Extending Pretrained Segmentation Networks with Additional Anatomical Structures

Firat Ozdemir, Orcun Goksel
Computer-Assisted Applications in Medicine (CAiM)
ETH Zurich, Switzerland

Abstract—Comprehensive surgical planning require complex patient-specific anatomical models. For instance, functional musculoskeletal simulations necessitate all relevant orthopedic structures to be segmented, which could be performed in real-time using deep neural networks given sufficient annotated samples. Such large datasets of multiple structure annotations are costly to procure and are often unavailable in practice. Nevertheless, annotations from different studies and centers can be readily available, or become available in the future in an incremental fashion.

We propose a class-incremental segmentation framework for extending a deep network trained for some anatomical structure to yet another structure using a small incremental annotation set. Through distilling knowledge from the current state of the framework, we bypass the need for a full retraining. This is a meta-method to extend any choice of desired deep segmentation network with only a minor addition per structure, which makes it suitable for lifelong class-incremental learning and applicable also for future deep neural network architectures.

We evaluated our methods on a public knee dataset of 100 MR volumes. Through varying amount of incremental annotation ratios, we show how our proposed method can retain the previous anatomical structure segmentation performance superior to the conventional finetuning approach. In addition, our framework inherently exploits transferable knowledge from previously trained structures to incremental tasks, demonstrated by superior results compared to non-incremental training.

With the presented method, new anatomical structures can be learned without catastrophic forgetting of older structures and without extensive increase of memory and complexity. By leveraging information of similar tissue from older examples, e.g., appearance similarity across different bones, a fraction of annotations was seen to be sufficient when extending to new anatomical structures, essential in incrementally building comprehensive segmentation models for patient-specific planning.

I. INTRODUCTION

Patient-specific surgical planning can be costly due to extended resources required to model anatomical structures of the patient (e.g., in order to plan guides for osteotomy, craft/pick the right implants for replacement surgery). Furthermore, a lot of additional musculoskeletal structures (i.e., muscles, tendons, ligaments) which directly influence the surgical outcome are currently completely ignored due to lack of resources to include them in interventional planning. In the pursuit of a functional surgery planning, accurate segmentation of the anatomical structures of interest is indeed mandatory. In 2014, there were more than 1 million hip and knee replacement surgeries in total in the US, which is expected to grow even more up to 171% by 2030. Based on the records from 2000 to 2014, revision surgeries are similarly expected to increase more than 142% by 2030, summing up to almost 0.2 million [1]. Personalized medicine can help reducing the number of revision surgeries, mainly when its due to implants loosening or instability due to patient anatomy not fitting to the used generic implant(s). With the increasing life expectancy, growing population, and the major impact of orthopedic conditions on the life quality, there is an active necessity to lower the cost of personalized medicine.

Convolutional Neural Networks (CNNs) are suitable candidates for surgical planning since they can infer fast and are known to handle complex segmentation tasks. For the ultimate goal of functional surgery planning, eventually all relevant anatomical structures will be needed. Conventional CNNs require a lot of dataset with corresponding annotations. Manual annotation is costly, especially in the medical field, where experts are needed for the task. Intuitively, for similar tissue types (e.g. different bones) knowledge from one class annotations can be transfered when segmenting another.

The most straightforward way to extend a pretrained network is to continue optimization (i.e., finetuning) for the new dataset (e.g., new class labels); however, it has been shown to lead to catastrophic forgetting [2] on the previous dataset/classes. The state-of-the-art approach for class-incremental learning for classification task, iCaRL [3], exploits the concept of knowledge distillation [4] to retain old-class classification performance. Distillation [4] has been initially introduced as an approach to train simpler networks using trained complex network outputs as “soft targets”. However, authors of [3] have empirically proven the usefulness on same size networks to preserve old-class classification accuracy.

It is essential in lifelong learning to estimate/bound the expanding needs of computational power and memory footprint. In [3], an approach was proposed to keep predefined number of total samples (i.e., exemplars) among all observed classes so far. In a different domain, Active Learning for Segmentation, in [5] and [6], an alternative was shown when selecting a subset of a parent sample set for the purpose of querying “representative” samples. Instead of picking samples that are closest to a predefined number of cluster means in a latent feature space [5], it was suggested to iteratively solve a maximum set-cover [7] problem. While lifelong learning is not a new concept, to the best of our knowledge, it has not been explored for class-incremental segmentation in the imaging community.

