{D}_c = Conf_{in} - Conf_{out}$ where higher $\mathcal{D}_c$ indicates better discrimination. However, there exists two major issues in continual learning scenario that can lead to the decrease of $\mathcal{D}_c$ including (1) the biased output value towards new classes as revealed in [10], [60]; (2) the decrease of output confidence compared with offline learning due to the objective of improving generalization ability to mitigate catastrophic forgetting [41], [99]. Both issues can result in performance degradation for existing "post-hoc" methods.

In this work, we first formulate the OOD detection in unsupervised continual learning scenario denoted as OOD-UCL and introduce the corresponding evaluation protocol. Then, we propose a novel OOD detection method that can address both issues mentioned above to achieve improved performance in unsupervised continual learning scenario.

3.2.2 Related Work

We focus on image classification problem and we review the existing methods that are related to our work including (1) unsupervised continual learning; (2) OOD detection.

Unsupervised Continual Learning

Compared with supervised case, unsupervised continual learning has not received much attention [81]. Stojanov *et al.* [100] introduced an unsupervised object learning environment to learn a sequence of single-class exposures. In addition, CURL [101] and STAM [102] are proposed for task-free unsupervised continual learning where task boundary is not given. Based on existing supervised protocol [46], the most recent work [103] proposed to use pseudo labels obtained based on cluster assignments to perform continual learning and show promising results on several benchmark datasets in unsupervised scenario. However, they only assume a simplified scenario where all the new data belong to new classes, which rarely happens in real life applications when the class labels are not available. Therefore, an OOD detector that can work under unsupervised continual learning scenario becomes indispensable.

Out-of-distribution Detection

We focus on image classification based OOD detection and analyze this problem in continual learning scenario where the training objective is more challenging. Therefore, we target on methods that can be applied to any trained classification model without modifying the training procedure, which is called "post-hoc" methods [95]. Existing "post-hoc" methods are originated from [96], which directly uses the maximum softmax probability as the confidence score to discriminate ID and OOD data. Then ODIN [97] applies temperature scaling and input perturbation to amplify the confidence difference D_c between ID and OOD data where a large temperature transforms the softmax score back to the logit space. Built on these insights, recent work [98] proposed to use energy score as output confidence for OOD detection, which maps the output to a scalar through a convenient log-sum-exp operator. However, none of the existing "post-hoc" methods consider the two issues in continual learning scenario, resulting in performance degradation.

3.2.3 Problem Formulation

The objective is to perform OOD detection in continual learning scenario to discriminate unlabeled learned tasks data (as ID) and new task data (as OOD), which can be then incorporated into any existing unsupervised continual learning methods to apply in real life applications. We formulate the out-of-distribution in unsupervised continual learning (OOD-UCL) problem based on the existing unsupervised class-incremental learning protocol [103] to evaluate the OOD detection performance before each incremental learning step. Specifically, the continual learning for image classification problem $\mathcal{T}$ can be expressed as learning a sequence of N tasks $\{\mathcal{T}^1, ..., \mathcal{T}^N\}$ corresponding to $(N-1)$ incremental learning steps where the learning of the first task $\mathcal{T}^1$ is not included. Each task contains M non-overlapped classes, which is known as incremental step size. Let $\{D^1, ..., D^N\}$ denote the training data and $\{S^1, ..., S^N\}$ denote the testing data for each task, we formulate the OOD-UCL with the following properties. **Property 1:** The OOD detection is performed at beginning of the learning step for each new task $\mathcal{T}^K$ where $K \in \{2, ...N\}$. The test data belonging to learned tasks $S^{\mathrm{i}}, \mathrm{i} \in \{1, ...K-1\}$ is regarded as ID data and the test data belonging to

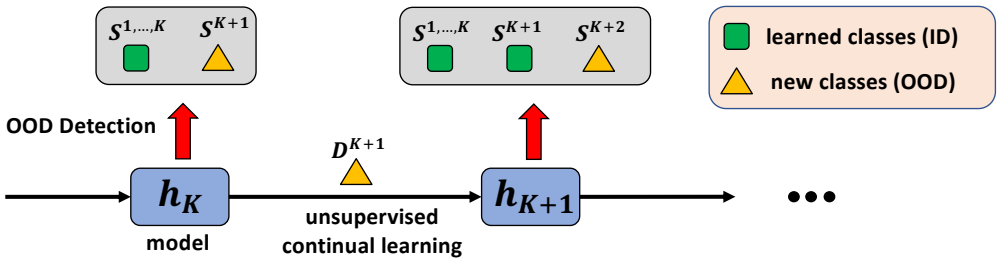


Figure 3.3. Formulation of out-of-distribution detection in unsupervised continual learning (OOD-UCL). h_K refers to the updated incremental models after learning $\mathcal{T}^K$. D^K and S^K denote the corresponding training and testing splits for task K, respectively.

the current incremental step S^K is regarded as the OOD data. Figure 3.3 illustrates the evaluation protocol, where we perform total $(N-1)$ times OOD detection for continually learning a sequence of N tasks $\{\mathcal{T}^1, ..., \mathcal{T}^N\}$.

Property 2: The training data allowed for OOD detection before learning $\mathcal{T}^K$ is restricted to (1) the training set of D^{K-1} and (2) the stored exemplars belonging to $\{\mathcal{T}^1, ..., \mathcal{T}^{K-2}\}$ if applicable. This restricts the usage of most existing methods [88]–[90] which requires all training data for learned classes to train an OOD detector.

Evaluation metrics: In OOD detection, each test data is assigned with a confidence score where samples below the pre-defined confidence threshold are considered as OOD data. By regarding the ID data as positive and OOD data as negative, we can obtain a series of true positives rate (TPR) and false positive rate (FPR) by varying the thresholds. One of the commonly used metrics for OOD detection is **FPR95**, which measures the FPR when the TPR is 0.95 and lower value indicates better detection performance. Besides, we can also calculate the area under receiver operating characteristic curve (**AUROC** [104]) based on FPR and TPR as well as the area under the precision-recall curve (**AUPR** [105]). For both AUROC and AUPR, a higher value indicates better detection performance.

3.2.4 Proposed Method

In this section, we introduce a novel "post-hoc" OOD detection method with the goal of improving the performance under unsupervised continual learning scenario, *i.e.* increase

the confidence difference D_c between ID and OOD data for better discrimination. The overview of the proposed method is shown in Figure 3.4, which can be directly applied without requiring any change to the existing classification models. There are two main steps including **bias correction** and **confidence enhancement** where we first correct the biased output value and then enhance the confidence difference D_c based on task discriminativeness, which are described in Section 3.2.4 and Section 3.2.4, respectively.

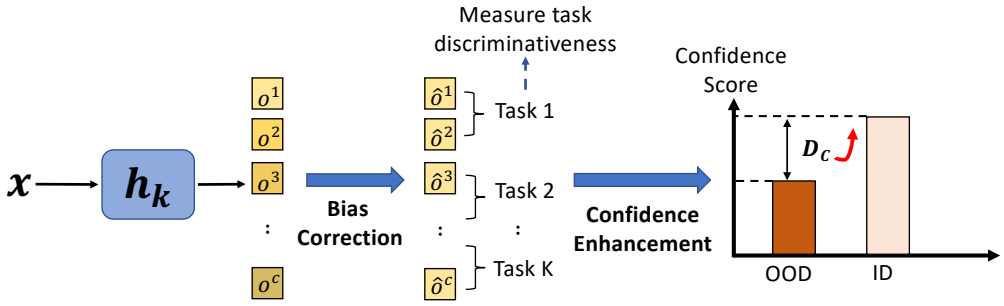


Figure 3.4. The overview of our proposed method where **x** refers to input data and h_K denotes the continual model after learning task $\mathcal{T}^K$. We first correct the bias of output O to obtain $\hat{O}$ and then perform confidence enhancement to further increase the confidence difference D_c to improve OOD detection performance.

Bias Correction Output bias towards new classes is a widely recognized issue [10], [60] caused by the lack of training data for learned tasks during continual learning. This results in the increase of the output value towards the biased classes for both ID and OOD data, therefore decreases the confidence difference D_c, *i.e.* the degradation of OOD detection performance. Motivated by WA [60] which shows the existence of biased weights in the FC classifier, we propose to perform bias correction by normalizing output logits based on the norm of weight vectors in the classifier corresponding to each learned class. Specifically, we denote the weight parameters in the classifier as $P \in \mathcal{R}^{d\times C}$ where d is the dimension of extracted feature of each input sample and C refers to the total number of classes seen so far. The weight norm of P corresponds to each learned class is calculated as

$$|W^{\mathrm{i}}| = L_2(P^{1,\mathrm{i}}, P^{2,\mathrm{i}}, ...P^{d,\mathrm{i}}), \mathrm{i} \in \{1, 2, ...C\} \tag{3.4}$$

where $L_2()$ denotes the l_2 normalization and $P^{\mathrm{j},k}$ refers to the element of j^{th} row and k^{th} column in P. Let $O = \{o^1, o^2, ..., o^C\}$ denote the output from the classifier, we normalize it through

$$\hat{o}^{\mathrm{i}} = o^{\mathrm{i}}/|W^{\mathrm{i}}|, \mathrm{i} \in \{1, 2, ...C\} \tag{3.5}$$

where $\hat{o}^{\mathrm{i}}$ refers to the corrected output for class i. Our weight-based normalization generates the corrected output by efficiently mitigating the bias effect from the classifier.

Confidence Enhancement

The learning objective also changes in continual learning scenario. Besides learning new tasks, we also need to maintain the learned knowledge. As shown in [106], higher confident output can decrease the model's generalization ability, which leads to catastrophic forgetting. Most existing continual learning methods address this problem by adding regularization to restrict the change of parameters [21], [41], [46], [47], [64], [107] when learning new tasks, which decrease the output confidence for both ID and OOD data, resulting in the decrease of confidence difference D_c. Our goal is to increase D_c to achieve better detection performance. Our proposed confidence enhancement method is motivated by the most recent work [108], [109], which show that the continual learning model is able to maintain the discriminativeness within each learned task. Ideally, an ID data should be more confident and task-discriminative than OOD data. Therefore, after correcting the biased output, we apply softmax on $\hat{O} = \{\hat{o}^1, \hat{o}^2, ..., \hat{o}^C\}$ to obtain $\hat{S} = \{\hat{s}^1, \hat{s}^2, ..., \hat{s}^C\}$. We extract the maximum value as $\hat{S}_{max} = max(\hat{S})$ and its corresponding task index $I_{max} = argmax_{\mathrm{i}=1,2...K}(\hat{S})$ where K denotes the total number of tasks $\{\mathcal{T}^1, ..., \mathcal{T}^K\}$ learned so far. The softmax output value for task $\mathcal{T}_{I_{max}}$ is extracted from $\hat{S}$ as $\hat{S}_{I_{max}} = \{\hat{s}^1_{I_{max}}, \hat{s}^2_{I_{max}}, ...\hat{s}^M_{I_{max}}\}$ where M refers to the number of classes in each task, *i.e.* the incremental step size. We then measure the discriminativeness based on entropy as in Equation 3.6 where lower entropy H indicates more discriminative.

$$H_{I_{max}} = \sum_{\mathrm{i}=1}^{M} \hat{s}^{\mathrm{i}}_{I_{max}} \times log_M(\hat{s}^{\mathrm{i}}_{I_{max}}) \tag{3.6}$$

Finally, we calculate the confidence score as

$$Conf = \frac{\hat{S}_{max}}{H_{I_{max}} + \epsilon} \tag{3.7}$$

where $\epsilon = 0.00001$ is used for regularization. Test samples assigned with larger score is regarded as ID data.

3.2.5 Experimental Results

Our proposed OOD detection method can work easily with any unsupervised continual learning approach. In this section, we show its effectiveness by incorporating the baseline in [103] to perform unsupervised continual learning. We follow the proposed evaluation protocol by comparing the OOD detection results with existing "post-hoc" methods including **MSP** [96], **ODIN** [97] and **Energy Score** [98]. We run each experiment 5 times and report the average results.

We use the CIFAR-100 [2] dataset and divide the 100 classes into splits of 20, 10 and 5 tasks with corresponding incremental step size 5, 10 and 20, respectively. For unsupervised continual learning baseline [103], we apply ResNet-32 [53] and train 120 epochs for each incremental step and the learning rate is decreased by 1/10 for every 30 epochs. Exemplar size is set as $2,000$. We perform OOD detection at the beginning of each new task except the first one.

Results on CIFAR-100 Table 3.3 shows the average OOD detection results on CIFAR-100 in terms of AUROC, AUPR and FPR95. We observe consistent improvements for OOD detection in unsupervised continual learning scenario compared with existing "post-hoc" methods. Besides, we also include **ours (w/o BC)** and **ours (w/o CE)** for ablation study where *BC* and *CE* denote bias correction and confidence enhancement steps as illustrated in Section 3.2.4. Note that the MSP [96] can be regarded as **ours (w/o BC and CE)**. Thus, both BC and CE improves the detection performance compared with MSP and our method including both steps achieve the best performance. In addition, the AUROC on CIFAR-100 for each incremental step is shown in Figure 3.5. Our method outperforms existing approaches at each step especially with larger margins for smaller step size, as both output

bias and confidence decrease problems become more severe due to the increasing number of incremental learning steps.

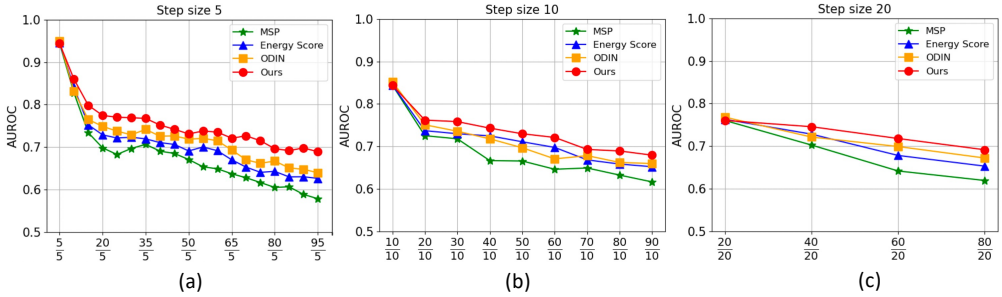


Figure 3.5. Results on CIFAR-100 with step size (a) 5 (b) 10 and (c) 20. The numerator and denominator of x-axis refers to the number of learned classes and new added classes, which are regarded as in-distribution and out-of-distribution data, respectively.

Table 3.3. Average AUROC, AUPR and FPR95 on CIFAR-100 with step size 5, 10 and 20. BC and CE denotes bias correction step and confidence enhancement step, respectively. Best results are marked in bold.

