Leveraging Old Knowledge to Continually Learn New Classes in Medical Images

Evelyn Chee[1,2], **Mong Li Lee**[1,2], **Wynne Hsu**[1,2]

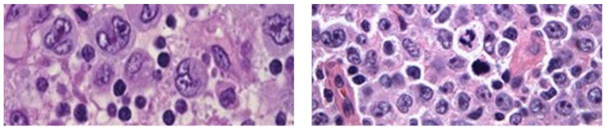
[1] School of Computing, National University of Singapore
[2] Institute of Data Science, National University of Singapore
{echee, leeml, whsu}@comp.nus.edu.sg

Abstract

Class-incremental continual learning is a core step towards developing artificial intelligence systems that can continuously adapt to changes in the environment by learning new concepts without forgetting those previously learned. This is especially needed in the medical domain where continually learning from new incoming data is required to classify an expanded set of diseases. In this work, we focus on how old knowledge can be leveraged to learn new classes without catastrophic forgetting. We propose a framework that comprises of two main components: (1) a dynamic architecture with expanding representations to preserve previously learned features and accommodate new features; and (2) a training procedure alternating between two objectives to balance the learning of new features while maintaining the model's performance on old classes. Experiment results on multiple medical datasets show that our solution is able to achieve superior performance over state-of-the-art baselines in terms of class accuracy and forgetting.

Introduction

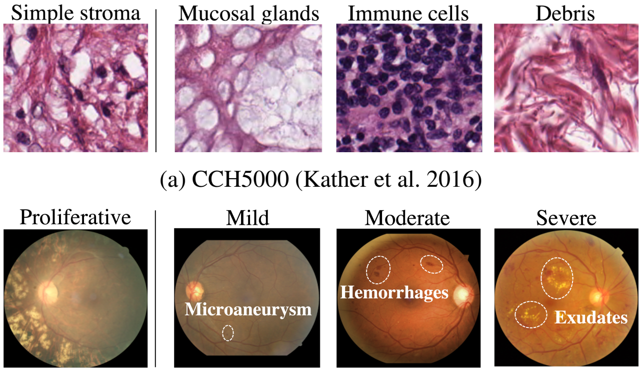
Deep neural networks (DNNs) have excelled in many machine learning classification tasks and shown to achieve human-level performance in medical imaging applications (McKinney et al. 2020; Ardila et al. 2019; Esteva et al. 2017). However, this is under the condition that all classes are known prior to training. This assumption is often violated in the medical domain as some diseases are rare and discovering new disease is not uncommon. In both cases, not all classes would have readily available data for training the DNN. Figure 1 shows samples of two types of lymphoma with the latter being a rarer disease (El-Mallawany et al. 2012). A system that has only learn to diagnose the former can easily misclassify the other as the same type with high confidence due to their similar appearance. This can have dire consequences as the treatment strategies for the two are different and initiating the correct regimen is crucial to achieve good outcome. Hence, any DNN systems in the medical domain must be able to continually learn an expanding set of classes as and when new data becomes available. Further, they should do so without negatively affecting the performance for diagnosing previously seen diseases.



anaplastic large cell lymphoma (ALCL)

peripheral T-cell lymphoma-not otherwise specified (PTCL-NOS)

Figure 1: Sample pathology images of different lymphomas (El-Mallawany et al. 2012).



(a) CCH5000 (Kather et al. 2016)

(b) EyePACS (Kaggle and EyePacs 2015)

Figure 2: Sample images illustrating new class (first image) have shared features from old classes.

Existing work on class-incremental continual learning focuses on catastrophic forgetting (McCloskey and Cohen 1989; Goodfellow et al. 2014), where performance of DNNs degrades significantly on previously learned classes as more classes are being introduced. For the medical domain, we should in fact go beyond just alleviating forgetting. As shown in our previous example, one observation is that diseases tend to have overlapping features. Particularly, classification of a new class could depend on features from various old classes. Figure 2(a) shows the structure of simple stroma that could be described using combination of texture from other tissues. Similarly, the progression of a disease includes earlier clinical symptoms. Figure 2(b) shows that the proliferative stage of diabetic retinopathy includes clinical signs from mild, moderate and severe stages. This motivates us to explore how we can leverage on the previously learned knowledge to acquire new features that allow us to discriminate the expanding set of classes, and possibly even lead to performance improvement on the old disease classes.

There has been limited research on class-incremental continual learning in the medical domain. Li et al. (2020) introduced an ensemble strategy to update representation of old classes. However, it still lacks in terms of utilizing previously learned features. A Bayesian generative model for medical use cases is proposed by Yang et al. (2021) where old classes are represented using statistical distributions. It allows the model to preserve old knowledge, but faces difficulty learning new features due to the fixed pre-trained feature extractor.

In this work, we design a framework that leverages on what has been learned to derive new features for recognizing the cumulative set of classes. We employ a dynamic architecture that allows features from the old classes to be reused while expanding the set of feature extractors to learn novel features from the new classes. The proposed architecture has a single low-level feature extractor shared across all classes, which helps to promote utilization of previously learned knowledge. We propose a new training strategy that alternates between two objective functions, one focusing on new classes and the other on the old. With this, the classifier can perform well on new classes while still able to maintain, or even improve, its performance on old classes.

We validate our proposed framework on three publicly available medical datasets. Empirical results show that our approach outperforms state-of-the-art methods, especially when the dataset is highly skewed and the incremental classes per step is small. Aside from being able to alleviate forgetting, the results show that we are able to utilize newly learned features to better discriminate old classes.

Related Work

Research on class-incremental continual learning has largely been concentrated in the natural image domain. These works can broadly be categorized into three approaches: regularization-based, replay-based, and architecture-based.