In this work, we propose an architectural extension for
a given network which allows for incremental learning for segmentation. Although we conduct our studies on a network similar to UNet [8], our proposed extension is simple and can easily be adapted for most other segmentation architectures. We further show the significance of distilling knowledge from the previous state of the network when incrementally adapting to new data. In addition, we also demonstrate the benefit of keeping informative samples throughout a lifelong learning scenario, if possible. We have presented preliminary results of this work in [9]. Our specific contributions herein are:

1) Analysis on the larger and publicly available SKI10 dataset, targeting the knee anatomy and related surgical interventions.
2) Multiple experimental settings to assess feasibility and generalizability of our proposed framework.
3) Study of incremental set imbalance.
4) Investigating the effects of physical image resolution on our incremental learning.

In the next section, we first describe the architectural changes needed to allow for class-incremental segmentation from a typical network. Then, we explain how the knowledge from previous dataset optimization is retained through distillation while also adapting for a new dataset. Finally, we propose an approach to increase knowledge distillation through informative sample retention.

II. METHODS

In most clinical MRI screening protocols, the appearance of different musculoskeletal structures (i.e., two different bones) on similar imaging protocols are not independent. In addition, most CNN architectures are over-parametrized as this helps for an easier optimization [10]. Therefore, training separate networks for datasets of different structures which share similar contextual and/or texture content may be redundant, if not suboptimal.

A. Extending to Additional Structures while Retaining Old Information

We propose a multi-headed framework for class-incremental segmentation as a cost-effective solution to lifelong learning for segmentation. Our idea is to modify a generic segmentation network (i.e., similar to U-Net), where an additional block of several convolutional layers will be appended to the network “body” in parallel as shown in Fig. 1 whenever a class-incremental dataset is available. We call the branching by each set of such additional convolutional layers as a “head” (c.f. Fig. 1), where each head segments a specific target structure(s). While such extension is rather inexpensive in number of additional parameters to be optimized, it is sufficient to combine and alter the generic features trained from the network “body” for the specific dataset which the “head” is responsible of.

Retaining Knowledge Through Distillation. In the proposed multi-headed architecture, majority of the learned kernels will be shared for all the heads. In order to prevent catastrophic forgetting, one has to ensure that the old heads can retain their segmentation performance for the past dataset. This can be achieved through soft targets [4] generated by the current heads for a new dataset. At the time when a new dataset $D_i$ is introduced for the purpose of incremental learning, all previous heads $H_{1\in[0, ..., i-1]}$ are used for generating soft targets $t_j$ where $j \in \{0, ..., i - 1\}$ for the currently existing segmentation labels. The aim is to take the corresponding soft targets also into consideration for all weights that concern a given head (i.e., shared body and the convolutional blocks in the head) when optimizing the network weights for the incremental dataset. Hence, the network is jointly optimized for both segmenting the incremental dataset $D_i$ and producing the same prediction proposals $t_j$ for the heads prior to the last incremental step. While any objective function of choice $L_{seg}$ can be used for the new head, all other heads are meanwhile optimized for the distillation loss

$$L_{\text{dist}} = \sum_j \sum_c w_j^{(c)} t_j^{(c)} p_j(c),$$  \hspace{1cm} (1)

where $j \in \{0, ..., i - 1\}$ is the identifier for the old heads, $t_j^{(c)}$ is the soft target for class $c$ in head $j$, $w_j^{(c)}$ is the weighting scalar for the corresponding class in the head, and $p_j(c)$ is the prediction proposal for the head $j$ during the incremental training. Note that when optimizing for the anatomical structure segmented by a given head, the weights of all other heads (except for the shared body) are frozen. Total loss then becomes $L_{\text{total}} = L_{\text{seg}} + L_{\text{dist}}$. We call this approach LwFseg, a naming inspired by [11]. We propose LwFseg as a viable option for most occasions, where transfer of patient data from one research group to another is problematic due to ethic and privacy concerns; however, a trained model can be shared.

B. Keeping Informative Samples

There are growing number of datasets being made publicly available over the past years [12], [13] and one can expect to see them grow. Provided that it is available, one could benefit from such datasets prior to class-incrementing their models with their own propriety dataset. Therefore, it is important to consider the option where the dataset which was used to train...
old heads can be available at a later time in order to retain knowledge.

The most intuitive approach is to simply keep all samples from all past dataset and simply use them with the \( L_{\text{seg}} \) loss function when optimizing the corresponding head. Unfortunately, this can quickly lead to significant storage requirements and is not scalable. Furthermore, with growing amount of dataset, one also needs to increase the training time at each incremental learning stage. Therefore, it is important to pick an informed subset of the old dataset, and constrain the set; i.e., fixed set size.