Methods	Step size 5			Step size 10			Step size 20		
	AUROC↑	**AUPR↑**	**FPR95↓**	**AUROC↑**	**AUPR↑**	**FPR95 ↓**	**AUROC↑**	**AUPR↑**	**FPR95↓**
MSP [96]	0.679	0.947	0.855	0.685	0.899	0.873	0.681	0.834	0.877
ODIN [97]	0.723	0.950	0.810	0.715	0.909	0.831	0.715	0.858	0.839
Energy Score [98]	0.707	0.950	0.824	0.714	0.907	0.837	0.706	0.853	0.844
Ours (w/o BC)	0.712	0.951	0.823	0.719	0.912	0.845	0.706	0.851	0.842
Ours (w/o CE)	0.708	0.947	0.836	0.713	0.907	0.851	0.699	0.844	0.854
Ours	**0.754**	**0.959**	**0.793**	**0.736**	**0.915**	**0.824**	**0.729**	**0.874**	**0.814**

4. APPLICATION BASED CONTINUAL LEARNING

4.1 Online Continual Learning For Visual Food Classification

4.1.1 Overview

Food classification serves as the first and most crucial step for image-based dietary assessment [110], which aims to provide valuable insights for prevention of many chronic diseases. Ideal food classification system should be able to update using each new recorded food image continually without forgetting the food class that has been already learned before. Achieving this goal would bring significant advantage for deploying such a system for automated dietary assessment and monitoring.

From the perspective of visual food classification, although recent works [13], [14], [111], [112] have been proposed using advanced deep learning based approaches to increase model performance, they use only static datasets for training and are not capable of handling sequentially available new food classes. Therefore, the classification accuracy could drop dramatically due to the unavailability of old data, which is also known as catastrophic forgetting [3]. Although retraining from scratch is a viable option, it is impractical to do whenever a new food is observed, which is time consuming and require high computation and memory resource especially for large scale food image datasets. For example, a model already learned $1,000$ food classes need to retrain from scratch for only 1 new observed food.

From the perspective of continual learning, an increasing number of approaches [21], [58], [113], [114] have been proposed to address catastrophic forgetting and to learn new knowledge incrementally in online scenario. Compared to offline scenario where data can be used multiple epochs for training, online scenario is more challenging where each new data is observed only once by the model, but is more practical for real-life application such as food image classification system. Representative techniques to mitigate forgetting include (1) storing a small number of learned data as exemplars for replay [46], and (2) applying knowledge distillation [9] using a teacher model to maintain the learned performance. However, continual learning for food image classification is still lacking and there are two major obstacles which make the above mentioned techniques less effective for food images. (i) Food images exhibit higher intra-class variation [14] compared with commonly seen objects in real

life, which is due to different culinary culture and cooking style. Most existing continual learning methods [10], [21], [46], [47], [60], [64] apply herding algorithm [51] to select exemplars for each learned class based on class mean only, which is difficult to cover the diversity for food types within the same class. Therefore, catastrophic forgetting could become worse if stored exemplars are not good representations of learned classes. (ii) The distribution of future food classes is usually unpredictable and imbalanced due to the variance of consumption frequencies [115] among different food categories. Nevertheless, most online approaches only study continual learning on balanced datasets containing the same number of data per class such as CIFAR [2] and MNIST [116] without considering the class-imbalance problem that is common for food images. In addition, as indicated in [67], the knowledge distillation term becomes less effective if teacher model is not trained on balanced data.

In this work, we address the challenging problem of food image classification for online continual learning by first introducing a novel exemplar selection algorithm, which clusters data for each class based on visual similarity and then selects the most representative exemplars from each generated cluster based on cluster mean. We apply Power Iteration Clustering [117], which does not require the number of cluster beforehand. Therefore, our algorithm can adapt to different food categories, *i.e.*, food with higher variation will generate more clusters and vice versa. In addition, we propose an effective online learning regime by using balanced training batch for old and new class data and apply knowledge distillation loss between original and augmented exemplars to better maintain the model performance. Our method is evaluated on a large scale real world food database, Food-1K [13], and outperforms state-of-the-arts including ICARL [46], ER [113], [114], ILIO [21] and GDUMB [58], which are all implemented in online scenario and use exemplars for replay during continual learning.

4.1.2 Related Work

Food Classification

Food classification refers to the task of labeling image with food category, which assumes each input image contains only one single food item. Earlier work [118] use fusion

of features including SIFT [119], Gabor, and color histograms for classification. Later, the modern deep learning models have been widely applied as backbone network to extract more class-discriminative features as in [13], [20], [112], [120]–[124], which significantly improves the performance. Recent works [14], [111] leveraging hierarchy structure based on visual information are able to achieve further improvements. However, all these methods use static food image datasets for training and none of them is capable of learning from sequentially available data, making it difficult to apply in real life applications as new foods are observed over time.

Continual Learning

The major challenge for continual learning is called catastrophic forgetting [3], where the model quickly forgets already learned knowledge due to the unavailability of old data. Below, we review and summarize existing knowledge-preserving techniques that are most relevant to our proposed method.

Replay-based methods store a small number of representative data from each learned class as exemplars to perform knowledge rehearsal during the continual learning. Herding dynamic algorithm [51] is first applied in ICARL [46] to select exemplars that are closer to the class mean. It has gradually became a common exemplar selection strategy that is being used in most existing methods [10], [21], [46], [47], [60], [64], where ICARL adopts a nearest class mean classifier [125] while others use softmax classifier for classification. In addition, reservoir sampling [126] along with random retrieval is applied in Experience Replay (ER) based methods [113], [114], which ensures each incoming data point has the same probability to be selected as exemplar in the memory buffer. A greedy balancing sampler with random selection is recently used in GDUMB [58] to store as much data as memory allowed, which also achieves competitive performance.

Regularization-based methods restrict the impact of learning new tasks on the parameters that are important for learned tasks. Knowledge distillation [9] is a popular representative technique, which makes the model mimic the output distribution for learned classes from a teacher model to mitigate forgetting during continual learning [10], [41], [46], [47], [64], [65],

[103]. For most recent work, He *et al.* proposed ILIO [21], which applies an accommodation ratio to generate a stronger constraint for knowledge distillation loss to achieve improved performance.

However, among these methods, only a few [21], [46], [58], [113], [114] are feasible in online scenario to use each data only once for training. In addition, none of the existing methods focus on food images and as introduced in Section 4.1.1, the high intra-class variance and imbalanced data distribution make both exemplar and distillation based techniques less effective to address catastrophic forgetting. Therefore, we propose a novel exemplar selection algorithm to select exemplars from each generated cluster based on visual similarity to adapt to the variability of different food categories. Besides, we propose an effective online learning regime using balanced training batch and apply distillation on augmented exemplars to better maintain performance on learned classes.

4.1.3 Proposed Method

An overview of our proposed method is illustrated in Figure 4.1, including a novel exemplar selection method and an effective online training regime. Specifically, instead of selecting exemplars based on class mean as in herding [51], we first generate clusters based on similarity and then select exemplars from each cluster using the corresponding cluster mean. During the continual learning phase, each new class data from data stream is paired with one randomly selected exemplar from exemplar set to produce balanced training batch B_o that contains the same number of original new and old class samples. Then we apply data augmentation on selected exemplars in B_o to generate a contrastive training batch B_c and the knowledge distillation term is applied between the teacher output of B_o and the current model output of B_c to maintain the already learned knowledge. Details of each component is described in the remaining section.

Exemplar Selection From Clusters The main challenge of existing exemplar selection methods is that they cannot adapt to the intra-class variation especially for food images due to its high variability. For example, the images in apple category may contain many types such as green apple, red apple, sliced apple, diced apple, whole apple and etc. Therefore,

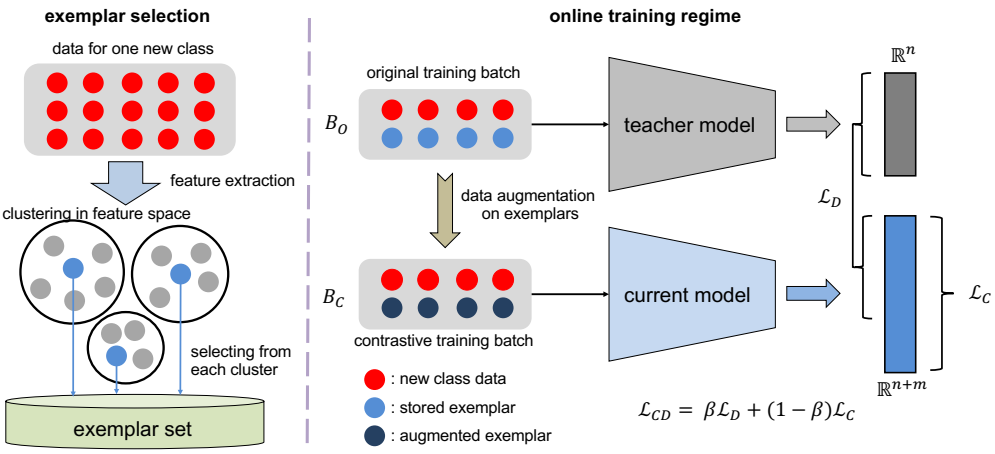


Figure 4.1. Overview of proposed method. The left side shows our exemplar selection algorithm, which selects the most representative data from center of each cluster generated based on visual similarity in feature space. Right part shows our online learning regime where each new class data is paired with one randomly selected exemplar to produce the original balanced training batch B_O. We perform data augmentation on selected exemplars to generate contrastive training batch B_C and the distillation loss $\mathcal{L}_D$ is applied between the output of the teacher model using B_O and the output of the current model using B_C. n and m denote the number of already learned classes and new added classes, respectively. β is a hyper-parameter to combine $\mathcal{L}_D$ with cross-entropy loss $\mathcal{L}_C$. (Best viewed in color)

selecting from class mean as in Herding [51] will not work well when there exists more than one main types within that food class. Our proposed method addresses this problem by first clustering the data for each class based on visual similarity and then select exemplars from each generated cluster. We consider Power Iteration Clustering (PIC) [117] as our clustering approach, which is a graph based method and shown to be effective even in large scale database [127]. But other clustering methods are also feasible such as K-means [12]. One advantage of PIC is the number of generated clusters are not set beforehand, so there is more clusters if one class contains more main types and vice versa.

Given n_c images $\{(\mathbf{x}_1, y), ...(\mathbf{x}_{n_c}, y)\}$ for one new class c, we first generate nearest neighbor graph by connecting to their 10 neighbor data points in the Euclidean space using extracted feature embeddings. Let $f(\mathbf{x}_i)$ denotes the extracted feature for the i-th image, we apply the

sparse graph matrix $G = \mathbb{R}^{n_c \times n_c}$ with zeros on the diagonal and the remaining elements of G are defined by

$$\mathrm{e_{i,j}} = exp^{-} \frac{|(\mathbf{x_i}) - f(\mathbf{x_j})||^2}{\sigma^2}$$

where σ denotes the bandwidth parameter and we empirically use $\sigma = 0.5$ in this work. Then, we initialize a starting vector $s^{n_c \times 1} = [\frac{1}{n_c}, ..., \frac{1}{n_c}]^T$ and iteratively update it using Equation 4.1

$$s = L_1(\alpha(G + G^t)s + (1 - \alpha)s) \tag{4.1}$$

where $\alpha = 0.001$ refers to a regularization parameter and $L_1([.5])$ denotes the L-1 normalization step. The generated clusters are given by the connected components of a directed unweighted subgraph of G denoted as $\tilde{G}$. We set $\tilde{G}_{\mathrm{i,j}} = 1$ if $\mathrm{j} = argmax_{\mathrm{j}} \mathrm{e_{i,j}}(s_{\mathrm{j}} - s_{\mathrm{i}})$ where s_{i} refers to the i-th element of the vector. Note that there is no edge starts from i if $\{\forall \mathrm{j} \neq \mathrm{i}, s_{\mathrm{j}} \leq s_{\mathrm{i}}\}$, *i.e.* s_{i} is a local maximum.

Online Learning Regime Since future food class distribution is usually unpredictable and imbalanced, it becomes more challenging to maintain the learned knowledge due to potential class-imbalanced problem. However, almost all existing online continual learning methods use balanced datasets such as MNIST [116] and CIFAR [2] which contain the same number of training data for each class. In addition, the knowledge distillation term also becomes less effective when the teacher model is not trained on balanced data [67]. Therefore, we propose a more effective online learning regime, which consists of two main parts: using balanced training batch and applying knowledge distillation on augmented exemplars.

Suppose the model is already trained on n classes and the data stream $\{(\mathbf{x}_1^k, y_1^k)...\} \in D^k$ for incremental step k contains m newly added classes where $y^k \in \{n+1, n+2, ..., n+m\}$. We pair each new class data $(\mathbf{x}_{\mathrm{i}}^k, y_{\mathrm{i}}^k)$ with a randomly selected exemplar $(\mathbf{v}_{\mathrm{j}}, y_{\mathrm{j}}) \in E^{k-1}$ where E^{k-1} denotes exemplar set containing stored exemplars for classes $\{1, 2, ..., n\}$ belonging to $\{\mathcal{T}^0, ..., \mathcal{T}^{k-1}\}$. Therefore, each training batch B contains exactly $\frac{b}{2}$ new class data and $\frac{b}{2}$ augmented old class exemplars given batch size $b = |B|$.

To make the distillation term more effective, instead of using the identical training batch for both current model and teacher model as done in existing approaches, we propose to apply data augmentation on selected exemplars in original training batch B_o to generate its

corresponding contrastive training batch B_c where B_c and B_o are used as input to current model and teacher model, respectively.

The output logits of the current model is denoted as $p^{(n+m)}(B_c(\mathbf{x})) = (o^{(1)}, ...o^{(n+m)})$, the teacher's output logits is $\hat{p}^{(n)}(B_o(\mathbf{x})) = (\hat{o}^{(1)}, ..., \hat{o}^{(n)})$ where $B_c(\mathbf{x})$ and $B_o(\mathbf{x})$ denote the data in augmented and original training batch. The knowledge distillation loss [9] is formulated as in Equation 4.2, where $\hat{p}_T^{(\mathrm{i})}$ and $p_T^{(\mathrm{i})}$ are the i-th distilled output logit as defined in Equation 4.3

$$\mathcal{L}_D(B_c(\mathbf{x}), B_o(\mathbf{x})) = \sum_{\mathrm{i}=1}^{n} -\hat{p}_T^{(\mathrm{i})}(B_o(\mathbf{x}))log[p_T^{(\mathrm{i})}(B_c(\mathbf{x}))] \tag{4.2}$$

$$\hat{p}_T^{(\mathrm{i})} = \frac{\exp\left(\hat{o}^{(\mathrm{i})}/T\right)}{\sum_{\mathrm{j}=1}^{n}\exp\left(\hat{o}^{(\mathrm{j})}/T\right)} \ , \ p_T^{(\mathrm{i})} = \frac{\exp\left(o^{(\mathrm{i})}/T\right)}{\sum_{\mathrm{j}=1}^{n}\exp\left(o^{(\mathrm{j})}/T\right)} \tag{4.3}$$

$T > 1$ is the temperature scalar used to soften the distribution, which forces the network to learn more fine grained knowledge. The cross entropy loss to learn new classes can be expressed as in Equation 4.4

$$\mathcal{L}_C(B_c(\mathbf{x})) = \sum_{\mathrm{i}=1}^{n+m} -\hat{y}^{(\mathrm{i})}log[p^{(\mathrm{i})}(B_c(\mathbf{x}))] \tag{4.4}$$

where $\hat{y}$ is the one-hot label for input data x. The overall cross-distillation loss function is formed as in Equation 4.5 by using a hyper-parameter β to tune the influence between two components.

$$\mathcal{L}_{CD}(B_c(\mathbf{x})) = \beta\mathcal{L}_D(B_c(\mathbf{x}), B_o(\mathbf{x})) + (1-\beta)\mathcal{L}_C(B_c(\mathbf{x})) \tag{4.5}$$

In this work, we set $T = 2$ and $\beta = 0.5$. We also notice that using stronger random data augmentation techniques to generative contrastive training batch can achieve better performance to maintain the knowledge for learned classes. Therefore our data augmentation pipeline includes *random flip*, *random color distortions* and *random Gaussian blur*.