Regularization-based approach preserves old knowledge by using additional regularization term to penalize changes in previously learned features. Previous works (Kirkpatrick et al. 2017; Zenke, Poole, and Ganguli 2017; Aljundi et al. 2018) estimate the importance of each weight in a previously learned model and apply penalty if there are updates to the important weights. Other works use distillation loss to ensure that the features learned of the old classes are preserved (Li and Hoiem 2017; Hou et al. 2018; Douillard et al. 2020; Kim and Choi 2021). Another interesting direction introduced in recent work (Tao et al. 2020) is the use of topology-preserving loss to maintain the feature space topology. However, one difficulty faced by approaches in this category is balancing the regularization term such that learning of new classes would not be hindered.

Data replay-based methods interleaves data related to previous tasks with new data so that the model is 'reminded' of the old knowledge. One way to obtain the old data is by training deep generative models to synthesize 'fake' samples that mimics data from previous classes (Shin et al. 2017; Wu et al. 2018; Rao et al. 2019; Wang et al. 2022) but there is the issue of generating realistic data samples. Another way is to directly store a small number of previous class samples to train together with the new data (Rebuffi et al. 2017; Chaudhry et al. 2019). However, due to the limited memory buffer, there is an imbalance between the small number of old class samples and the relatively large amount of data for new classes. Attempts to address this issue include employing a two-stage learning where the classifier is fine-tuned using a balanced dataset (Castro et al. 2018) or by correcting the biased weights after training (Wu et al. 2019; Zhao et al. 2020). On the other hand, a few works focus on how best to select data such that old class performance could be maintained. Particularly, Aljundi et al. (2019) proposed to store data instances that suffer most by the parameters update of the new model, while Shim et al. (2021) selects samples that best preserve decision boundaries between classes. Other works propose to parameterize and construct the representational data of each seen class through bi-level optimization (Liu et al. 2020; Chaudhry et al. 2021).

Architecture-based approaches focus on dynamically expanding the network structure and allocate new model parameters to accommodate new information while keeping previously learned parameters fixed to preserve old knowledge (Hung et al. 2019; Mallya, Davis, and Lazebnik 2018; Fernando et al. 2019; Yoon et al. 2017). Most of these methods use different part of the network for each task which requires task identity during inference, but this identity information is usually unavailable. The work by Yan, Xie, and He (2021) introduces the notion of expandable representation learning by adding new branches to the feature extractor of existing network to learn novel concepts for the incoming data while fixing the weights of old branches to preserve previously learned features. However, there is minimal exploitation of old knowledge since each feature extractor branch is independent of each other.

Proposed Approach

In class-incremental continual learning, the model is required to learn from a stream of data. At each incremental step $t \in [1..T]$, let Y_t be the set of new classes and D_t be the dataset comprising samples $(\boldsymbol{x}, y)$, where $\boldsymbol{x}$ denotes an input image and $y \in Y_t$ is the corresponding label. The goal is to maximize the overall classification accuracy of all seen classes up to step t, i.e. $Y_{[1:t]} = \cup_{i=1}^{t} Y_i$.

Our proposed framework leverages previously learned features to learn new classes. We utilize a dynamically expanding network to accommodate new features without compromising old ones. To maintain performance on previously seen classes, regularization loss and data replay strategy are also employed. Further, to handle the highly skewed class distribution, we use cost-sensitive learning (Ting 2000) and assign higher penalty to samples from under-represented old classes.

Model Architecture

Figure 3 shows the details of our proposed dynamic architecture at incremental step t. There are three main compo-

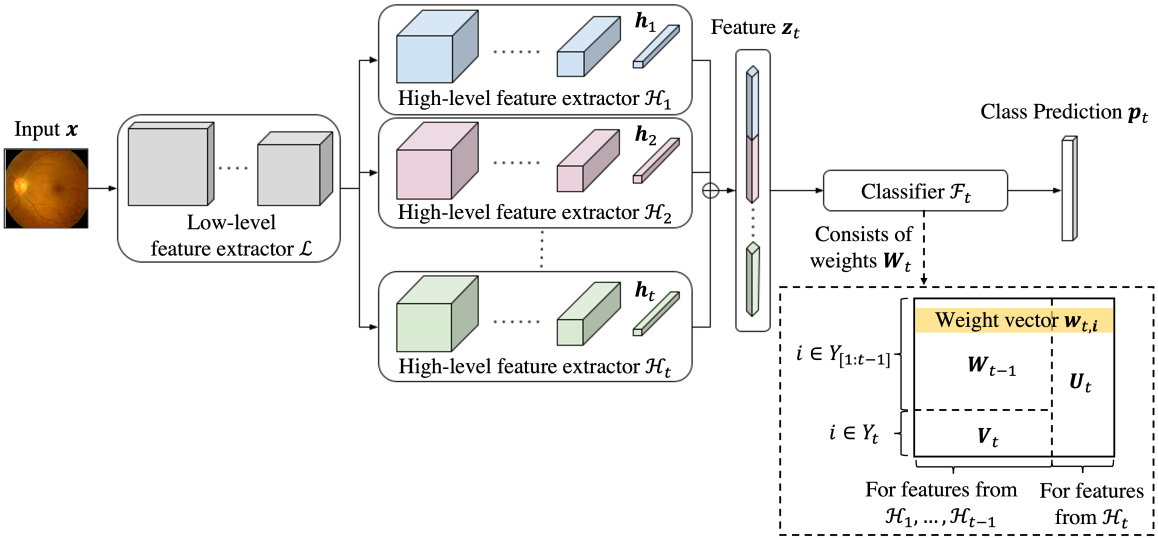

Input $\boldsymbol{x}$
Low-level feature extractor $\mathcal{L}$
$\boldsymbol{h}_1$
High-level feature extractor $\mathcal{H}_1$
$\boldsymbol{h}_2$
High-level feature extractor $\mathcal{H}_2$
$\boldsymbol{h}_t$
High-level feature extractor $\mathcal{H}_t$
Feature $\boldsymbol{z}_t$
Classifier $\mathcal{F}_t$
Class Prediction $\boldsymbol{p}_t$
Consists of weights $\boldsymbol{W}_t$
Weight vector $\boldsymbol{w}_{t,i}$
$i \in Y_{[1:t-1]}$
$\boldsymbol{W}_{t-1}$
$\boldsymbol{U}_t$
$i \in Y_t$
$\boldsymbol{V}_t$
For features from $\mathcal{H}_1, \dots, \mathcal{H}_{t-1}$
For features from $\mathcal{H}_t$