Ideally, one can try to (i) find samples that are most representative of the entire set (i.e., cluster means of the representations of the dataset distribution). However, given the class imbalance within datasets, it is likely that (ii) some of such representable image clusters will be redundant for the segmentation task at hand. One possible approach to account for both of these conditions is to first pick a large batch of samples which the trained model is “confident” with, and then prune them with the aim of maximizing how well they can represent the whole dataset.

**Model Confidence Approximation.** Although conventional CNNs are deterministic during inference by nature, a recent study [14] has proposed a simple way to obtain Monte Carlo estimates from an architecture through the use of Dropout layers [15] at test time. Using \( n_{\text{MC}} \) prediction samples \( S_{\text{MC}}^j(I) \) s.t. \( j \in \{0, ..., n_{\text{MC}} - 1\} \) for a given image \( I \), one can quantify how confident the model is with the prediction in many ways; e.g., averaged voxel variance per sample. For every label, we can then define confidence as

\[
m_{\text{conf}}(I, c) = \begin{cases} -Z^{-1} \sum_x \text{var}(S_{\text{MC}}(I)), & \text{if } c \text{ exists in } I \\ -\infty, & \text{otherwise} \end{cases}
\]

where variance is computed across \( n_{\text{MC}} \) samples, and result is averaged over spatial voxels \( x \). If the class of interest \( c \) does not exist in the image \( I \), it is possible that none of Monte Carlo estimates will predict it either. Therefore, we ignore such images. For each class, \( n_{\text{cer}} \) highest \( m_{\text{conf}} \) image samples are picked for the current sample set \( A_{\text{cer},i-1} \) using \( D_{i-1} \).

**Sample Pruning.** Within the procured set \( A_{\text{cer},i-1} \), there could be two samples which are almost identical (e.g. consecutive slices) for which the trained model is equally confident. Hence, a second crucial step in order to reduce redundant sample retention is to drop samples from \( A_{\text{cer},i-1} \) which already have similar looking alternatives. Although one can quantify image similarity at the intensity level (e.g. through correlation), data-driven features of a network would be expected to better represent discriminative aspects of images of a certain dataset it has been trained on. For this purpose, we apply global average pooling (i.e., average over spatial dimensions) on the feature map at the abstraction layer in Fig. [4] as to represent any image \( I \) as a vector (i.e., image descriptor [5]). Given any similarity metric between two vectors (e.g., cosine angle), we can then quantify how similar is one image to another within \( D_{i-1} \). We then maximize set cover [7] over \( D_{i-1} \) as we iteratively populate the representative set \( A_{\text{rep},i-1} \) with \( n_{\text{rep}} \) samples picked from \( A_{\text{cer},i-1} \) using image descriptors, where \( n_{\text{rep}} \leq n_{\text{cer}} \).

During the incremental training, batches are randomly generated from either incremental dataset \( D_i \) or exemplar set(s) \( A_{\text{rep},e} \) s.t. \( e \in \{0, ..., i - 1\} \). If the batch consists of an exemplar set from dataset \( D_e \), then the corresponding head \( H_e \) and the shared body weights are updated based on the distillation loss \( L_{\text{dis}} \) computed at the prediction of head \( H_e \). Otherwise, as it is done in LwFSeg, every weight on the model is updated based on the segmentation loss \( L_{\text{seg}} \) and the cumulative distillation loss computed at the heads \( H_i \) and \( \sum_{j \in [0, ..., i - 1]} H_j \) respectively. We call the extension of LwFSeg with exemplar dataset described above as Abstraction-layer Exemplar-based Incremental Segmentation, AciSeg.

### III. Experiments and Results

**Data.** We have comparatively studied our proposed class-incremental methods on publicly available MR Dataset SKI10 MICCAI Grand Challenge [12]. The dataset consists of 100 knee volumes (10874 2D slices) collected at over 80 centers from different vendors with a pixel spacing of \( 0.4 \times 0.4 \times 1.0 \)mm. Due to varying field-of-view, digital resolution of volumes are not consistent in SKI10. In order to best utilize available computational power, we have resized all in-plane image slices to \( 224 \times 224 \)px and use this resolution during training unless stated otherwise. Reported results in the section below respect the original image resolution of \( 0.4 \times 0.4 \)mm, which we achieved through bilinear up sampling of the network logits per label, prior to final segmentation. Although majority of the images are acquired using 1.5T MR machines, some of them are acquired with 1T and 3T machine. The dataset consists of both T1- and T2-weighted images.