4.1.4 Experimental Results

In this section, we first compare our proposed online continual learning method with existing approaches including **ICARL** [46], **ER** [113], [114], **GDUMB** [58] and **ILIO** [21], which all have already been discussed in Section 4.2.2. We also include **Fine-tune** and **Upper-bound** for comparison. **Fine-tune** use only new class data and apply cross-entropy loss for continual learning without considering the previous task performance, *i.e.*, neither exemplar set nor distillation loss is used and it can be regarded as the lower-bound. **Upper-bound** trains a model using all the data seen so far for each incremental learning step using cross-entropy loss in online scenario. Results are discussed in Section 4.1.4.

In the second part of this section, we conduct ablation study to show the effectiveness of each component of proposed method including exemplar selection algorithm and online training regime, which is illustrated in Section 4.1.4.

Datasets

In this work, we use **Food1K** to evaluate our method, which is a recently released challenging food dataset consisting of $1,000$ selected food classes from Food2K [13]. The dataset is originally divided as 60%, 10% and 30% for training, validation and testing, respectively. Note that no class label is given in test set so we use images in validation set as testing data. In addition, we also construct a subset of Food1k using 100 randomly selected food classes denoted as **Food1K-100** for experiment. Specifically, for **Food1K-100**, we randomly arrange 100 classes into the splits of 1, 2, 5, 20 as step size (number of new class added for each step) and for **Food1K** we perform large scale continual learning using 100 new classes for each incremental step.

Implementation Details

Our implementation is based on Pytorch [55]. We use ResNet-18 as our backbone network by following the setting suggested in [53] with input image size 224×224. We use stochastic gradient descent optimizer with fixed learning rate of 0.1 and weight decay of 0.0001. We

store $q = 20$ exemplars per class in exemplar set as suggested in [46] and the batch size is set as 32 (with 16 new class data paired with 16 randomly selected exemplars). For all experiments, each data (except stored exemplars) is used only once to update the model in online scenario.

Evaluation protocol: after each incremental learning step, we evaluate the updated model on test data belonging to all classes seen so far and we use Top-1 accuracy for Food1K-100 and Top-5 accuracy for Food1K. Besides, we also report average accuracy (*Avg*) and last step accuracy (*Last*) for comparison where *Avg* is calculated by averaging the accuracy for all incremental steps to show the overall performance for entire continual learning process and *Last* accuracy shows the final performance on the entire dataset after the last step of continual learning. We repeat each experiment 5 times using different random seeds to arrange class and the average results are reported.

Table 4.1. Average accuracy and Last step accuracy with step size 1, 2, 5, 10, 20 on Food1K-100 and step size 100 on Food-1K. Best results (except upper-bound) are marked in bold.

Datasets	**Food1K-100**										**Food1K**	
Step size	1		2		5		10		20		100	
Accuracy	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
Fine-tune	0.043	0.009	0.081	0.029	0.182	0.018	0.379	0.134	0.497	0.233	0.265	0.099
Upper-bound	0.805	0.759	0789	0.752	0.807	0.743	0.827	0.749	0.813	0.744	0.788	0.805
ICARL [46]	0.619	0.539	0.694	0.615	0.581	0.502	0.729	0.603	0.769	0.660	0.573	0.474
ER [113], [114]	0.645	0.586	0.612	0.582	0.528	0.520	0.694	0.599	0.728	0.633	0.533	0.428
GDUMB [58]	0.606	0.430	0.612	0.441	0.573	0.507	0.591	0.456	0.754	0.623	0.506	0.289
ILIO [21]	**0.695**	**0.670**	0.681	**0.643**	0.501	0.452	0.703	0.633	0.708	0.596	0.515	0.428
Ours	0.692	0.661	**0.702**	0.641	**0.643**	**0.563**	**0.762**	**0.669**	**0.786**	**0.699**	**0.612**	**0.504**

Comparison With Existing Methods

Table 4.1 summarizes the average accuracy (*Avg*) and last step accuracy (*Last*) for all incremental step sizes. Overall, we notice that the online continual learning performance vary a lot for different step sizes. Given fixed total number of classes to learn, smaller step size will produce more incremental steps so catastrophic forgetting appears more frequently. On the other hand, for larger step size, although there will be less incremental steps, learning more classes for each step is also a challenging task especially in online scenario to use each

data only once for training. Specifically, we observe severe catastrophic forgetting problem by using *Fine-tune* where both *Avg* and *Last* accuracy are much lower compared with *Upper-bound* due to the lack of training data for learned tasks during the continual learning process. All existing methods achieve significant improvement compared with *Fine-tune* especially for *ILIO* [21], which works more effectively when step size is very small as their final prediction is given by the combination of outputs for both the teacher model and current model. Note that *ILIO* requires the teacher model for both training and inference phases which greatly increases the memory storage while other methods included ours only use teacher model during the training phase. However, as incremental step size increase, our method achieves best performance even for very large scale continual learning for $1,000$ classes in Food1K. We also show the accuracy evaluated after each incremental learning step with step size 5, 10, 20 and 100 in Figure 4.2. Our method outperforms state-of-the-art for all learning steps with smallest performance gap compared with *upper-bound*. Note that we did not provide the figures for step size 1 and 2 as they contain too many learning steps (100 and 50 respectively), which is difficult for visualization.

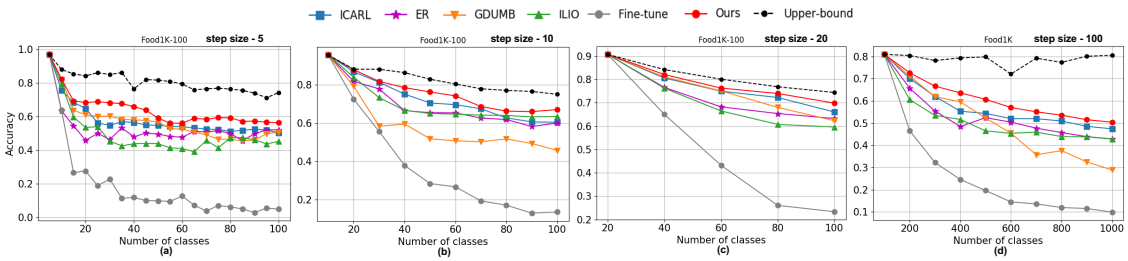


Figure 4.2. Accuracy for each incremental step with step size (a) 5 (b) 10 (c) 20 on Food1K-100 and (d) step size 100 on Food-1K. (Best viewed in color)

Ablation Study

In this part, we conduct ablation studies to analyze the effectiveness of (1) **component-1:** our proposed exemplar selection algorithm that selects representative data from clusters generated based on visual similarity and (2) **component-2:** our online training regime using

balanced training data for new and old class, and contrastive training batch for knowledge distillation. Specifically, we consider the following methods for comparisons:

- **baseline**: removing both component-1 and component-2 from our method, *i.e.*, use herding [51] for exemplar selection instead and pair new class data in training batch with the random number of exemplars
- **baseline + our exp**: baseline + component-1
- **baseline + our training regime**: baseline + component-2
- **Ours**: baseline + component-1 + component-2

Figure 4.3 shows the results for each incremental step with step size 5, 10, 20 and 100. Compared with *baseline*, we observe performance improvement by incorporating each component of proposed method. The best performance is obtained when combining both components. In addition, we notice that our training regime using balanced training batch performs more effectively than our exemplar selection since severe class-imbalanced problem exists in this Food1K dataset, where the number of training data ranges from $[91, 1199]$ per food class.

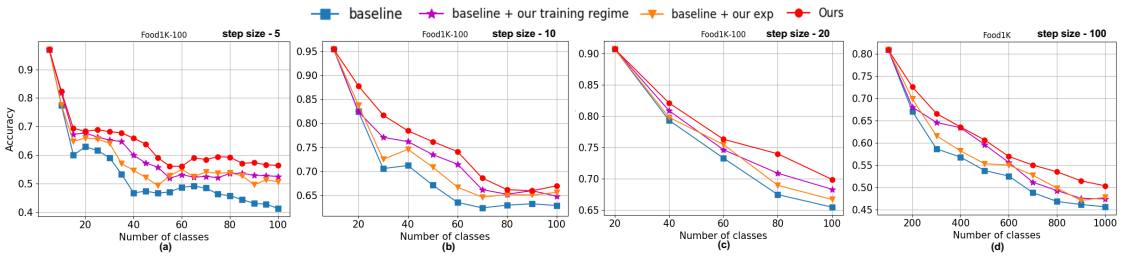


Figure 4.3. Ablation study with step size (a) 5 (b) 10 (c) 20 on Food1K-100 and (d) step size 100 on Food-1K. (Best viewed in color)

Influence of Exemplar Size

For experiments in Section 4.1.4, we follow the protocol [46] to use 20 exemplars per class. In this part, we vary the number of exemplar stored for each class $q \in \{10, 50, 100\}$ and compare *baseline + our exp* using our proposed exemplar selection algorithm with *baseline*

using Herding selection [51]. We use Food1K-100 with step size 5 and the average accuracy are shown in Table 4.2. In general, the performance becomes better for both methods when more exemplars are used. However, the memory storage capacity is one of the most important factors for continual learning especially in online scenario and we observe that our proposed approach is more efficient which outperforms *baseline* for a larger margin when using less exemplars.

Table 4.2. Average accuracy on Food1K-100 with step size 5 by varying exemplar size. Best results marked in bold.

Method	$q = 10$	$q = 50$	$q = 100$
baseline	0.486	0.629	0.697
baseline + our exp	0.527	0.651	0.706

Visualization of Selected Exemplars

A t-SNE [128] visualization comparing herding [51] and our proposed exemplar selection method is shown in Figure 4.4 where we randomly select three food classes from Food1K as denoted by blue, green and orange dots, respectively and red dots refer to the selected exemplars. As shown in the left half of the figure, most exemplars selected by herding are concentrated in a small area for each class as indicated by the black box. Therefore, the model gradually forgets the knowledge outside the black box during the continual learning process, leading to catastrophic forgetting. Our method addressed this problem by performing clustering at first based on visual similarity and then select exemplars from all generated clusters to better represent the intra-class diversity for each food class as illustrated in Section 4.1.3. In the right half of this figure, we find that the exemplars selected by our method covers a wider region for each food class, which helps to produce higher quality classifiers to retain the learned knowledge due to better generalization ability of our selected exemplars as shown in Figure 4.3 by comparing **baseline** with **baseline + our exp**.

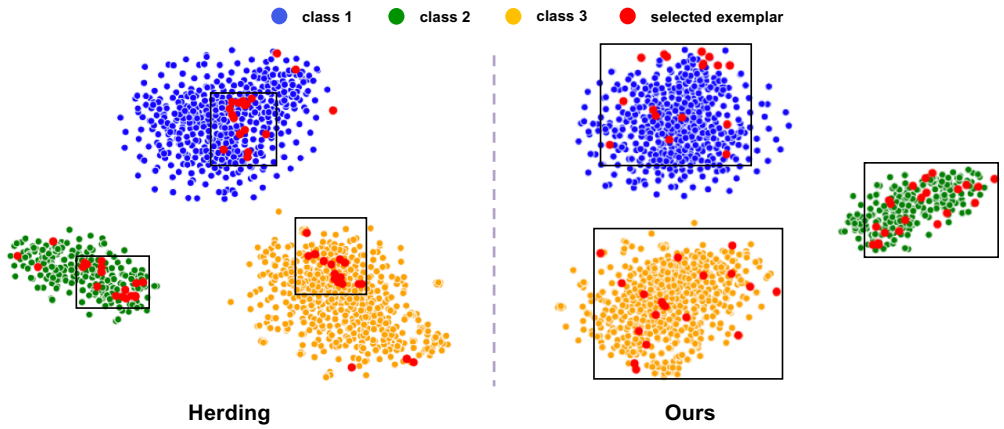


Figure 4.4. A t-SNE [128] visualization by comparing herding [51] with our proposed exemplar selection algorithm. We randomly select three classes from Food1K corresponds to three different colors and the red dots represent the selected exemplars. The black box indicates the area where most exemplars are located for each class. (Best viewed in color)

Visualization of Contrastive Training Batch

Figure 4.5 shows the exemplars for learned food classes in original and contrastive training batch using our proposed data augmentation pipeline including *random flip*, *random color distortions* and *random Gaussian blur*. By comparing results of **baseline** with **baseline + our training regime** as shown in Figure 4.3, we observe that using augmented data is more effective to help retain the already learned knowledge to achieve better performance. One explanation is that each exemplar stored in the exemplar set can be selected for more than once to pair with new class data during the online training phase, so the data augmentation step helps to improve the classifier's generalization ability to obtain higher accuracy on learned classes. In addition, the knowledge distillation term also becomes more efficient to maintain the performance for old classes by using balanced training batch for old and new class data and transferring the learned knowledge from teacher model using original training batch to the current model using contrastive training batch as formulated in Equation 4.2.

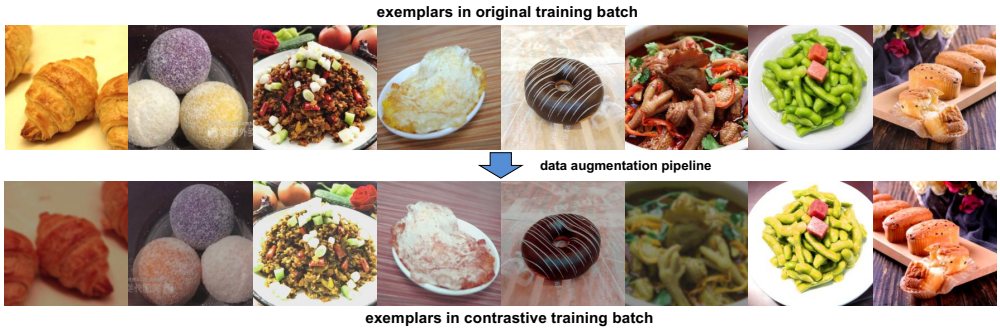


Figure 4.5. Visualization of contrastive training batch generated by our proposed data augmentation pipeline including *random flip*, *random color distortions* and *random Gaussian blur*. (Best viewed in color)

4.2 Exemplar-Free Online Continual Learning

4.2.1 Overview

Though modern deep learning based approaches have achieved significant progress to address computer vision problems such as image recognition, it is still challenging to learn new tasks incrementally from data stream due to the unavailability of learned task data. The major obstacle is called catastrophic forgetting [3] where the performance on old tasks drop dramatically during the learning phase of new task. To overcome this issue, online continual learning [21], [56], [68] has emerged, which defines the learning protocol that both new tasks and their data come sequentially overtime and each data is used only once for training. During inference, the model should perform well on all tasks learned so far without knowing the task index. While existing methods [57], [58] have made remarkable progress by storing part of learned task data as exemplars during continual learning, there are several drawbacks associated with exemplar-based approaches: (i) it requires extra storage consumption, which is a significant constraint for online continual learning; (ii) it poses a new challenging problem of how to select the most representative data as exemplars, (iii) for certain applications such as health or medical research, the data may not be allowed to be kept for a long time due to privacy concern. In this work, we propose a novel exemplar-free online continual learning method for image classification task, which addresses the aforementioned limitations of current approaches.