Figure 3: Model architecture at incremental step t, consisting of low-level feature extractor $\mathcal{L}$, high-level feature extractors $\{\mathcal{H}_1, \dots, \mathcal{H}_t\}$ and unified classifier $\mathcal{F}_t$. Low-level features learned are shared across all tasks and a new high-level feature extractor $\mathcal{H}_t$ is added each step t. The classifier $\mathcal{F}_t$ consists of weights with vectors $\boldsymbol{w}_{t,i}$ for all classes $i \in Y_{[1:t]}$.

nents: a low-level feature extractor, a set of high-level feature extractors, and a unified classifier. The low-level feature extractor $\mathcal{L}$, parameterized by $\boldsymbol{\theta}_{\mathcal{L}}$, is shared throughout the learning process. Each high-level feature extractor $\mathcal{H}_k$ has the same architecture that receives processed input from $\mathcal{L}$ and builds upon the low-level features to learn discriminating features for the new set of classes. A new high-level feature extractor $\mathcal{H}_k$, parameterized by $\boldsymbol{\theta}_{\mathcal{H}_k}$, is added to the model at each step t. By allowing the low-level feature extractor to be shared across various tasks, our approach encourages the reuse of similar features by the high-level feature extractors. The output is a vector $\boldsymbol{h}_k$ of dimension d. To alleviate forgetting, we freeze the set of old parameters $\{\boldsymbol{\theta}_{\mathcal{L}}, \boldsymbol{\theta}_{\mathcal{H}_1}, \dots, \boldsymbol{\theta}_{\mathcal{H}_{t-1}}\}$.

The unified classifier $\mathcal{F}_t$ is a single layer with weight matrix $\boldsymbol{W}_t$ comprising of vectors $\boldsymbol{w}_{t,i}$ for each class $i \in Y_{[1:t]}$. These vectors are derived from the weight matrix of previous step $t-1$ by expanding row-wise to accommodate new classes Y_t and column-wise to include new features from $\mathcal{H}_t$. Let $\boldsymbol{W}_{t-1}$ be the weight matrix comprising of vectors $\boldsymbol{w}_{t-1,i}$ of all seen classes $i \in Y_{[1:t-1]}$. Then the matrix $\boldsymbol{W}_t$ is given by $\boldsymbol{W}_t = \big[[\boldsymbol{W}_{t-1} \circ \boldsymbol{V}_t]; \boldsymbol{U}_t\big]$, where $\boldsymbol{U}_t$ is a matrix with weight vectors corresponding to features from $\mathcal{H}_t$ for all classes in $Y_{[1:t]}$ and $\boldsymbol{V}_t$ is a matrix of feature weights from $\mathcal{H}_1$ up to $\mathcal{H}_{t-1}$ for new classes in Y_t.

Suppose $\boldsymbol{z}_t = [\boldsymbol{h}_1 \circ \boldsymbol{h}_2 \circ \cdots \circ \boldsymbol{h}_t]$ is the concatenated outputs from all high-level feature extractors for an input image $\boldsymbol{x}$. Then the probability of input $\boldsymbol{x}$ belonging to class i can be estimated as follows:

$$p_i(\boldsymbol{z}_t) = \frac{e^{\eta \cdot \mathrm{sim}(\boldsymbol{z}_t, \boldsymbol{w}_{t,i})}}{\sum_{j \in Y_{[1:t]}} e^{\eta \cdot \mathrm{sim}(\boldsymbol{z}_t, \boldsymbol{w}_{t,j})}} \tag{1}$$

where η is a learnable scalar and $\mathrm{sim}(\cdot)$ is the cosine similarity between two vectors (Hou et al. 2019).

Training Procedure

Following the data replay strategy, our training samples S_t at each incremental step $t > 1$ consists of the incoming dataset D_t and a limited number of samples from each seen classes in memory M_{t-1}, that is, $S_t = M_{t-1} \cup D_t$. Our goal is to optimize the new model parameters $\boldsymbol{\theta}_{\mathcal{H}_t}, \boldsymbol{V}_t, \boldsymbol{U}_t$ using S_t while keeping old parameters $\boldsymbol{\theta}_{\mathcal{L}}, \boldsymbol{\theta}_{\mathcal{H}_1}, \dots, \boldsymbol{\theta}_{\mathcal{H}_{t-1}}, \boldsymbol{W}_{t-1}$ fixed to preserve previous knowledge.

We design two objectives L^{new} and L^{old} and use them *alternately* during each training step. The objective L^{new} focuses on learning discriminative features of the new incoming set of data and has four components namely, classification loss, auxiliary loss, distillation loss and margin loss:

$$L^{new} = L_{class} + \lambda_1 L_{aux} + \lambda_2 L_{dist} + \lambda_3 L_{marg} \tag{2}$$

where λ_1, λ_2 and λ_3 are hyper-parameters for balancing the losses.

On the other hand, L^{old} focuses on the under-represented old classes by assigning higher penalty for the misclassified samples and does not require auxiliary loss. We have

$$L^{old} = L_{class} + \lambda_4 L_{dist} + \lambda_5 L_{marg} \tag{3}$$

where λ_4 and λ_5 are hyper-parameters.