**Methods.** We compared our methods with the conventional approaches (i.e., models trained solely on the current or incremental dataset, hence not incremental); CurSeg (a single head model trained on current dataset) and IncSeg (a single head model trained on incremental dataset). In addition, due to the lack of a state-of-the-art, a naive baseline; finetuning has been also evaluated in the following experiments to serve as a lower bound. For the finetune method, architecturally, we apply the same multi-headed approach of LwFSeg through appending new head for the incremental data and continuing training above CurSeg. However, during the incremental training, only \( L_{\text{seg}} \) is used as the objective function for the dataset \( D_i \). Without the loss of generality, for some extended evaluations we used annotations of Femur bone vs. background as the current dataset (Cur), and Tibia bone vs. background as the incremental dataset (Inc), whereas for most metrics we also evaluated opposite direction to study generalizability per structure order.

**Experimental Scenarios.** Similar to leave-patient-out approach, the dataset has been split into “current”, “incremental”, “validation” and “test” sets at the volume level in order to prevent having slices from a volume split into multiple sets (c.f. Table [1]). While one may argue that the datasets should be
of similar size at every incremental step, in practice, necessary resources for the annotation may not be always available when expanding to new anatomical structure annotations. In addition, it is important to analyze the detrimental effects of smaller incremental dataset; e.g. to have an intuition on how much additional samples one need to expand a current segmentation model to new classes. In a desired scenario, after training with a large dataset, one should need only a small set of new anatomy annotations, leveraging appearance similarities. Hence, we experimented with 4 different incremental ratios (IRs) from current to incremental dataset as shown in Table I, where the incremental dataset size is logarithmically scaled. For example, given 70 volumes for training, as one goes from balanced (35/35 ⇒ 100%, i.e., IR100) to extremely imbalanced (1/69 = 1.44% ≈ 1%, i.e., IR01).

A. Implementation Details

The proposed modified U-Net architecture is developed for 2D images with single channel, where number of filters at first convolutional layer \( n_{\text{fil}} = 32 \). The rest of the layers follow the same logic of UNet, where number of filters are defined as \( 2^n_{\text{fil}} \) where \( l \) is the level of spatial coarsening. Note that all heads consist of 2 convolutional layers with each having \( n_{\text{fil}} \) filters. All convolutional layers are batch normalized prior to ReLU activation and consist of \( 3 \times 3 \) kernel, with the exception of final prediction layer, which has a \( 1 \times 1 \) convolutional kernel. Spatial dropout layers with rate of 0.5 were utilized as shown with red arrows in Fig. 1. Note that additional dropout layers were used along the spatial upsampling path prior to deconvolution filters. Each conducted experiment is set to have 4 images per batch. Every method was trained for fixed number of steps \( n_{\text{stp}} = 20000 \) (i.e., for \( n_{\text{stp}} \) batches) using Adam optimizer with learning rate 0.001. We used inverse frequency weighted cross-entropy loss for \( L_{\text{seg}} \). Similarly, class weights \( w \) in \( L_{\text{dis}} \) are computed from the training set of the corresponding dataset as the inverse label frequency. For incremental methods (finetuning, LwFSeg, and AeiSeg), additional \( n_{\text{stp}} \) steps of training were conducted following the best validation set state; i.e., CurSeg. Batches were randomly scaled and horizontally flipped during training for the purpose of data augmentation.

In order to prevent overfitting to training set, model Dice score on a separate validation set was used to determine the state of a model to be used on the test set. For the incremental methods, we picked the model state which maximizes the average Dice score from both current and incremental classes on the validation set. Note that for the finetuning method, this resulted in using the model state at training step as early as \(<100\) due to catastrophic forgetting of the current dataset class knowledge. For AeiSeg, we empirically picked \( n_{\text{MC}} = 29 \), \( n_{\text{cer}} = 1000 \) and \( n_{\text{ cep}} = 100 \).

B. Results

We used Dice coefficient score and mean surface distance (MSD) for comparing different methods and IRs. Fig. 2 shows the distribution of segmentation performances for each test volume over 2 holdout sets (i.e., 50 volumes), where the median, the lower and upper 25%iles as well as the outliers can be seen. In Table II we show the quantitative results averaged over two holdout sets for all compared methods. For suboptimal method and IR combinations, some of network segmentation outputs did not overlap with the gold standard at all (i.e., Dice = 0.0%). Furthermore, some segmentation outputs did not even contain any foreground labels (i.e., MSD = ∞). We have excluded such volumes from the average scores in Table II where the number of missed volumes are given in parentheses, when non-zero.