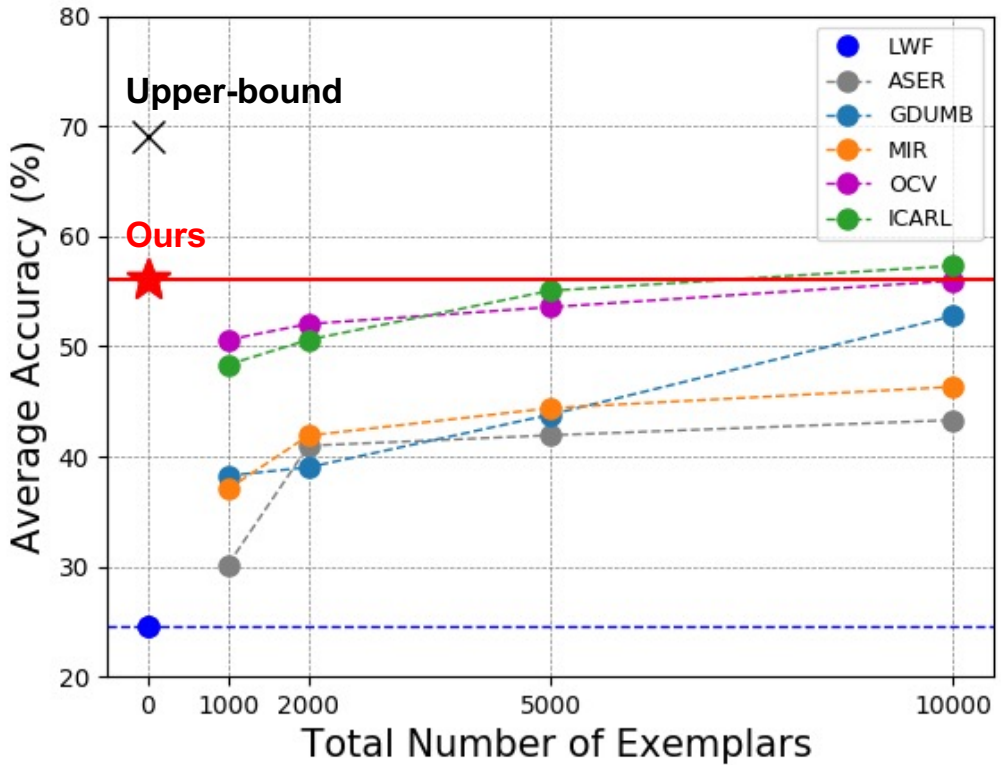


Figure 4.6. CIFAR-100 Top-1 average accuracy after learning all tasks with incremental step size 5. Dash lines show the results of existing work, which require stored exemplars (except LWF) and the red solid line shows result of our method. Upper-bound is obtained by training a model using all training samples from all classes.

One of the main reasons for catastrophic forgetting is the biased predictions caused by biased parameters in the classifier towards new classes due to the lack of old data [10]. The most recent work [108] addresses this problem by selecting candidates at first and then performing classification using distance-based classifier [61] based on stored exemplars. Inspired by this, we instead leverage nearest-class-mean(NCM) classifier, which uses class mean vector for classification and does not require any exemplar. In addition, compared with the NCM used in ICARL [46] where the class mean is estimated using stored exemplars, our mean vector for each seen class is calculated on all data seen so far during training phase through online mean update criteria, which is more representative especially when the allowed exemplar size is limited. Furthermore, our NCM is performed only on selected

candidates [108] so the class mean are better separated than using all classes as in ICARL [46], thus achieving higher accuracy for classification.

As shown in Fig 4.6, without using any exemplar, our method applied on CIFAR-100 [2] not only outperforms existing methods [41], [46], [57], [58], [108], [129] with large margins under standard experimental protocol as proposed in [46] (2,000 exemplars in total), but also achieves competitive performance given increased exemplar size.

4.2.2 Related Work

With the objective of mitigating catastrophic forgetting, continual learning has been studied in both offline [10], [41], [46], [47], [60], [64], [103] and online scenarios [21], [56], [58], [68], [107], [108], [129]. In the online scenarios, each data is observed only once by the model, which is more related to real life applications. In this section, we summarize existing methods that are closely related to our work.

Regularization-based methods retain learned knowledge by restricting the change of corresponding weights. Knowledge distillation loss [9] is widely used in [10], [41], [46] and a variant distillation loss is introduced in ILIO [21] to achieve improved performance in online scenario. Besides, A-GEM [56] is an efficient version of GEM [68] where both methods use stored exemplars to ensure that the loss for learned tasks does not increase during each learning step. Most recently, OFR [107] proposed a novel clustering based exemplar selection approach and showed its effectiveness on food image classification task.

Reply-based approach aims to address catastrophic forgetting by storing part of old task data to perform knowledge replay during continual learning. ICARL [46] proposed to apply herding algorithm [51] to select and store exemplars based on class mean. Random retrieval is applied in Experience-Replay(ER) [113], [114] to ensure that each new data has the same probability to be stored as exemplar in memory buffer. MIR [129] proposed a controlled sampling of memories. A greedy balancing sampler was introduced in GDUMB [58] which randomly selected as much data as the memory allowed and the classifier was trained on stored exemplars only. A exemplar scoring method was proposed in ASER [57] to preserve latent decision boundary. Instead of using original data as exemplar, OCV [108] only selected

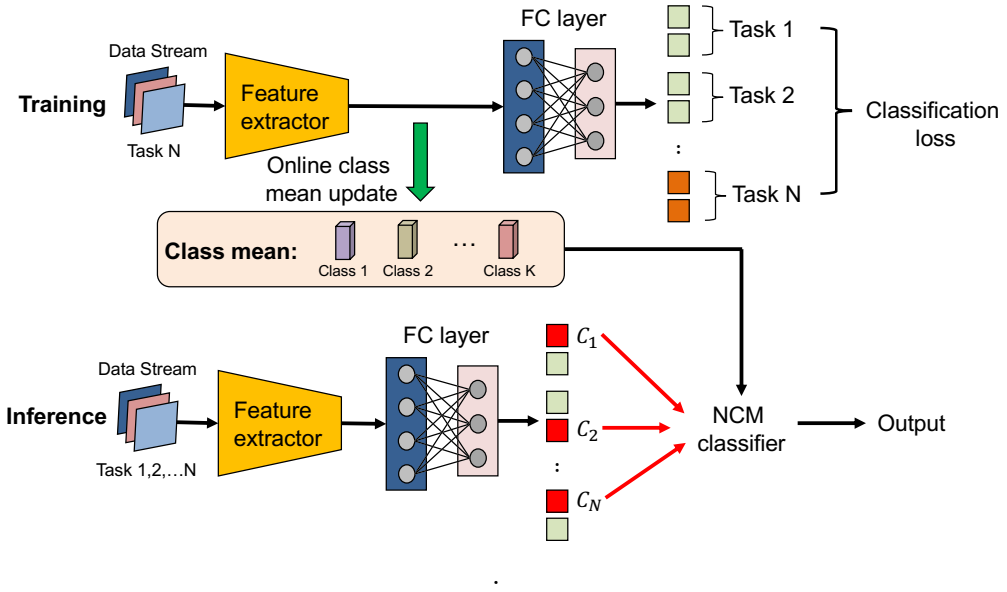


Figure 4.7. The overview of our method. The upper half shows the online training phase where we estimate the class mean dynamically using online mean update criteria on all data seen so far. The FC output for learned tasks are fixed to maintain the discrimination. For the inference phase, we first select candidates from each learned task, denoted as $C_1, ...C_N$, and then apply NCM classifier for classification based on estimated class mean $Class_{C_1}, ...Class_{C_N}$

and stored extracted feature embeddings. The use of exemplars may not always be feasible in real life. In contrast, our proposed method does not require to store any exemplar while achieving competitive performance compared to exemplar-based methods even for very large exemplar sizes.

4.2.3 Proposed Method

The overview of our proposed method is shown in Fig 4.7. A fixed feature extractor pretrained on large scale image datasets, *e.g.*, ImageNet [1] is applied as backbone network in both training and inference phases, which provides more discriminative embeddings than original images as input for online continual learning [108].

Training Phase

The Nearest-Mean-of-Exemplar(NME) classifier proposed in ICARL [46] achieves remarkable progress. However, the performance greatly relies on the exemplar size as the class mean vectors used for classification is only estimated through stored exemplars, which struggles when allowed storage is limited or the selected exemplars are not representative enough. As shown in the upper half of Fig 4.7, our method addresses this issue by estimating the class mean on all data seen so far using online mean update criteria. Specifically, during the learning phase of $\mathcal{T}^N$, for each new data $(\mathbf{x}_{\mathrm{i}}^N, y_{\mathrm{i}}^N)$, we calculate the class mean vector using Eq (4.6).

$$\mathbf{v}_{y_{\mathrm{i}}^N} = \frac{n}{n+1}\mathbf{v}_{y_{\mathrm{i}}^N} + \frac{1}{n+1}\mathcal{F}(\mathbf{x}_{\mathrm{i}}^N) \tag{4.6}$$

where $\mathbf{v}$ denotes the class mean, which is initialized as zero for each new class. $\mathcal{F}$ refers to the feature extractor and n is the number of data seen so far for class y_{i}. Instead of using knowledge distillation loss [9] for regularization, we apply cross-entropy as classification loss to maximally maintain the discrimination for each learned task [108], which provides the basis for selecting candidates in the inference phase.

Inference Phase

As indicated in [10], catastrophic forgetting is largely due to the prediction bias towards new classes. A recent work [108] addressed this issue by selecting candidates at first and then applying a distance-based classifier that does not have biased parameters. However, it still stores exemplars and require synthesized data to tune hyper-parameters in final prediction equation. Our method addresses this problem by leveraging nearest-class-mean (NCM) classifier on selected candidates for classification. Specifically, after the learning phase of $\mathcal{T}^N$, we denote $\{o_1^1, o_2^1, ...o_M^1, ..., o_1^N, o_2^N, ...o_M^N\}$ as the output of the FC layer where $o_{\mathrm{j}}^{\mathrm{i}}$ refers to the output logit for the class j in task i and M is the incremental step size. The total number of classes seen so far is $K = M \times N$. For the output of each task i $\in \{1, 2, ...N\}$, we

select candidate by $C_{\mathrm{i}} = argmax\{o_1^{\mathrm{i}}, ...o_M^{\mathrm{i}}\}$. Finally, the NCM classifier make prediction for test data $\mathbf{x}_t$ by using Eq (4.7).

$$\begin{aligned} y_{\mathbf{x}_t} &= \underset{C_1, C_2, ..., C_N}{argmin} \{d^{C_1}, ...d^{C_N}\} \\ where \quad d^{\mathrm{i}} &= ||\mathbf{v}_{\mathrm{i}} - \mathcal{F}(\mathbf{x}_t)||_2 \end{aligned} \tag{4.7}$$

where $d^{\mathrm{i}}, \mathrm{i} \in \{C_1, ...C_N\}$ denotes the Euclidean distance between class mean $\mathbf{v}_{\mathrm{i}}$ and extracted embedding of test data $\mathcal{F}(\mathbf{x}_t)$. Compared with existing approaches [46], [108], our method neither requires storing exemplars nor needs to tune any hyper-parameter.

4.2.4 Experimental Results

We validate our method on two benchmark datasets including CIFAR-100 [2] and Food-1k [13]. For CIFAR-100, we follow the protocol in [79] to construct **Split CIFAR-100** by dividing the 100 categories into 20 tasks, each contains 5 classes. For Food-1k, same as [107], we first randomly select 100 food categories to construct **Food1k-100** and then divide the subset into 5, 10, and 20 splits to conduct experiments.

4.2.5 Implementation Details

We follow the benchmark experimental protocol in [46], [68] to use ResNet-18 [53] as backbone network. We use SGD optimizer with fixed learning rate of 0.1. Batch size is 16 and each training data is used only once (1 epoch). The splits of both datasets uses identical random seed as in [46], [107] and the backbone network for all compared methods is pre-trained on ImageNet [1] to ensure fair comparison in all experiments. we use top-1 accuracy as the evaluation metric.

4.2.6 Results on Split CIFAR-100

We compare our method with **ASER** [57], **GDUMB** [58], **GSS** [71], **MIR** [129], **A-GEM** [56] and OCV [108]. We vary the exemplar size Q for existing methods with $Q \in \{1,000, 2,000, 5,000, 10,000\}$ and the result is shown in Figure 4.8. We observe sig-

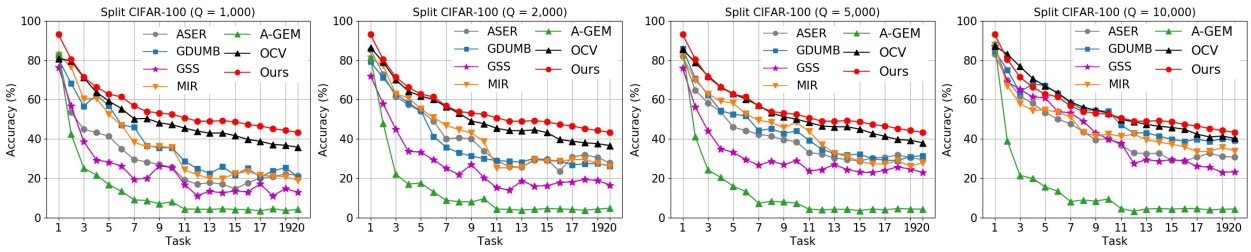


Figure 4.8. Results on Split CIFAR-100 by varying the allowed exemplar size $Q \in \{1,000, 2,000, 5,000, 10,000\}$. There are 20 tasks in total with each task contains 5 non-overlapped classes.

nificant improvements for the exemplar size $Q \in \{1,000, 2,000\}$ while still achieving competitive performance with larger Q. Note that $Q = 2,000$ is a standard protocol [46] for exemplar-based approaches. Although storing more exemplars will result in performance improvements for exemplar-based methods, it requires extra storage memory, which is a significant constraint for online continual learning and may not always be feasible in real life applications. The average accuracy *Avg* and last step accuracy *Last* are summarized in Table 4.3 where *Avg* is calculated by averaging all accuracy obtained after each learning step, which shows the overall performance for the entire online continual learning problem and the *Last* accuracy shows the performance after the continual learning for all classes seen so far. Our method achieves the best results in terms of *Avg* and *Last* while does not require storing exemplars compared with existing work. Besides, we also include **Ours(w/o)** for comparison, which performs NCM on all classes instead of on candidates as in **Ours**, the result shows the effectiveness of selecting candidates when applying NCM for classification under online continual learning scenario.