To prevent the model from neglecting or compromising the newly learned features when optimising L^{old}, we freeze the weights $\boldsymbol{\theta}_{\mathcal{H}_t}$ and $\boldsymbol{U}_t$. This preserves the new knowledge learned and the classifier can make use of them to refine the decision boundary between the old and new classes. Details of each loss component are given below.

Classification Loss Instead of the commonly used cross-entropy loss where serious forgetting would occur due to the imbalance between the new and old classes (Rebuffi et al. 2017), we adopt the class-balanced focal loss (Cui et al.

2019). Our classification is given by:

$$L_{class} = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim S_t} \left[-\frac{1-\beta}{1-\beta^n} \Big(1 - p_y(\boldsymbol{z}_t)\Big)^{\gamma} \log \Big(p_y(\boldsymbol{z}_t)\Big) \right] \tag{4}$$

where n is the number of samples in S_t with class label y.

By assigning higher penalty on old classes, this loss takes into account class imbalance when learning features to better differentiate old classes from the new. This avoids discarding samples valuable for learning an accurate decision boundary. Note that the hyper-parameters β and γ used for this loss in L^{old} are different from those in L^{new} so that misclassification of old class samples are penalized more under the former.

Auxiliary Loss To learn the discriminating features of the new classes, we introduce an auxiliary loss. It also uses the class-balanced focal loss but focuses only on features extracted by $\mathcal{H}_t$, that is, $\boldsymbol{h}_t$, as well as the corresponding weight vector $\boldsymbol{u}_{t,y}$ for the class y as follows:

$$L_{aux} = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim S_t} \left[-\frac{1-\beta}{1-\beta^n} \Big(1 - p_y(\boldsymbol{h}_t)\Big)^{\gamma} \log \Big(p_y(\boldsymbol{h}_t)\Big) \right] \tag{5}$$

where

$$p_y(\boldsymbol{h}_t) = \frac{e^{\mathrm{sim}(\boldsymbol{h}_t, \boldsymbol{u}_{t,y})}}{\sum_{i \in Y_{[1:t]}} e^{\mathrm{sim}(\boldsymbol{h}_t \, \boldsymbol{u}_{t,i})}} \tag{6}$$

Note that we use existing parameters in the classifier $\mathcal{F}_t$ to learn a better decision boundary in the new feature dimension, unlike previous work (Yan, Xie, and He 2021) which introduces an additional auxiliary classifier.

Distillation Loss This regularization term is designed to alleviate forgetting by transferring knowledge from the old model to the new. Since our architecture is designed to freeze previously learned features, we use logits-level distillation loss (Rebuffi et al. 2017) by minimizing Kullback–Leibler divergence (Kullback and Leibler 1951) between the probabilities of old classes $Y_{[1:t-1]}$ predicted by the model at previous step as follows:

$$L_{dist} = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim S_t} \left[\Big| Y_{[1:t-1]} \Big| \sum_{i \in Y_{[1:t-1]}} p_i(\boldsymbol{z}_{t-1}) \log \frac{p_i(\boldsymbol{z}_{t-1})}{p_i(\boldsymbol{z}_t)} \right] \tag{7}$$

We weight the loss to take into account that the need to preserve previously learned knowledge varies with the number of old classes.

Margin Loss Since the training set is dominated by new classes, the predictions may be biased towards them. To reduce the bias, we use margin ranking loss (Hou et al. 2019) such that samples extracted from memory M_{t-1} have well-separated ground truth old class from all the new classes with margin of at least m. Given a training sample $(\boldsymbol{x}, y)$, let K be the set of corresponding new classes with the k highest predicted confidence. Then the loss is defined as:

Algorithm 1: Proposed Training Procedure

```
Require: Initialized model parameters $\mathcal{L}, \mathcal{H}_1, \mathcal{F}_1$
         Number of incremental steps $T$
         Datasets at each step $\{D_1, \cdots, D_T\}$
         Number of training epochs $N$
 1: Train $\mathcal{L}, \mathcal{H}_1, \mathcal{F}_1$ with $L_{class}$ using $D_1$ for $N$ epochs
 2: for $t \leftarrow 2, \ldots, T$ do
 3:     Expand architecture with $\mathcal{H}_t$ and $\mathcal{F}_t$
 4:     Construct memory $M_{t-1}$
 5:     $S_t \leftarrow M_{t-1} \cup D_t$
 6:     repeat
 7:         Train with $L^{new}$ from Eq. (2) using $S_t$
 8:         Train with $L^{old}$ from Eq. (3) using $S_t$
 9:     until epoch = $N$
10: end for
```

$$L_{marg} = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim M_{t-1}} \left[\sum_{i \in K} \max \Big(\mathrm{sim}(\boldsymbol{z}_t, \boldsymbol{w}_{t,i}) - \mathrm{sim}(\boldsymbol{z}_t, \boldsymbol{w}_{t,y}) + m, 0 \Big) \right] \tag{8}$$

Algorithm 1 gives the details of our proposed training approach. In Line 1, D_1 is used to train the low-level feature extractor $\mathcal{L}$, high-level feature extractor $\mathcal{H}_1$, and classifier $\mathcal{F}_1$ with only classification loss as there is no old knowledge to preserve. For each subsequent step t, the architecture is expanded with a new high-level feature extractor $\mathcal{H}_t$ and a larger classifier $\mathcal{F}_t$ (Line 3). Line 4 selects old samples for data replay M_{t-1} while Line 5 merges the samples with D_t to obtain training dataset S_t. The training procedure (Lines 6-9) is repeated by first optimizing L^{new}, followed by L^{old}.