4.2.7 Results on Food1k-100

In this section, we evaluate our method using the challenging food images. Therefore, besides comparing with existing methods: **ICARL** [46], **ER** [113], [114], **GDUMB** [58], **ILIO** [21] and **OFR** [107], we follow the experimental setting in [107] to further include **Fine-tune** (using only new class data to update model without considering the learned task performance) as baseline and **Upper-bound** (training the model using all training samples

Table 4.3. Average accuracy and Last step accuracy on Split CIFAR-100. Best results marked in bold.

Datasets	Split CIFAR-100							
Size of exemplar set	$Q = 1,000$		$Q = 2,000$		$Q = 5,000$		$Q = 10,000$	
Accuracy(%)	Avg	Last	Avg	Last	Avg	Last	Avg	Last
A-GEM [56]	13.9	4.3	14.1	4.83	13.8	4.4	13.9	4.5
MIR [59]	37.4	19.1	41.9	26.1	44.4	27.9	46.3	34.0
GSS [71]	24.2	12.9	26.9	16.7	31.5	23.1	43.2	23.3
ASER [57]	29.9	21.6	40.9	27.8	41.2	29.5	43.3	30.8
GDUMB [58]	38.3	20.7	39.2	26.4	43.8	31.6	52.8	39.2
OCV [108]	50.6	35.8	52.0	36.7	53.6	38.1	56.0	40.3
Ours (w/o)		Avg:	54.7		Last:	41.6		
Ours		Avg:	**56.3**		Last:	**43.4**		

from all seen classes at each step) for experiment. The exemplar size is fixed with $Q = 2,000$ as in protocol [46], [107] and we vary the step size for 5, 10 and 20 corresponding to 20, 10 and 5 incremental steps, respectively. The result is shown in Fig 4.9. We observe severe performance degradation by comparing **Fine-tune** with **Upper-bound** due to the lack of training data for learned tasks, which shows the necessity to address catastrophic forgetting problem for online incremental learning. In addition, by comparing **Fine-tune** and **Upper-bound** results for different step sizes, we notice that the problem becomes more challenging when the step size is smaller due to the increase of incremental steps. Our proposed method not only achieves the best results with smallest performance gap between **Upper-bound** for all step sizes, but also outperforms the existing work with a larger margin in the challenging case of small step size.

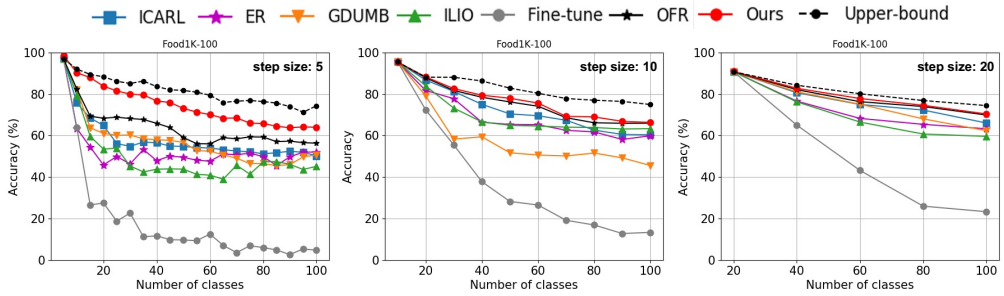


Figure 4.9. Results on Food1k-100 by varying the incremental step size $M \in \{5, 10, 20\}$ (Best viewed in color)

5. IMAGE-BASED DIETARY ASSESSMENT

5.1 Multi-Task Classification and Portion Estimation For Single-Item Food Images

5.1.1 Overview

Multi-task learning aims to solve more than one tasks simultaneously, which is typically done with either hard or soft parameter sharing of hidden layers. However, hard parameter sharing is not a feasible solution for our application since it is difficult for the two tasks to share one common feature space. In this work, we introduce soft parameter sharing where each task has its own feature space and the lower layer of the two models are regularized. Our goal is to investigate the connection between the two tasks and our experimental results show that the performance of both food classification and food portion size estimation can be improved by regularizing the lower layers using L2 norm. In addition, due to the difficulty of directly mapping an RGB image to a numeric portion size, we apply cross-domain feature adaptation that concatenates the feature vectors extracted from the classification network with the feature vectors extracted from the regression network. The feature vectors from the classification task can provide prior knowledge to better inform the portion size estimation given the food category is known. To adapt the features extracted from different domains for joint regression, we extensively studied the use of normalization techniques [130], [131].

5.1.2 Related Work

Image-Based Dietary Assessment

Food is an important component of daily life. The type of foods and amount consumed can directly impact people's health. The recent success of modern deep learning techniques [53], [132] have greatly improved the performance of image-based diet assessment in recent years.

Food Classification. The most common food image recognition method is to apply state-of-the-art models [53], [132] to train a deep network that can recognize a variety of food items. For example, authors in [16] use UEC-100 [133] and UEC-256 [134] food im-

age datasets for testing, and ImageNet-ILSVRC [1] for training. Their methods contain a combination of baseline feature extraction and neural network fine-tuning. An ensemble of deep networks are proposed in [18] to improve the classification performance. A novel deep learning-based food image recognition algorithms is proposed in [17], which is inspired by [135], [136].

Food Portion Estimation. Automatic estimation of food portion size from an input food image is an open problem and there are many different methods to address it. In [22], food portion is divided into discrete serving sizes and food portion estimation is treated as a classification problem to determine the fixed serving size. [23] uses pre-defined 3D food models that are projected onto the scene to find the best fit with camera calibration. In [24], food volume is estimated from the predicted depth map of the eating scene. The depth map is then converted to voxel representation which is used to estimated food volumes. An end-to-end approach for food energy estimation is proposed in [26], where the concept of energy distribution map [27] replaces the 'depth map' in [24] and the final food energy estimation is reported.

Multi-task Learning

Multi-task learning [137] (MTL) has been applied to many computer vision problems that intended to impose knowledge sharing while solving multiple related tasks simultaneously. In the context of deep learning, MTL is typically done with either hard or soft parameter sharing of hidden layers.

Hard parameter sharing is the most common method used in MTL where all tasks share the feature extraction layers while keeping task-specific output layers. In [15], the authors used MTL to improve the classification performance by clustering visually similar foods together. In [138], the authors applied MTL for food attribute prediction including food classes, ingredients, cooking instruction and food energy. However, sharing the feature map for cross domain tasks greatly impact the performance. In addition, the dataset used in [138] for food energy is obtained by web crawler from a cooking website and cannot be verified for its accuracy.

Soft parameter sharing is another approach in MTL where each task has its own model with its own parameters and the distance between the parameters of lower layers is then regularized in order to force the parameters to be similar. [139] proposed to use L2 distance for regularization and then [140] used the trace norm.

5.1.3 Proposed Method

In this work, we propose an end-to-end framework for food classification and portion size estimation. The overall network structure is shown in figure 5.1.

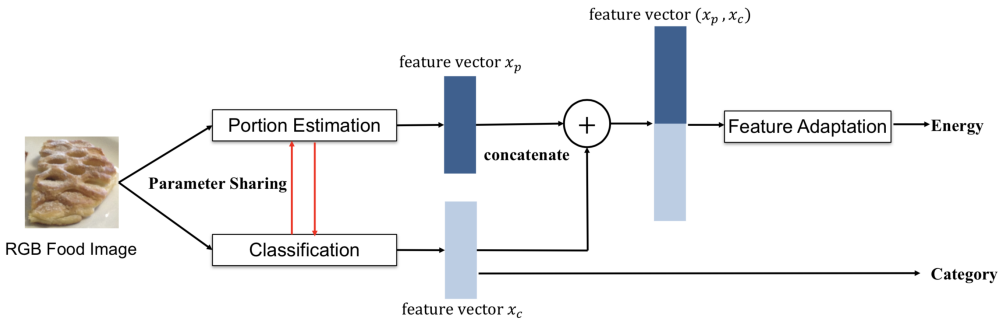


Figure 5.1. The architecture of our proposed model for image-based food classification and portion size estimation. L2-norm base soft parameter sharing is used to jointly train two tasks simultaneously. The feature vectors from each task are then concatenated together and we apply cross-domain feature adaptation to further improve the performance of food portion size estimation.

Multitask: Soft Parameter Sharing

Multi-task Learning (MTL) is the most common method to simultaneously solve multiple tasks. Since we are performing two different tasks, i.e., classification and regression, using hard parameter sharing, where both tasks share the same feature map, is not suitable. Instead, we apply soft parameter sharing where each task has its own model with its own parameters. The distance between the parameters of lower layers of the two models is then regularized in order to force the parameters of the two models to be similar. The idea is that although the two tasks are different, they can be regarded as dependent tasks, i.e., the classification task can provide useful knowledge for optimizing regression task and at the

same time the regression task can provide useful knowledge for learning classification task. This is based on the fact that it will be easier to know the food category if we know the food portion value at first and also it will be easier to get the food portion size if we know the food category as *a priori*.

Given the input data $(\mathbf{x}, y, z)$ where $\mathbf{x}$ is the input image, y and z denoted the groundtruth for food category and portion size, respectively. We use cross-entropy loss $\mathcal{L}_c$ for classification and apply L1-norm loss $\mathcal{L}_r$ for portion size estimation. The two loss functions can be written as

$$\begin{aligned} \mathcal{L}_c &= \sum_{\mathrm{i}=1}^{n} -\hat{y}^{(\mathrm{i})} log[f_c^{(\mathrm{i})}(x)] \\ \mathcal{L}_r &= |z - f_r(x)| \end{aligned} \tag{5.1}$$

where $\hat{y}$ is the one hot label for food category and n is the dimension of the feature vector. f_c and f_r denote the models for classification and regression, respectively. Note that for regression task, the vector dimension is 1.

For parameter sharing, we use L2-norm to regularize the parameters of the two models. Let p_c and p_r denoted as the parameters of lower layers of classification model and regression model respectively, the loss function can be expressed as

$$\mathcal{L}_{ps} = \sum_{\mathrm{i}=1}^{m} (p_c^{(\mathrm{i})} - p_r^{(\mathrm{i})})^2 \tag{5.2}$$

where m is the size of parameters of two model. Note that since we apply the same network structure to the two tasks, we have the same number of parameters.

Then, the overall loss function can be written as

$$\mathcal{L}_{overall} = \mathcal{L}_c + \mathcal{L}_r + \mathcal{L}_{ps} \tag{5.3}$$

Cross Domain Feature Adaptation

Different from classification task, it is difficult to map a RGB image to a numeric portion size value, e.g. if the input image is of size $224 \times 224 \times 3$, then direct approach would map $\mathcal{R}^{224\times224\times3} \rightarrow \mathcal{R}^{1\times1\times1}$ and it is difficult to learn such a mapping. Therefore, we concate-

nate the feature vector extracted using classification network as part of the feature vector extracted by the regression network. The feature vector for classification task can provide prior knowledge to assist the portion size estimation since it will be easier to estimate the food portion size if we already know the food category. We denote the features extracted from classification network as $\mathbf{x}_c$ (of dimension $R^{512\times1}$) and the features extracted from the original portion estimation network as $\mathbf{x}_p$ (of dimension $R^{512\times1}$). However, simply concatenating the features $(\mathbf{x}_p, \mathbf{x}_c)$ (of dimension $R^{1024\times1}$) and applying fully-connected layers have fundamental issues. Features from the two domains have significant differences reflected by the mean and variance of the feature vectors. To adapt the features extracted from different domains and to remove imbalance in feature space for joint regression, we extensively studied the use of normalization techniques.

In this work, we apply Batch Normalization (BN) [131] and Layer Normalization (LN) [130]. LN is defined as:

$$y_{\mathrm{i}} = \gamma\hat{x}_{\mathrm{i}} + \beta, \text{ where } \hat{x}_{\mathrm{i}} = \frac{x_{\mathrm{i}} - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}} \tag{5.4}$$

where γ and β are learnable parameters, $\hat{x}_{\mathrm{i}}$ is the normalized source domain sample for x_{i} and y_{i} is the mapped sample based on learned normalization. σ_L and μ_L are defined as:

$$\mu_L = \frac{1}{H}\sum_{\mathrm{i}=1}^{H} x_{\mathrm{i}}, \quad \sigma_L^2 = \frac{1}{H}\sum_{\mathrm{i}=1}^{H}(x_{\mathrm{i}} - \mu_L)^2 \tag{5.5}$$

where H denotes the number of hidden units in a layer.

BN is defined as:

$$y_{\mathrm{i}} = \gamma\hat{x}_{\mathrm{i}} + \beta, \text{ where } \hat{x}_{\mathrm{i}} = \frac{x_{\mathrm{i}} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{5.6}$$

Similarly γ and β are learnable parameters, $\hat{x}_{\mathrm{i}}$ is the normalized source domain sample for x_{i} and y_{i} is the mapped sample. Let $B = \{x_1, ..., x_m\}$ denote the mini-batch of input samples, σ_B and μ_B are defined as:

$$\mu_B = \frac{1}{m}\sum_{\mathrm{i}=1}^{m} x_{\mathrm{i}}, \quad \sigma_B^2 = \frac{1}{m}\sum_{\mathrm{i}=1}^{m}(x_{\mathrm{i}} - \mu_B)^2 \tag{5.7}$$

5.1.4 Experiment

In this part, we evaluate the performance of our proposed method. For portion estimation, we use Mean Absolute Error (MAE), defined as

$$\text{MAE} = \frac{1}{N}\sum_{\text{i}=1}^{N}|\tilde{w}_{\text{i}} - \bar{w}_{\text{i}}| \tag{5.8}$$

where $\tilde{w}_{\text{i}}$ is the estimated portion value of the i-th image, $\bar{w}_{\text{i}}$ is the groundtruth portion size of the i-th image and N is the number of testing images. We use accuracy to evaluate classification performance. However, since have a multi-task for both classification and regression, we need a better metric that can balance the performance of MAE and classification accuracy. We propose a new metric called MAE to Correctly Classified Ratio (MCCR):

$$\text{MCCR} = \frac{C\sum_{\text{i}\in I}|\tilde{w}_{\text{i}} - \bar{w}_{\text{i}}|}{||I||^2} \tag{5.9}$$

where I denote the correctly classified image. C is a constant, in this experiment, we use $C = 1$. Note that we only calculate the mean absolute portion size error for correctly classified food in this new metric since if the classification result is wrong then it is meaningless to give an estimated portion size. The multi-task network has better performance when the metric has a smaller value.

Dataset

The performance of modern deep learning based methods greatly rely on the availability of good datasets, particularly datasets with correct annotation for computer vision problems. In this work, we aim to build a deep learning framework that can achieve the food classification and portion size estimation simultaneously. However, currently there is no available food related public dataset that contains both the groundtruth food categories and corresponding portion sizes. Therefore, we introduce an eating occasion image to food energy dataset that is collected from a nutrition study. The groundtruth portion size is provided by registered dietitians. The dataset is collected as part of an image-assisted 24-hour dietary recall (24HR)

study [141] conducted by registered dietitians. The study participants are healthy volunteers aged between 18 and 70 years old. A mobile app is used to capture images of the eating scenes for 3 meals (breakfast, lunch and dinner) over a 24-hour period. Foods are provided in buffet style in which pre-weighted foods and beverages in certain categories are served to the participants and they are asked to capture the eating scene images before they start to eat for each meal. The food energy is calculated and used as groundtruth. The dataset contains 96 eating occasion images and we manually crop each food item from each eating occasion. A total of 834 single food images belong to 21 categories are included in this dataset which contains both the category and portion size groundtruth.

Implementation Detail

Our implementation is based on Pytorch [55]. We use standard 18-layer ResNet and the ResNet implementation follows the setting suggested in [53]. We train the network for 100 epochs using Adam optimizer. The learning rate is set to 0.1 and reduces to 1/10 of the previous learning rate after 30, 60, 90 and epochs. The weight decay is set to 0.0001 and the batch size is 32.

Results

Results are shown in Table 5.1. Compared to the two baseline methods that separately train two networks for portion estimation and classification, our method improves both the classification accuracy and the mean absolute error for estimated portion size. In addition, we show that directly using the concatenating features $(\mathbf{x}_p, \mathbf{x}_c)$ causes the performance degradation in MAE since the features from two domain have significant differences reflected by the mean and variance of the two feature vectors from two tasks. We also compared the results using three normalization methods, BN, LN and LN+BN. As shown in Table 5.1, by using LN+BN, we are able to achieve the best classification accuracy and MAE. For correctly classified food, the MAE is only 50.86 Kcal.

Table 5.1. Experimental results for food classification and portion size estimation on food image dataset. The first two rows indicate the results by independently training two tasks. HPS and SPS denoted hard/soft parameter sharing multitask network respectively. CDFA corresponds to using cross domain feature adaptation. LN and BN refer to layer normalization and batch normalization (Best results marked in bold).

Method	Accuracy (%)	MAE (kcal)	MAE-Correct (kcal)	MCCR
Classification	86.08	-	-	-
Portion Estimation	-	62.27	-	-
HPS	50.23	62.53	-	-
SPS	84.96	63.51	-	-
SPS+CDFA	85.14	66.64	61.10	0.7091
SPS+CDFA+BN	86.32	57.94	57.45	0.6577
SPS+CDFA+LN	80.42	62.94	54.83	0.6736
SPS+CDFA+LN+BN	**88.67**	**56.82**	**50.86**	**0.5667**

5.2 End-to-End Food Analysis System For Multi-Food Images

5.2.1 Overview

Modern deep learning techniques have achieved great success in image-based dietary assessment for food localization and classification [14]–[21], as well as food portion size estimation [20], [22]–[27]. However, none of these methods can achieve food localization, classification and portion size estimation in an end-to-end fashion, which makes it challenging to integrate into a complete system for fast and streamlined process.

Image based food localization and classification problems can be viewed as specialized tasks in computer vision. The goal of food localization is to locate each individual food region for a given image with a bounding box. Pixels within the bounding box are assumed to represent a single food, which is the input to the food classification task. Food localization serves as a pre-processing step since it is common for food images in real life to contain multiple food items. However, accurate estimation of an object's portion size is a challenging task, particularly from a single-view food image as most 3D information has been lost when the eating scene is projected from 3D world coordinates onto 2D image coordinates. An object's portion size is defined as the numeric value that is directly related to the spatial quantity of the object in world coordinates. The goal of food portion size estimation is to

derive the food energy from an input image since energy intake is an important indicator for diet assessment. There are existing methods [26], [27] that can estimate food portion size for the entire input image by generating a food energy distribution map, however, they cannot estimate the portion size of each food item separately. This is important as an individual food item can vary greatly in the energy contribution leading to significant estimation error. In this work, we address this problem by using a four-channel RGB-Distribution image, where the individual energy distribution map is obtained by applying food localization results on the entire food energy distribution map generated using conditional GAN.

5.2.2 Proposed Method

Food Localization and Classification

The goal of food localization is to locate individual food region for a given input image by providing a bounding box, where each bounding box should contain only one food item. Deep learning based methods for localization such as Faster R-CNN have shown success in many computer vision applications. It proposes potential regions that may contain the object with bounding boxes. Advanced CNN architectures such as VGG [54] and ResNet [53] can be used as the backbone structure for these methods.

The localization network locates all individual food items within the input food image and then sends them to the classification network. We apply Convolutional Neural Networks (CNNs) to classify the food item within each bounding box, which has been widely used in image classification applications. We use cross-entropy loss $\mathcal{L}_c$ for classification task as shown below:

$$\mathcal{L}_c = \sum_{\mathrm{i}=1}^{n} -\hat{y}^{(\mathrm{i})} log[f_c^{(\mathrm{i})}(\mathbf{x})] \tag{5.10}$$

where $\mathbf{x}$ is the cropped food image and $\hat{y}$ is its corresponding one hot label for the food category, f_c denotes the output of classification with dimension n. The food localization and classification pipeline are described in Figure 5.2.

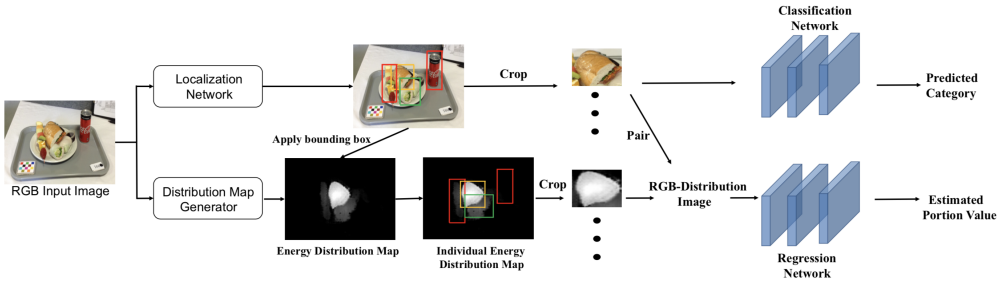


Figure 5.2. The overview of our proposed end-to-end framework that integrates food localization, classification and portion size estimation. Given an input eating occasion image, the localization network locates each individual food item by generating a bounding box around the food region. Meanwhile, an energy distribution map is generated using conditional GAN. Then we directly apply a generated bounding box on an energy distribution map to get a corresponding energy distribution map for each food item. The cropped RGB food image is sent to a classification network to predict the food category. It is also used to generate the four-channel RGB-Distribution image by pairing the cropped RGB image with an individual energy distribution map, which are sent to a regression network to estimate portion size value.

Food Portion Size Estimation

Portion size is a property that strongly relates to the presence of an object in 3D space, so it is very difficult to accurately estimate an object's portion size by given an arbitrary 2D image. In [27], a synthetic intermediate result of 'energy distribution' image was proposed, where the 'energy distribution' image has pixel-to-pixel correspondence and weights at different pixel locations to represent how food energy is distributed in the eating occasion. For example, pixels corresponding to steak have much higher weights than pixels of apple. [26] then uses the generated distribution image to estimate food portion size by applying a regression network. On the other hand, [20] uses RGB food image only and apply feature adaptation to estimate food portion size. Our method combines the two methods and use a RGB-Distribution image to improve the estimate of the food portion size.

Generate energy distribution map: We first train an energy distribution map generator by using a Generative Adversarial Networks [142] under conditional settings [143]. We define:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{5.11}$$

where G is the generator, D is the discriminator, $\mathcal{L}_{L1}(G)$ is the L1 reconstruction loss, and $\mathcal{L}_{cGAN}(G, D)$ is the conditional GAN loss as defined in [143]:

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x}, \mathbf{y})] + \\ \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x}),\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z}))] \end{aligned} \tag{5.12}$$

where $\mathbf{x}$ is the source domain (RGB image), $\mathbf{y}$ is the target domain (energy distribution map) and $\mathbf{z}$ is random noise. The energy distribution map is a single-channel image where higher pixel value indicates higher energy distribution.

Apply food localization bounding box: After we generate the energy distribution map for the entire eating occasion food image, we apply the generated bounding box generated to obtain the energy distribution map for individual food item.

Generate RGB-Distribution image: We then combine the cropped RGB single food image with its corresponding energy distribution map to generate a RGB-Distribution image, which has four channels: R, G, B, and distribution map. The RGB-Distribution image is sent to a regression network to estimate food portion size. L1-norm loss $\mathcal{L}_r$ is used for portion size estimation:

$$\mathcal{L}_r = |\hat{y} - f_r(x)| \tag{5.13}$$

where $\hat{y}$ is the groundtruth portion size value and f_r denotes the output of regression network with dimension 1. The lower half of Figure 5.2 shows the pipeline for estimating portion size for each individual food item.

5.2.3 Experiment

In this section, we evaluate our proposed end-to-end framework. For the localization and classification tasks, mean Average Precision (mAP) is the most common performance

metrics. We firstly define several related terminologies: The intersection of union (IoU) refers to the ratio of overlapped region between predicted bounding box and groundtruth bounding box over the union of the two bounding boxes. True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). For example, TP means the predicted bounding box is assigned with correct food label and the corresponding IoU socre is larger than a threshold. Based on these definitions, we can calculate precision (Equation 5.14) and recall (Equation 5.15).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.14}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.15}$$

Average Precision (AP) for each category is the average precision value for recall value over 0 to 1 for each food category, and mAP is the mean value of all APs of all categories.

Since we use L1-norm loss as shown in Equation 5.13 to train the regression network, we use the Mean Absolute Error (MAE) to evaluate portion size estimation, defined as

$$\text{MAE} = \frac{1}{N}\sum_{\text{i}=1}^{N}|w_{\text{i}} - \bar{w}_{\text{i}}| \tag{5.16}$$

where w_{i} is the estimated portion size of the i-th image, $\bar{w}_{\text{i}}$ is the groundtruth portion size of the i-th image and N is the number of testing images.

Dataset

Annotated image datasets have been instrumental for driving progress in many deep learning based applications such as food detection and classification. Existing food images datasets may contain groundtruth bounding box and food label information [144], [145] or just the food label [146], [147] which is not suitable for portion size estimation due to the lack of groundtruth information. In this paper, we introduce an eating occasion image to food energy dataset containing bounding box information, food category and portion size value. Food images were collected from a nutrition study as part of an image-assisted 24-hour dietary recall (24HR) study [148] conducted by registered dietitians. The study participants

were healthy volunteers aged between 18 and 70 years old. A mobile app was used to capture images of the eating scenes for 3 meals (breakfast, lunch and dinner) over a 24-hour period. Foods are provided in buffet style where pre-weighed foods and beverages are served to the participants. Based on the known foods and their weight, food energy is calculated and used as groundtruth. The dataset contains 154 annotated eating occasion images, with a total of 915 individual food images which belong to 31 categories. The corresponding groundtruth information includes bounding box to locate individual food, food category and portion size (in Kcal). We split the dataset with 15% for validation 15% for testing and the remaining for training. The problem with a small dataset is that the models trained on them cannot generalize well for data from the validation and test set. Hence, these models suffer from the problem of overfitting. Data augmentation is an efficient way to address this problem, where we increase the amount of training data by rotation (90 degrees, 270 degrees) and flip (x-axis, y-axis, both). We randomly implemented the operations based on the number of training images for that category, i.e. we implemented less operations for the category which contains more images. We augment the training data while keeping the groundtruth information unchanged before and after the augmentation operations.

Implementation Detail

Our implementation is based on Pytorch [55]. ResNet-50 is used as the backbone of Faster R-CNN. For regression network, a standard 18-layer ResNet is applied. The ResNet implementation follows the setting suggested in [53].

Results for localization and classification

The mAP results for food localization and classification tasks on our proposed dataset under different thresholds are shown in Table 5.2. 0.5 is commonly used and practical IoU threshold and we achieve satisfactory but 0.75 is a challenging threshold as we set the threshold of $IoU > 0.75$. In addition, our dataset is challenging since the number of training data is insufficient although some data augmentation methods are implemented. We also

calculate the the mAP by changing the IoU threshold from 0.5 to 0.95 with a step size of 0.05 as shown in last column.

Table 5.2. mAP results for food localization and classification on our introduced dataset. mAP@.5 and mAP@.75 indicate IoU larger than 0.5 and 0.75 respectively. mAP@[.5,.95] calculates AP for IoU from 0.5 to 0.95 with step size of 0.05.

mAP@.5	mAP@.75	mAP@[.5,.95]
0.6235	0.2428	0.2919

5.2.4 Results for portion size estimation

Compare to state-of-the-art methods: We compare our result of food portion size estimation with two state-of-the-art food portion estimation methods: [26] and [20] that directly using food distribution map or single RGB image for regression respectively. The input for our proposed method to estimate food portion size is a generated RGB-Distribution image of cropped RGB image and cropped energy distribution image using the localization network. As shown in Table 5.3, our method outperforms the other two methods for single food item portion size estimation with smallest MAE as our proposed method takes into consideration for both the RGB and energy distribution information.

Table 5.3. MAE results for food portion size estimation on our introduced dataset. Best result is marked in bold.

Methods	Mean Absolute Error (MAE)
Fang *et al.* [26]	109.94 Kcal
He *et al.* [20]	107.55 Kcal
Our Method	**105.64** Kcal

Compare to human estimates: We also compare our results for food portion size estimation of the entire eating occasion image that containing multiple single food items with 15 participants' estimates from the same study. During the data collecting time, the participants are required to estimate the portion size of the meal they just consumed in a structured interview while viewing the eating occasion images. We sum up all single food portion size estimated by our proposed method, [26] , [20] and human estimates respectively

for each eating occasion image. We apply error percentage as metric in this part which is defined as

$$EP = \frac{\sum_{i=0}^{N} |w_i - \hat{w}_i|}{\sum_{i=0}^{N} \hat{w}_i} \times 100\% \tag{5.17}$$

where w_i is the estimated portion size and $\hat{w}_i$ is the groundtruth portion size.

Table 5.4. Error percentage for food portion size estimation. Best result is marked in bold.

Methods	Error Percentage
Human Estimates	62.14%
Fang *et al.* [26]	35.06%
He *et al.* [20]	25.32%
Our Method	**11.22%**

As shown in Table 5.4, the error percentage (EP) of human estimates is 62.14%, which also indicates that predicting food portion size using only information from food images is really an challenging task for majority of people. Our method gives the best result on EP for 11.22%, which improves more than 50% in terms of EP compared with human estimates. Figure 5.3 shows the results for each eating occasion image in test set. Our predicted energy (red dots) is most closest to the groundtruth energy (black line).