Experiments

Datasets and Settings We follow the protocol where we train the model using half of the classes and split the remaining classes evenly for training in each incremental step (Hou et al. 2019). We use the following datasets and settings in our experiments:

- CCH5000 (Kather et al. 2016): This dataset consists of histological images with each belonging to one of 8 tissue categories that represents textures in of human colorectal cancer. It has a uniform class distribution of 625 images per class, which we randomly select 20% from each for testing. We use 4 classes to train an initial model, and the remaining are split into groups of 1 and 2 classes for the two respective incremental learning settings.
- EyePACS (Kaggle and EyePacs 2015; Cuadros and Bresnick 2009): This Kaggle dataset consists of 35125 retinal images that have been graded for the severity of diabetic retinopathy (DR). There are 5 classes: no DR, mild DR, moderate DR, severe DR and proliferative DR. It has highly skewed distribution, with 73% of the images having no DR and only 2% having severe DR. We use the

first 3 classes to train an initial model and incrementally learn the remaining two classes.

- HAM10000 (Tschandl, Rosendahl, and Kittler 2018; Tschandl 2018): This dataset is a collection of 7 types of pigmented skin lesions. There are 10015 images, from which 20% is randomly selected for evaluation. The class distribution is highly uneven, with number of images per class varying from 115 to 6705. We train an initial model on 3 classes, and use the remaining 4 classes for incremental learning under two settings where 1 or 2 classes are introduced at each step respectively.

For each setting, the experiments are repeated on three random class order. All the training images are augmented with random flipping and cropping. Following previous work (Rebuffi et al. 2017), we use the herd selection strategy (Welling 2009) to select a fixed number of 20 samples per class for data replay in all experiments.

Baselines We compare our approach with the following baselines: (a) iCaRL (Rebuffi et al. 2017) alleviates forgetting using logits-level distillation loss; (b) UCIR (Hou et al. 2019) preserves previously learned knowledge by fixing the weight vectors of old classes and uses feature-level distillation loss as well as margin loss in their solution; (c) PODNet (Douillard et al. 2020) prevents forgetting with spatial-based distillation loss and uses two-stage learning to address class imbalance; (d) DER (Yan, Xie, and He 2021) uses dynamically expandable representation to handle new classes without forgetting and also two-stage learning for the class imbalance issue.

Evaluation Metrics After each incremental step t, the model is evaluated only on all the classes seen so far. We denote the accuracy of classifying samples with classes in Y_i using model trained at step t as A_t^i. The overall accuracy is computed as $Acc = \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{t}\sum_{i=1}^{t} A_t^i\right]$. In addition to the Acc metric, we also quantify the amount of forgetting of old classes as the difference between accuracy at current step and the maximum obtained before that. This is given by $Fgt = \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{t-1}\sum_{i=1}^{t-1}\max_{j=1}^{t-1}(A_j^i - A_t^i)\right]$. To determine the significance in performance improvement of our approach over the next-best-performing baseline, we run paired t-test on the metrics.

Implementation Details All methods are implemented in PyTorch (Paszke et al. 2019) and trained on NVIDIA V100 GPU with 32GB memory. We adopt ResNet-18 (He et al. 2016) as the the network backbone and cosine normalization (Hou et al. 2019) in the classifier layer. For our approach, we use the first two residual blocks as the low-level feature extractor and introduce duplicates of the remaining two residual blocks at each incremental step. Backbone of all models are initialized with weights pre-trained on ImageNet (Deng et al. 2009) and optimized using SGD optimizer with momentum value of 0.9 and weight decay of 0.0005. We use batch size of 32 for CCH5000 and HAM10000, and 128 for EyePACS. Details of the hyper-parameters, data preprocessing, data sampling can be found in our implementation code at https://github.com/EvelynChee/LO2LN.git.

Comparative Study

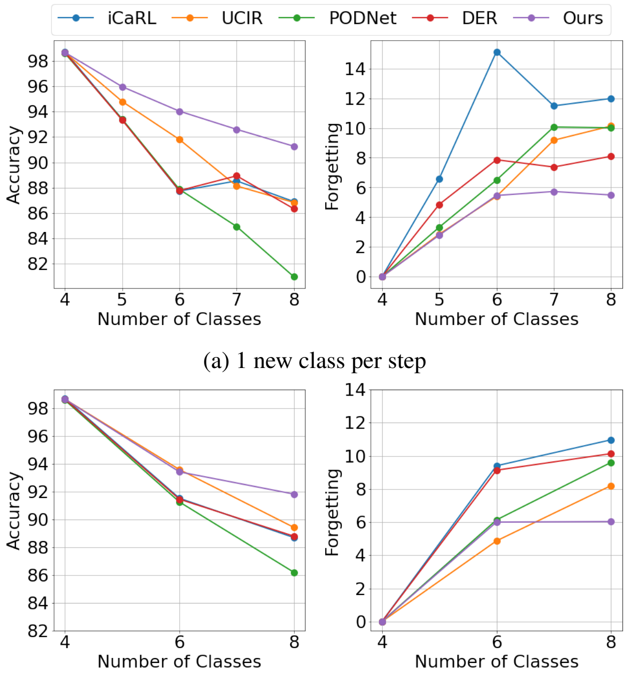
Results on CCH5000 Table 1 summarizes the results for the CCH5000 dataset. We see that our proposed approach outperforms all the baselines for both settings in terms of Acc and Fgt. Particularly, it is more advantageous under the setting of 1 new class per step, in which there is an increase of 2.5% in accuracy and a drop of 1.6% in forgetting when compared with the next-best-performing baseline. When trained incrementally with 2 new class per step, our model shows an improvement in accuracy and forgetting of 0.7% and 0.4% respectively.