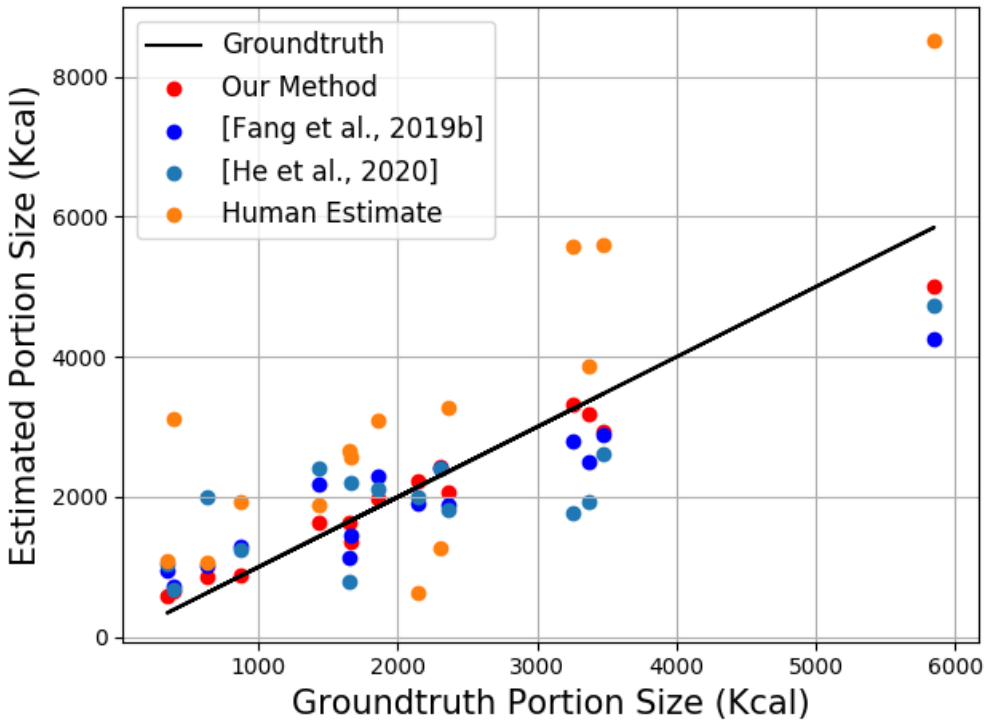


Figure 5.3. Food portion size estimation result for each eating occasion image in test set, where the dash line indicates the groundtruth and estimated energy are the same. The dots in different color shows the results for using different methods. (Best viewed in color)

6. SUMMARY AND FUTURE WORK

6.1 Continual Learning For Image Classification

In this thesis, we target on continual learning for image classification task. We study this problem from three perspectives including (1) online continual learning (2) unsupervised continual learning and (3) application based continual learning.

Online continual learning: we proposed an continual learning framework including a modified cross-distillation loss together with a two-step learning technique to address catastrophic forgetting in the challenging online learning scenario, and a simple yet effective method to update the exemplar set using the feature of each new observation of old classes data to mitigate concept drift. Our method has the following properties: (1) can be trained using data streams including both new classes data and new observations of old classes in online scenario, (2) has good performance for both new and old classes on future data streams, (3) requires short run-time to update with limited data, (4) has potential to be used in lifelong learning that can handle unknown number of classes incrementally. Our method outperforms current state-of-the-art on CIFAR-100 and ImageNet-1000 (ILSVRC 2012) in the challenging online learning scenario. Finally, we showed our proposed framework can be applied to real life image classification problem by using Food-101 dataset as an example and observed significant improvement compared to baseline methods. For future work, instead of performing only one phase experiment on Food-101, we will provide additional analysis when more incremental phases are included.

In addition, we proposed a novel and effective method for continual learning in online scenario under class-incremental setting by maintaining the classifier's discriminability for classes within each learned task and make final prediction through candidates selection together with prior incorporation using stored exemplars selected by our online sampler. Feature embedding instead of original data is stored as exemplars, which are both memory-efficient and privacy-preserving for real life applications and we further explore exemplar augmentation in feature space to achieve improved performance especially when given very limited storage capacity. Our method achieves best performance compared with existing online approaches on benchmark datasets including Split CIFAR10, Split CIFAR100 and

CORE-50. In addition, we vary the incremental step size and achieves comparable performance even with offline approaches on CIFAR-100. Finally, our analysis on norms of weight vectors in the classifier also shows great potential for addressing catastrophic forgetting in online scenario that can significantly reduce the weight bias problem. For future work, we plan to study the catastrophic forgetting from the feature space to analyze how the learned feature drifts when learning new classes.

Unsupervised continual learning: We explore a novel problem of unsupervised continual learning under class-incremental setting where the objective is to learn new classes incrementally while providing semantic meaningful clusters on all classes seen so far. We proposed a simple yet effective method using pseudo labels obtained based on cluster assignments to learn from unlabeled data for each incremental step. We introduced a new experimental protocol and evaluate our method on benchmark image classification datasets including CIFAR-100 and ImageNet (ILSVRC). We demonstrate that our method can be easily embedded with various existing supervised approaches implemented under both online and offline modes to achieve competitive performance in unsupervised scenario. Finally, we show that our proposed exemplar selection method works effectively without requiring ground truth and iteratively updating pseudo labels will cause performance degradation under continual learning context. However, one of the limitations of our method is that there lacks learning process of feature extractor towards new tasks. Therefore, our future work will apply self-supervised learning method to learn the visual representation from new tasks and then using the learned feature to perform clustering and obtain pseudo labels.

Besides, we formulated the problem of out-of-distribution detection in unsupervised continual learning (OOD-UCL) and introduce the corresponding evaluation protocol. Then a novel OOD detection method is proposed by correcting output bias and enhancing confidence difference between ID and OOD data. Our experimental results on CIFAR-100 show promising improvements compared with existing methods for various step sizes. For future work, instead of splitting the dataset with non-overlapped classes, we will focus on unsupervised continual learning in a more realistic scenario where each new task may contain both new classes and learned classes data. Therefore, a more efficient method that can perform continual learning based on the output of OOD detection is needed for real life applications.

Application based continual learning: We proposed a novel exemplar selection algorithm that selected representative data from each cluster generated based on visual similarity to alleviate the high intra-class variation problem of food images. In addition, an effective online learning regime was introduced using balanced training batch for old and new class and we proposed to apply knowledge distillation using contrastive training batch to help retain the learned knowledge. Our method achieved promising results on a challenging food dataset, Food1K, with significant performance improvement compared with existing state-of-the-art especially when the number of new food classes added for each incremental step increased, showing great potential for large scale continual learning of food image classification in real life.

Furthermore, We proposed a novel exemplar-free method by leveraging nearest-class-mean (NCM) classifier based on class mean estimated on all data seen so far during the training phase through online mean update criteria. In addition, we apply NCM on selected candidates only instead of all classes to improve the performance. Compared with state-of-the-arts, our method neither requires storing exemplars nor contains hyper-parameters tuning while still achieving promising results on CIFAR-100 by varying exemplar size $Q \in \{1,000, 2,000, 5,000, 10,000\}$ for existing approaches. Besides, we validate our method on the challenging Food-1k dataset and show improved performance for different incremental step sizes $M \in \{5, 10, 20\}$.

Our future work for application based continual learning will focus on varied incremental step size rather than using the fixed incremental step size (*i.e.* the number of new classes added for each step is same) as in existing experimental protocol. The varying of step size poses a new challenge for continual learning in addition to catastrophic forgetting and we plan to conduct experiments to analyze the performance of existing methods in this challenging scenario at first.

6.2 Image-Based Dietary Assessment

In this work, we proposed a multi-task framework for food classification and food portion size estimation by using L2-norm based soft parameter sharing. We also investigated cross-

domain feature adaptation together with different normalization techniques to further reduce portion estimation error. Our method is evaluated on a real life eating occasion food image dataset with groundtruth category and portion size provided by registered dietitians. Our best result achieved 88.67% classification accuracy, with the mean absolute errors of 56.82 Kcal for all food and 50.86 Kcal for correctly classified food for portion size estimation, surpassing the baseline results which are 86.08% and 62.27 Kcal respectively. In addition, we compared our portion estimation results with human estimates, showing an impressive 28.57% reduction in error percentage.

In addition, we propose an end-to-end image-based food analysis framework that integrates food localization, classification and portion size estimation. We introduce a novel method to estimate individual food portion size using RGB-Distribution image, where the individual energy distribution map is obtained by applying localization results on the entire energy distribution map generated by conditional GAN. Our framework is evaluated on a real life eating occasion food image dataset with groundtruth information of bounding box, food category and portion size. For localization and classification, we calculate the mAP under different thresholds and we show a satisfactory result. Our proposed method for food portion size estimation outperforms existing methods in terms of MAE as we consider both the RGB information and energy distribution information when estimating the portion size using a regression network. Our method also achieves the best improvement of error percentage from 62.14% to 11.22% when compared with human estimates for the entire eating occasion image, showing great potential for advancing the field of image-based dietary assessment.

Our future work for image-based dietary assessment will focus on developing more efficient portion size estimation method. Instead of directly perform regression to the portion value, we plan to apply metric learning and obtain the portion size based on maximum likelihood.

6.3 Contributions Of This Thesis

In this thesis, we proposed new methods to address catastrophic forgetting targeted on online continual learning, unsupervised continual learning and application based continual

learning. Besides, we designed end-to-end integrated food analysis system and introduce novel portions size estimation method for image-based dietary assessment. The main contributions are listed as follows:

- Online Continual Learning
  - We introduce a modified cross-distillation loss together with a two-step learning technique to address catastrophic forgetting in online scenario.
  - A continual learning framework is proposed, which is capable of lifelong learning and can be applied to a variety of real life online image classification problems where new data can belong to both new or learned class. We provide a simple yet effective method to mitigate concept drift by updating the exemplar set using the feature of each new observation of old classes.
  - Instead of using original data exemplars, we propose a simple yet effective method to store feature embeddings to reduce the memory burden and an online sampler is designed to select exemplars from sequentially available data stream through dynamic mean update criteria.
  - A novel candidates selection algorithm is introduced to mitigate forgetting in online scenario by reducing the output bias.
- Unsupervised Continual Learning
  - We explore a novel problem for continual learning using pseudo labels instead of human annotations, which is under-studied yet and a new benchmark evaluation protocol for unsupervised continual learning is introduced for future research work.
  - An unsupervised continual learning framework is proposed by using pseudo labels obtained from cluster assignments, which can be easily adapted by existing supervised continual learning techniques and we achieve competitive performance with supervised method but without human annotation.

  - We formulate the problem and proposed the corresponding evaluation protocol for out-of-distribution detection in unsupervised continual learning (OOD-UCL), which remains under-explored.
  - A novel method is introduced for OOD detection by correcting output bias and enhancing output confidence difference based on task discriminativeness.

- Application Based Continual Learning
  - To the best of our knowledge, we are the first to study online continual learning for food image classification task. We proposed a novel clustering based exemplar selection algorithm and a new online training regime to address catastrophic forgetting.
  - We proposed a novel exemplar-free online continual learning method by leveraging NCM classifier with class mean estimated on all data seen so far to reduce the memory burden and address privacy concerns in real life applications.

- Image-Based Dietary Assessment
  - We introduce a food image datasets collected from a nutrition study with the groundtruth food portion provided by registered dietitians.
  - A soft-parameter sharing multi-task framework is introduced for single-item food image analysis, which is capable of simultaneously food classification and portion size estimation.
  - We proposed to use four-channel RGB-Distribution food images and introduce an end-to-end food analysis system for multi-item food images by integrating localization, classification and portion size estimation.

6.4 Publications Resulting From This Thesis

- **Jiangpeng He**, Runyu Mao, Zeman Shao, Fengqing Zhu, "Incremental Learning In Online Scenario", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2020, Virtual Conference.

- **Jiangpeng He**, Zeman Shao, Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu,"Multi-Task Image-Based Dietary Assessment For Food Recognition And Portion Size Estimation", Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval, August 2020, Virtual Conference.

- **Jiangpeng He**, Runyu Mao, Zeman Shao, Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu,"An End-to-End Food Image Analysis System", Electronic Imaging, January 2021, Virtual Conference.

- **Jiangpeng He**, Fengqing Zhu, "Unsupervised Continual Learning Via Pseudo Labels", International Joint Conference on Artificial Intelligence, CSSL Workshop, August 2021, Virtual Conference.

- **Jiangpeng He**, Fengqing Zhu, "Online Continual Learning For Visual Food Classification", Proceedings of the IEEE International Conference on Computer Vision, Large Fine Food AI Workshop, October 2021, Virtual Conference.

- **Jiangpeng He**, Fengqing Zhu, "Online Continual Learning Via Candidates Voting", Proceedings of the IEEE Winter Conference on Applications of Computer Vision, January 2022, Hawaii.

- **Jiangpeng He**, Fengqing Zhu, "Out-Of-Distribution Detection In Unsupervised Continual Learning", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Continual AI Workshop, June 2022, New Orleans.

- **Jiangpeng He**, Fengqing Zhu, "Exemplar-Free Online Continual Learning", Proceedings of the IEEE International Conference on Image Processing, October 2022, Bordeaux, France.

6.5 Other Publications Not Related to This Thesis

- **Jiangpeng He**, Kyle Ziga, Judy Bagchi, Fengqing Zhu,"CNN Based Parameter Optimization for Texture Synthesis", Electronic Imaging, January 2019, San Francisco.

- Runyu Mao, **Jiangpeng He**, Zeman Shao, Sri Yarlagadda, Fengqing Zhu,"Visual Aware Hierarchy Based Food Recognition", Proceedings of International Conference on Pattern Recognition, Workshops and Challenges, January 2021, Virtual Conference.

- Zeman Shao, Shaobo Fang, Runyu Mao, **Jiangpeng He**,Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu,"Towards Learning Food Portion From Monocular Images With Cross-Domain Feature Adaptation", Proceedings of IEEE 23rd International Workshop on Multimedia Signal Processing, October 2021, Virtual Conference.

- Runyu Mao, **Jiangpeng He**, Luotao Lin, Zeman Shao, Heather Eicher-Miller, Fengqing Zhu,"Improving Dietary Assessment Via Integrated Hierarchy Food Classification", Proceedings of IEEE 23rd International Workshop on Multimedia Signal Processing, October 2021, Virtual Conference.

- Zeman Shao, Yue Han, **Jiangpeng He**, Runyu Mao, Janine Wright, Deborah Kerr, Carol Boushey, Fengqing Zhu,"An Integrated System for Mobile Image-Based Dietary Assessment", Proceedings of ACM International Multimedia Conference, Workshop of AIxFood, October 2021, Virtual Conference.

- Zeman Shao, **Jiangpeng He**, Ya-Yuan Yu, Luotao Lin, Alexandra Cowan, Heather Eicher-Miller, Fengqing Zhu,"Towards the Creation of a Nutrition and Food Group Based Image Database", Electronic Imaging, January 2022, Virtual Conference.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[2] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," *Technical Report*, 2009.

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

[4] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in psychology*, vol. 4, p. 504, 2013.

[5] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," *arXiv preprint arXiv:1810.12488*, 2018.

[6] D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, and B. E. Bejnordi, "Conditional channel gated networks for task-aware continual learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3931–3940, 2020.

[7] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," *Neural Networks*, vol. 116, pp. 56–73, 2019.

[8] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *arXiv preprint arXiv:2101.10423*, 2021.

[9] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Proceedings of the NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531.

[10] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2019.

[11] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," *Proceedings of THE International Conference on Machine Learning, Workshop on challenges in representation learning*, vol. 3, no. 2, 2013.

[12] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[13] W. Min, Z. Wang, Y. Liu, M. Luo, L. Kang, X. Wei, X. Wei, and S. Jiang, "Large scale visual food recognition," *CoRR*, vol. abs/2103.16107, 2021.

[14] R. Mao, J. He, Z. Shao, S. K. Yarlagadda, and F. Zhu, "Visual aware hierarchy based food recognition," *arXiv preprint arXiv:2012.03368*, 2020.

[15] H. Wu, M. Merler, R. Uceda-Sosa, and J. 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.

[16] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*, pp. 1–6, Jul. 2015.

[17] 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," *International Conference on Smart Homes and Health Telematics*, pp. 37–48, 2016.

[18] P. Pandey, A. Deepthi, B. Mandal, and N. B. Puhan, "Foodnet: Recognizing foods using ensemble of deep networks," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1758–1762, 2017.