Figure 4 compares the performance of all methods at each incremental step. We see that the gap between our approach and state-of-the-art methods widens as more classes are introduced. This confirms that our approach is effective in

Setting	1 new class per step		2 new classes per step	
Metric	Acc	Fgt	Acc	Fgt
iCaRL	91.1±1.8	9.0±3.3	93.0±0.2	6.8±1.0
UCIR	92.0±1.0	5.5±2.6	93.9±0.3	4.4±0.9
PODNet	89.2±0.5	6.0±1.2	92.0±0.3	5.2±0.4
DER	91.0±1.7	5.6±1.9	93.0±0.5	6.4±1.4
Ours	**94.5±0.8***	**3.9±2.0**	**94.6±0.4***	**4.0±0.8**

*Statistically significant improvement with p-value<0.05

Table 1: Results on CCH5000 over three runs. The upper bound Acc for the setting of 1 and 2 new classes per step, achieved by keeping all previous training data accessible, are 96.9% and 97.2% respectively.



(a) 1 new class per step

(b) 2 new classes per step

Figure 4: Accuracy and forgetting at each incremental step on CCH5000, averaged across three runs.

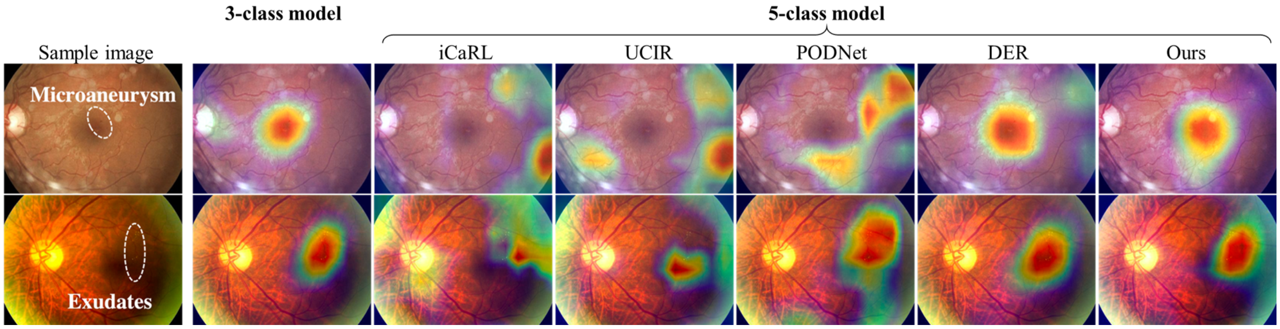


Figure 5: Activation maps on sample EyePACS images with classes seen at t=1. The second column is based on the model trained on the first 3 classes, while the remaining have been trained on all 5 classes.

Setting	1 new class per step	
Metric	Acc	Fgt
iCaRL	64.4±3.3	17.8±4.6
UCIR	70.2±7.6	15.4±11.4
PODNet	63.3±5.4	22.8±4.9
DER	58.7±9.4	30.2±6.9
Ours	**81.9± 2.5**[*]	**-0.2±0.8**

[*] Statistically significant improvement with p-value<0.1

Table 2: Results on EyePACS over three runs. The upper bound Acc for the setting of 1 new class per step, achieved by keeping all previous training data accessible, is 84.2%.

Setting	1 new class per step		2 new classes per step	
Metric	Acc	Fgt	Acc	Fgt
iCaRL	68.3±2.8	25.3±4.5	76.3±3.1	20.1±12.6
UCIR	74.1±3.1	16.3±9.1	79.1±1.4	16.8±9.5
PODNet	66.3±2.3	17.3±4.8	75.6±2.2	20.5±2.1
DER	66.9±4.5	24.7±4.8	76.2±2.8	24.8±10.9
Ours	**78.1±3.4**[*]	**10.1±3.9**	**82.0±1.3**[*]	**12.8±3.3**

[*] Statistically significant improvement with p-value<0.05

Table 3: Results on HAM10000 over three runs. The upper bound Acc for the setting of 1 and 2 new classes per step, achieved by keeping all previous training data accessible, are 89.3% and 89.1% respectively.

learning the new while not forgetting the old. Similarly, a larger improvement is seen for the smaller incremental class setting (1 new class per step) where our method outperforms the nearest baseline in terms of final accuracy by 4.4% (from 86.9% to 91.3%) and final forgetting by 2.6% (from 8.1% to 5.5%). As for the setting of 2 new classes per step, the performance gap at the final step are 3.1% (from 88.7% to 91.8%) and 2.2% (from 8.2% to 6.0%) respectively.

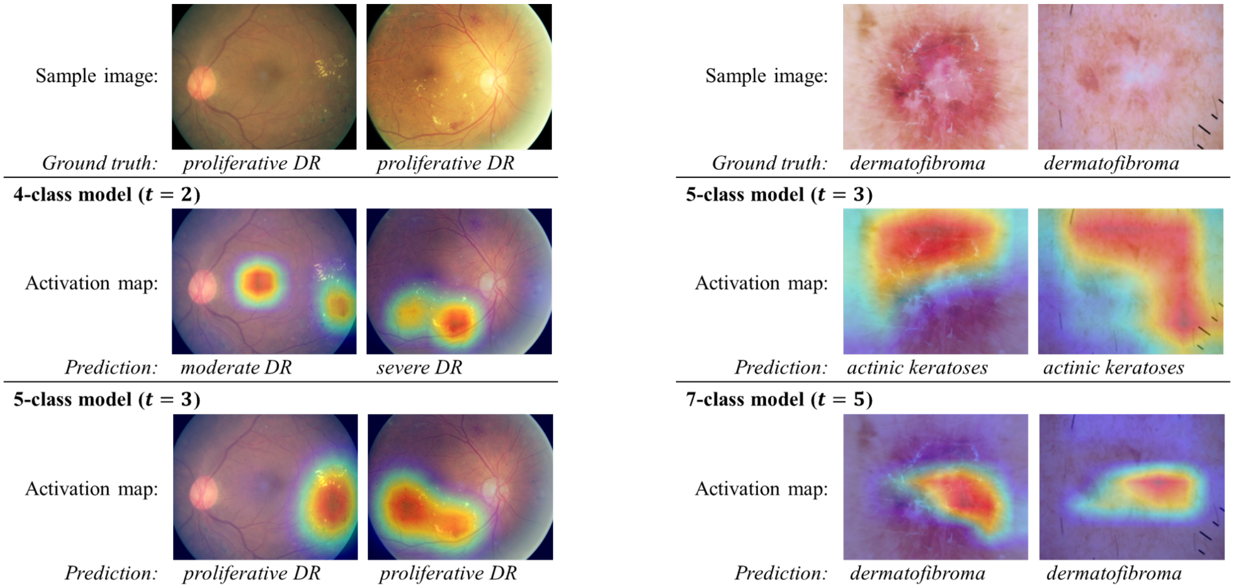
Results on EyePACS Table 2 shows the performance of the various methods on the EyePACS dataset where the number of new classes per step is 1. Our approach achieves significant performance improvement over UCIR where Acc is boosted by 11.7% and Fgt is reduced by 15.6%.

When we compare Tables 1 and 2, we see a general drop in accuracy and increased forgetting for all the baseline methods. This is because unlike CCH5000, EyePACS has a highly imbalanced class distribution, making continually learning in such skewed datasets more challenging as predictions could easily bias towards new classes that are heavily over-represented. In spite of this, our approach is able to overcome such issue with the use of alternate training objectives that focus on the under-represented old classes.

Figure 5 shows the class activation maps obtained using Grad-CAM (Selvaraju et al. 2017) for the model trained on the first three classes (3-class model) as well as those for the various methods after continual learning all the five classes (5-class model). We observe that the activation maps of DER and our proposed approach have the most overlapped regions with that of the 3-class model, indicating the effectiveness of dynamically expandable representation in preserving the old features. However, DER wrongly predicted both samples of moderate DR as proliferative DR. This is due to the two-stage learning approach used in DER, where the classifier is re-initialized and fine-tuned using a balanced dataset in the second stage. Discarding the previously learned class weight vectors corresponding to the old features leads to loss in accuracy and increased forgetting.

We also note that our approach is the only one to attain a negative forgetting, suggesting that we are able to utilize newly learned knowledge to improve the classification of previous classes. Figure 6a shows two samples that are misclassified by the 4-class model (already seen classes no DR, moderate DR, severe DR, and proliferative DR) at t=2 and their corresponding activation maps. After the class mild DR is introduced at t=3, we observe that the 5-class model focuses better on the relevant clinical symptoms (small bright exudates spots) and gives the correct predictions. This suggests that our model realizes the importance of this symptom due to its absence in milder cases and thus, updates its knowledge on old classes to achieve negative forgetting.

Results on HAM10000 We also compare the performance of the various methods on HAM10000. Table 3 shows that our approach outperforms the baselines by a wide margin for both Acc and Fgt. Again, the improvement over the next best method is more significant under the setting of 1 new class per step, where there is an increase of 4.0% in accuracy and a drop of 6.2% in forgetting.



(a) EyePACS

(b) HAM10000

Figure 6: Activation maps and predictions on sample images using our model trained at different incremental steps t.

Figure 6b shows the activation maps of two dermatofibroma samples characterized by central white patches (Zaballos et al. 2008). At t=3, the 5-class model receives and is trained with new data on this class but its activation maps for these two samples are outside the white patches, indicating that the model is not focusing on the correct features and hence misclassifies. In contrast, the 7-class model at t=5 is able to focus on the relevant regions and improves its predictions on the previously learned class.

Ablation Study

We analyze the effect of different loss components in our training objectives. Table 4 shows the results of ablation study using CCH5000 under the incremental setting of 1 new class per step. Besides the metric Acc, we also report the average accuracy on samples of classes newly learned at each incremental step t (i.e., $Acc_{new} = \frac{1}{T}\sum_{t=1}^{T} A_t^t$) and those of classes introduced at t=1 (i.e., $Acc_{old} = \frac{1}{T}\sum_{t=1}^{T} A_t^1$) since they depict the model's ability to learn new concepts and preserve old knowledge respectively.

With all the loss components incorporated, our approach achieves the highest Acc and the best balance between Acc_{new} and Acc_{old}. Removing the margin loss results in the steepest drop in Acc and Acc_{old}, indicating its role in alleviating forgetting. However, it has also resulted in the highest Acc_{new}, which demonstrates its impact on learning of new knowledge. As for L^{old}, there is a drop in Acc_{old} when it is not used, suggesting the effectiveness of our alternating objective functions in preserving old knowledge.

Aside from the loss components, we also analyze the effect of dynamically expanding the network. We train a model using the proposed objective functions but without adding a new high-level feature extractor at each incremental step. The Acc, Acc_{new} and Acc_{old} obtained are 91.9%, 96.3% and 90.6%, which corresponds to a drop of 2.6%, 0.3% and 3.4% respectively. The relatively large drop in Acc_{old} indicate its importance in preserving knowledge.

	Acc	Acc_{new}	Acc_{old}
All	**94.5±0.8**	96.6±0.9	94.0±0.7
Without L^{old}	93.8±1.1	97.9±1.0	92.6±1.1
Without L_{aux}	94.4±0.7	96.4±0.6	93.7±0.9
Without L_{dist}	93.4±0.4	96.5±0.6	92.4±0.6
Without L_{marg}	91.8±1.7	98.9±0.5	89.3±1.5

Table 4: Ablation study on effects of each loss component using CCH5000 over three runs.

Conclusion

In this paper, we proposed a class-incremental continual learning framework for the medical domain that leverages on previously learned features to acquire new knowledge. By utilizing a dynamic architecture with expanding representations, it is able to retain old features while learning new ones. We have achieved a good balance in the performance of old and new classes by alternating the training of the model using two objectives, with one focused on learning from new incoming data while the other emphasizing the under-represented old classes. Experiment results on three medical imaging datasets, including those with highly skewed distribution, have demonstrated the effectiveness of our proposed approach over state-of-the-art baselines. Future work includes investigating the need to expand the model when new classes are introduced and developing a metric to quantify the contribution of adding a feature extractor branch.