[19] M. Bolaños and P. Radeva, "Simultaneous food localization and recognition," *2016 23rd International Conference on Pattern Recognition*, pp. 3140–3145, 2016.

[20] J. He, Z. Shao, J. Wright, D. Kerr, C. Boushey, and F. Zhu, "Multi-task image-based dietary assessment for food recognition and portion size estimation," *arXiv preprint arXiv:2004.13188*, 2020. arXiv: 2004.13188 [cs.CV].

[21] J. He, R. Mao, Z. Shao, and F. Zhu, "Incremental learning in online scenario," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13 926–13 935, 2020.

[22] K. Aizawa, Y. Maruyama, H. Li, and C. Morikawa, "Food balance estimation by using personal dietary tendencies in a multimedia Food Log," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2176–2185, Dec. 2013.

[23] S. Fang, C. Liu, F. Zhu, E. Delp, and C. Boushey, "Single-view food portion estimation based on geometric models," *Proceedings of the IEEE International Symposium on Multimedia*, pp. 385–390, Dec. 2015, Miami, FL.

[24] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2Calories: towards an automated mobile vision food diary," *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2015, Santiago, Chile.

[25] J. Dehais, A. Greenburg, S. Shevchick, A. Soni, M. Anthimpoulos, and S. Mougiakakou, "Estimation of food volume and carbs," *Google Patents*, Feb. 2018, US Patent 9,892,501.

[26] 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, p. 877, 2019.

[27] S. Fang, Z. Shao, R. Mao, C. Fu, E. J. Delp, F. Zhu, D. A. Kerr, and C. J. Boushey, "Single-view food portion estimation: Learning image-to-energy mappings using generative adversarial networks," *Proceedings of the IEEE International Conference on Image Processing*, pp. 251–255, Oct. 2018, Athens, Greece.

[28] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. eprint: 1411.1784.

[29] C. Xu, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Image enhancement and quality measures for dietary assessment using mobile devices," International Society for Optics and Photonics, vol. 8296, 2012, 82960Q.

[30] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state-of-the-art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[31] A. Royer and C. H. Lampert, "Classifier adaptation at prediction time," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1401–1409, 2015.

[32] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, 44:1–44:37, Mar. 2014, ISSN: 0360-0300. DOI: 10.1145/2523813. [Online]. Available: http://doi.acm.org/10.1145/2523813.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] S. Ruping, "Incremental learning with support vector machines," *Proceedings of the IEEE International Conference on Data Mining*, pp. 641–642, 2001.

[35] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 409–415, 2001.

[36] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 31, no. 4, pp. 497–508, 2001.

[37] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2624–2637, 2013.

[38] I. Kuzborskij, F. Orabona, and B. Caputo, "From n to $n + 1$: Multiclass transfer incremental learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3358–3365, 2013.

[39] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," *arXiv preprint arXiv:1607.00122*, 2016.

[40] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *The National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[41] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[42] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1320–1328, 2017.

[43] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2990–2999, 2017.

[44] R. Venkatesan, H. Venkateswara, S. Panchanathan, and B. Li, "A strategy for an uncompromising incremental learner," *arXiv preprint arXiv:1705.00744*, 2017.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[46] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017.

[47] F. M. Castro, M. J. Marin-Jimenez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," *Proceedings of the European Conference on Computer Vision*, Sep. 2018.

[48] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016.

[49] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool, "Incremental learning of ncm forests for large-scale image classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3654–3661, 2014.

[50] S. Guerriero, B. Caputo, and T. Mensink, "Deep nearest class mean classifiers," *Proceedings of the International Conference on Learning Representations, Worskhop Track*, 2018.

[51] M. Welling, "Herding dynamical weights to learn," *Proceedings of the International Conference on Machine Learning*, pp. 1121–1128, 2009.

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

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

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

[55] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," *Proceedings of the Advances Neural Information Processing Systems Workshop*, 2017.

[56] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *arXiv preprint arXiv:1812.00420*, 2018.

[57] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial shapley value," *arXiv preprint arXiv:2009.00093*, 2020.

[58] A. Prabhu, P. H. Torr, and P. K. Dokania, "Gdumb: A simple approach that questions our progress in continual learning," *Proceedings of the European Conference on Computer Vision*, pp. 524–540, 2020.

[59] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," *Advances in Neural Information Processing Systems*, pp. 11 849–11 860, 2019.

[60] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13 208–13 217, 2020.

[61] D. F. Specht, "Probabilistic neural networks," *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.

[62] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," *arXiv preprint arXiv:1607.00122*, 2016.

[63] S. Hou, X. Pan, C. Change Loy, Z. Wang, and D. Lin, "Lifelong learning via progressive distillation and retrospection," *Proceedings of the European Conference on Computer Vision*, pp. 437–452, 2018.

[64] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

[65] K. Lee, K. Lee, J. Shin, and H. Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 312–321, 2019.

[66] J. He and F. Zhu, "Online continual learning for visual food classification," *arXiv preprint arXiv:2108.06781*, 2021.

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

[68] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, pp. 6467–6476, 2017.

[69] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, and Q. Sun, "Mnemonics training: Multi-class incremental learning without forgetting," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12 245–12 254, 2020.

[70] J. Pomponi, S. Scardapane, V. Lomonaco, and A. Uncini, "Efficient continual learning in neural networks with embedding regularization," *Neurocomputing*, vol. 397, pp. 139–148, 2020.

[71] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/e562cd9c0768d5464b64cf61da7fc6bb-Paper.pdf.

[72] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[73] A. Chaudhry, A. Gordo, P. K. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," *arXiv preprint arXiv:2002.08165*, 2020.

[74] Y. Xiang, Y. Fu, P. Ji, and H. Huang, "Incremental learning using conditional adversarial networks," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6619–6628, 2019.

[75] R. Kemker and C. Kanan, "Fearnet: Brain-inspired model for incremental learning," *arXiv preprint arXiv:1711.10563*, 2017.

[76] S. Guerriero, B. Caputo, and T. Mensink, "Deepncm: Deep nearest class mean classifiers," 2018.

[77] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," *arXiv preprint arXiv:1702.05538*, 2017.

[78] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Online continual learning with no task boundaries," *arXiv preprint arXiv:1903.08671*, 2019.

[79] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," *International Conference on Machine Learning*, pp. 3987–3995, 2017.

[80] V. Lomanco and D. Maltoni, "Core50: A new dataset and benchmark for continual object recognition," *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 17–26, 2017.

[81] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation," *arXiv preprint arXiv:2010.15277*, 2020.

[82] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," *Proceedings of the European Conference on Computer Vision*, pp. 132–149, 2018.

[83] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, "Online deep clustering for unsupervised representation learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6688–6697, 2020.

[84] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[85] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Scan: Learning to classify images without labels," *European Conference on Computer Vision*, pp. 268–285, 2020.

[86] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[87] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[88] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[89] C. S. Sastry and S. Oore, "Detecting out-of-distribution examples with gram matrices," *International Conference on Machine Learning*, pp. 8491–8501, 2020.

[90] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, "Uncertainty estimation using a single deep deterministic neural network," *International conference on machine learning*, pp. 9690–9700, 2020.

[91] Y. Wang, B. Li, T. Che, K. Zhou, Z. Liu, and D. Li, "Energy-based open-world uncertainty modeling for confidence calibration," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9302–9311, 2021.

[92] M. Hein, M. Andriushchenko, and J. Bitterwolf, "Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2019.

[93] J. Nandy, W. Hsu, and M. L. Lee, "Towards maximizing the representation gap between in-domain & out-of-distribution examples," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9239–9250, 2020.

[94] M. Mundt, I. Pliushch, S. Majumder, Y. Hong, and V. Ramesh, "Unified probabilistic deep continual learning through generative replay and open set recognition," *Journal of Imaging*, vol. 8, no. 4, p. 93, Mar. 2022. DOI: 10.3390/jimaging8040093. [Online]. Available: https://doi.org/10.3390%2Fjimaging8040093.

[95] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv preprint arXiv:2110.11334*, 2021.

[96] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proceedings of International Conference on Learning Representations*, 2017.

[97] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *Proceedings of International Conference on Learning Representations*, 2018.

[98] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," *Advances in Neural Information Processing Systems*, 2020.

[99] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3903–3911, 2020.

[100] S. Stojanov, S. Mishra, N. A. Thai, N. Dhanda, A. Humayun, C. Yu, L. B. Smith, and J. M. Rehg, "Incremental object learning from contiguous views," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8777–8786, 2019.

[101] D. Rao, F. Visin, A. A. Rusu, Y. W. Teh, R. Pascanu, and R. Hadsell, "Continual unsupervised representation learning," *arXiv preprint arXiv:1910.14481*, 2019.

[102] J. Smith, S. Baer, C. Taylor, and C. Dovrolis, "Unsupervised progressive learning and the stam architecture," *arXiv preprint arXiv:1904.02021*, 2019.

[103] J. He and F. Zhu, "Unsupervised continual learning via pseudo labels," *arXiv preprint arXiv:2104.07164*, 2021.

[104] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.

[105] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, e0118432, 2015.

[106] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *arXiv preprint arXiv:1701.06548*, 2017.

[107] J. He and F. Zhu, "Online continual learning for visual food classification," *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 2337–2346, Oct. 2021.

[108] J. He and F. Zhu, "Online continual learning via candidates voting," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3154–3163, Jan. 2022.

[109] J. He and F. Zhu, "Exemplar-free online continual learning," *arXiv preprint arXiv:2202.05491*, 2022.

[110] C. Boushey, M. Spoden, F. Zhu, E. Delp, and D. 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.

[111] H. Wu, M. Merler, R. Uceda-Sosa, and J. 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.

[112] W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, X. Wei, and S. Jiang, "Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network," *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 393–401, 2020.

[113] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[114] T. L. Hayes, N. D. Cahill, and C. Kanan, "Memory efficient experience replay for streaming learning," *Proceedings of the International Conference on Robotics and Automation*, pp. 9769–9776, 2019.

[115] L. Lin, F. Zhu, E. Delp, and H. Eicher-Miller, "The most frequently consumed and the largest energy contributing foods of us insulin takers using nhanes 2009–2016," *Current Developments in Nutrition*, vol. 5, pp. 426–426, 2021.

[116] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[117] F. Lin and W. W. Cohen, "Power iteration clustering," *Proceedings of International Conference on Machine Learning*, 2010.

[118] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," *Proceedings of 2010 IEEE International Symposium on Multimedia*, pp. 296–301, Dec. 2010. DOI: 10.1109/ISM.2010.51.

[119] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, 1999.

[120] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, pp. 1085–1088, 2014, Orlando, Florida, USA.

[121] R. Tanno, K. Okamoto, and K. Yanai, "Deepfoodcam: A dcnn-based real-time mobile food recognition system," *Proceedings of the 2Nd International Workshop on Multimedia Assisted Dietary Management*, MADiMa '16, pp. 89–89, 2016. DOI: 10.1145/2986035.2986044.

[122] N. Martinel, G. L. Foresti, and C. Micheloni, "Wide-slice residual networks for food recognition," *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, pp. 567–576, Mar. 2018. DOI: 10.1109/WACV.2018.00068.

[123] J. He, R. Mao, Z. Shao, J. L. Wright, D. A. Kerr, C. J. Boushey, and F. Zhu, "An end-to-end food image analysis system," *Electronic Imaging*, vol. 2021, no. 8, pp. 285-1-285–7, 2021, ISSN: 2470-1173. DOI: doi:10.2352/ISSN.2470-1173.2021.8.IMAWM-285". [Online]. Available: https://www.ingentaconnect.com/content/ist/ei/2021/00002021/00000008/art00011.

[124] Z. Shao, S. Fang, R. Mao, J. He, J. Wright, D. Kerr, C. J. Boushey, and F. Zhu, "Towards learning food portion from monocular images with cross-domain feature adaptation," *arXiv preprint arXiv:2103.07562*, 2021.

[125] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2624–2637, 2013.

[126] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software*, vol. 11, no. 1, pp. 37–57, 1985.

[127] M. Douze, H. Jégou, and J. Johnson, "An evaluation of large-scale methods for image instance and class discovery," *Proceedings of the on Thematic Workshops of ACM Multimedia*, pp. 1–9, 2017.

[128] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[129] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/15825aee15eb335cc13f9b559f166ee8-Paper.pdf.

[130] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016. arXiv: 1607.06450 [stat.ML].

[131] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. arXiv: 1502.03167 [cs.LG].

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

[133] Y. Kawano and K. Yanai, "Foodcam: A real-time food recognition system on a smartphone," *Multimedia Tools and Applications*, 2014.

[134] 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.

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

[136] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[137] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task cnn model for attribute prediction," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1949–1959, Nov. 2015.

[138] T. Ege and K. Yanai, "Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions," *Proceedings of the Workshops of ACM Multimedia on Thematic*, pp. 367–375, 2017, Mountain View, CA.

[139] L. Duong, T. Cohn, S. Bird, and P. Cook, "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 845–850, 2015.

[140] Y. Yang and T. M. Hospedales, "Trace norm regularised deep multi-task learning," *arXiv preprint arXiv:1606.04038*, 2016.

[141] C. J. Boushey, M. Spoden, F. M. Zhu, E. J. Delp, and D. A. 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, Aug. 2017.

[142] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, Dec. 2014, Montreal, Canada.

[143] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5967–5976, Jul. 2017, Honolulu, HI.

[144] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," *Proceedings of European Conference on Computer Vision Workshops*, pp. 3–17, Sep. 2014, Zurich, Switzerland.

[145] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 25–30, Jul. 2012, Melbourne, Australia.

[146] L. Bossard, M. Guillaumin, and L. V. Gool, "Food-101 – mining discriminative components with random forests," *Proceedings of European Conference on Computer Vision*, vol. 8694, pp. 446–461, Sep. 2014, Zurich, Switzerland.

[147] 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, Jun. 2015. DOI: 10.1109/ICMEW.2015.7169757.

[148] C. Schipp, J. Wright, C. Boushy, E. Delp, S. Dhaliwal, and D. Kerr, "Can images improve portion size estimation of the asa24 image-assisted food recall: A controlled feeding study," *Nutrition & Dietetics; 75 (Suppl. 1): 107*, 2018. DOI: 10.1111/1747-0080.12427.

VITA

Jiangpeng He was born in Shijiazhuang, China on February 6, 1995. He received the Bachelor of Science degree in Electronic and Electrical Engineering from University of Electronic Science and Technology of China, Chengdu, China. Mr. He then joined the Ph.D. program at the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana in August 2017. He worked at the Video and Image Processing Laboratory (VIPER) under the supervision of Professor Fengqing M. Zhu. While pursuing his Ph.D. at Purdue, he primarily worked on projects sponsored by the *Eli Lilly and Company*.

His research interests are image processing, computer vision, and deep learning. He is a student member of the CVF, the IEEE, the IEEE Computer Society, and the IEEE Signal Processing Society. He has served as the reviewer of the IEEE International Conference on Image Processing, IEEE Conference on Computer Vision and Pattern Recognition, IEEE International Conference on Computer Vision, European Conference on Computer Vision and the IEEE Winter Conference on Applications of Computer Vision.