Acknowledgments

This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-GC-2019-001-2A).

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 139–154.

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019. Online Continual Learning with Maximal Interfered Retrieval. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ardila, D.; Kiraly, A. P.; Bharadwaj, S.; Choi, B.; Reicher, J. J.; Peng, L.; Tse, D.; Etemadi, M.; Ye, W.; Corrado, G.; et al. 2019. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6): 954–961.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, 233–248.

Chaudhry, A.; Gordo, A.; Dokania, P.; Torr, P.; and Lopez-Paz, D. 2021. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6993–7001.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H. S.; and Ranzato, M. 2019. On Tiny Episodic Memories in Continual Learning. arXiv:1902.10486.

Cuadros, J.; and Bresnick, G. 2009. EyePACS: an adaptable telemedicine system for diabetic retinopathy screening. *Journal of diabetes science and technology*, 3(3): 509–516.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9268–9277.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. IEEE.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, 86–102. Springer.

El-Mallawany, N.; Frazer, J.; Van Vlierberghe, P.; Ferrando, A.; Perkins, S.; Lim, M.; Chu, Y.; and Cairo, M. 2012. Pediatric T-and NK-cell lymphomas: new biologic insights and treatment strategies. *Blood Cancer Journal*, 2(4): e65–e65.

Esteva, A.; Kuprel, B.; Novoa, R. A.; Ko, J.; Swetter, S. M.; Blau, H. M.; and Thrun, S. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639): 115–118.

Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A. A.; Pritzel, A.; and Wierstra, D. 2019. Pathnet: Evolution channels gradient descent in super neural networks. arXiv:1701.08734.

Goodfellow, I. J.; Mirza, M.; Da, X.; Courville, A. C.; and Bengio, Y. 2014. An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks. In *2nd International Conference on Learning Representations (ICLR), Conference Track Proceedings*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2018. Lifelong learning via progressive distillation and retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 437–452.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 831–839.

Hung, C.-Y.; Tu, C.-H.; Wu, C.-E.; Chen, C.-H.; Chan, Y.-M.; and Chen, C.-S. 2019. Compacting, picking and growing for unforgetting continual learning. *Advances in Neural Information Processing Systems*, 32.

Kaggle; and EyePacs. 2015. Kaggle Diabetic Retinopathy Detection. https://www.kaggle.com/c/diabetic-retinopathy-detection/data. Accessed: 2022-02-20.

Kather, J. N.; Weis, C.-A.; Bianconi, F.; Melchers, S. M.; Schad, L. R.; Gaiser, T.; Marx, A.; and Z"ollner, F. G. 2016. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6: 27988.

Kim, J.-Y.; and Choi, D.-W. 2021. Split-and-bridge: Adaptable class incremental learning within a single neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8137–8145.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

Li, Z.; Zhong, C.; Wang, R.; and Zheng, W.-S. 2020. Continual learning of new diseases with dual distillation and ensemble strategy. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 169–178. Springer.

Liu, Y.; Su, Y.; Liu, A.-A.; Schiele, B.; and Sun, Q. 2020. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 12245–12254.

Mallya, A.; Davis, D.; and Lazebnik, S. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 67–82.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

McKinney, S. M.; Sieniek, M.; Godbole, V.; Godwin, J.; Antropova, N.; Ashrafian, H.; Back, T.; Chesus, M.; Corrado, G. S.; Darzi, A.; et al. 2020. International evaluation of an AI system for breast cancer screening. *Nature*, 577(7788): 89–94.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Rao, D.; Visin, F.; Rusu, A.; Pascanu, R.; Teh, Y. W.; and Hadsell, R. 2019. Continual unsupervised representation learning. *Advances in Neural Information Processing Systems*, 32.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Selvaraju, R. R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128: 336–359.

Shim, D.; Mai, Z.; Jeong, J.; Sanner, S.; Kim, H.; and Jang, J. 2021. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9630–9638.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

Tao, X.; Chang, X.; Hong, X.; Wei, X.; and Gong, Y. 2020. Topology-preserving class-incremental learning. In *European Conference on Computer Vision*, 254–270. Springer.

Ting, K. M. 2000. A comparative study of cost-sensitive boosting algorithms. In *Proceedings of the 17th International Conference on Machine Learning*. Citeseer.

Tschandl, P. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. https://doi.org/10.7910/DVN/DBW86T. Accessed: 2022-04-04.

Tschandl, P.; Rosendahl, C.; and Kittler, H. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1): 1–9.

Wang, Z.; Liu, L.; Duan, Y.; and Tao, D. 2022. Continual learning through retrieval and imagination. In *AAAI Conference on Artificial Intelligence*, volume 8.

Welling, M. 2009. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 1121–1128.

Wu, C.; Herranz, L.; Liu, X.; van de Weijer, J.; Raducanu, B.; et al. 2018. Memory replay gans: Learning to generate new categories without forgetting. *Advances in Neural Information Processing Systems*, 31.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 374–382.

Yan, S.; Xie, J.; and He, X. 2021. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3014–3023.

Yang, Y.; Cui, Z.; Xu, J.; Zhong, C.; Wang, R.; and Zheng, W.-S. 2021. Continual learning with bayesian model based on a fixed pre-trained feature extractor. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 397–406. Springer.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2017. Lifelong learning with dynamically expandable networks. arXiv:1708.01547.

Zaballos, P.; Puig, S.; Llambrich, A.; and Malvehy, J. 2008. Dermoscopy of Dermatofibromas: A Prospective Morphological Study of 412 Cases. *Archives of Dermatology*, 144(1): 75–83.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 3987–3995. PMLR.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13208–13217.