To increase the statistical significance of the two extreme IR scenarios (IR100 and IR01), we ran additional 3 holdout experiments each. We show the distribution of the results from 25 × 5 = 125 volumes in Fig. 3 where the failed cases can be observed as the outliers in the Dice metric. The averaged metrics are shown in Table III omitting the volumes with failed segmentations given in parentheses.

One can argue that respecting the physical resolution is crucial for capturing the correct distribution of the dataset when learning convolutional kernels. Although we have augmented the resized training set with random scaling which could remedy for the wrongfully reduced image sizes during our pre-processing step, the question still stands and is not well explored in the literature. Hence, we create an experiment setup where we extract patches of size \( 224 \times 224 \text{px} \) from the original volumes and train models with these patches for the dataset incremental ratios (IR100 & IR01) for a holdout set. We present the averaged scores in Table IV and distributions in Fig. 3.

To show independence from the order of structures, we have repeated the experiments for current dataset annotations of Tibia bone vs. background (Cur* ← Tibia) and incremental dataset annotations of Femur bone vs. background (Inc* ← Femur) (c.f. Table V).

IV. DISCUSSIONS

Although the test set from 2 holdout sets were different, it is important to point out that all experimented data IRs had the same test volumes, hence quantitative numbers across different ratios and models for the same class are comparable in Table I. Looking at the baseline incremental model finetuning, one can point out that performance on the incremental class is always inferior to the conventional method IncSeg. This is simply because throughout the whole training of \( n_{\text{stp}} \) steps, the average validation performance for current and incremental classes never got better after a few incremental steps of

| #Volumes | IR100 | IR17 | IR04 | IR01 |
|----------|-------|------|------|------|
| Current (Cur) | 35 | 60 | 25 | 1 |
| Incremental (Inc) | 35 | 10 | 5 | 5 |
| Validation | 5 | 5 | 5 | 5 |
| Test | 25 | 25 | 25 | 25 |
training. Hence, finetune test results show minimal signs of fitting into Inc dataset.

Both of our proposed class-incremental methods (i.e., LwfSeg and AeiSeg) performed superior to the other compared methods in all data imbalance scenarios for the incremental class except for IR01 in Table II. However, this is due to 11 test volumes being omitted from results from IncSeg as opposed to 1 test volume from both LwfSeg and AeiSeg.

AeiSeg method can tolerate domain shift better than LwfSeg. Note that as the data imbalance increases (i.e., from IR100 to IR01), incremental methods (particularly LwfSeg) tend to perform worse in the current class (Cur). This phenomenon can be attributed to the fact that SK110 dataset contains acquisitions from a wide variety of vendors and imaging sequences, leading to particular holdout sets to have Inc data differ substantially from the Cur dataset distribution. Without exemplar data, LwfSeg struggles with the domain-shift between Cur and Inc dataset.

We observed the significance of having full field-of-view for incremental methods for small incremental datasets. In Fig. 4, the results between resized and patched images are seen comparable for Cur dataset in both balanced and imbalanced

---

**Table II:** Average Dice and MSD metrics for different incremental ratios (IRs) for the experiments in Fig. 2. Number of volumes omitted from the average are displayed in parentheses. Incremental method results superior to conventional models are shown in bold.

| Method  | IR100 | IR17 | IR04 | IR01 |
|---------|-------|------|------|------|
| CurSeg  | 96.0  | 95.5 | 96.6 | 95.2 |
| IncSeg  |   -   | 95.3 |  87.2|   -  |
| finetune| 84.0  | 83.7 | 74.1 | 58.7 |
| LwfSeg  | 95.6  | 93.5 | 88.2 | 83.7 |
| AeiSeg  | 95.8  | 95.4 | 93.7 | 92.5 |
| Mean Surface Distance [mm] |   |     |     |     |
| CurSeg  | 0.80  | 0.65 | 1.06 | 1.06 |
| IncSeg  |   -   | 2.04 |   -  | 7.85 |
| finetune| 4.31  |   -  | 8.27 | 16.97|
| LwfSeg  | 0.87  | 1.45 | 2.79 | 8.45 |
| AeiSeg  | 1.29  | 0.85 | 1.70 | 5.77 |

**Table III:** Average Dice and MSD metrics for the experiments in Fig. 3. Number of omitted volumes are displayed in parentheses. Incremental methods superior to the conventional approaches are shown in bold.

| Method  | IR100 | IR01 |
|---------|-------|------|
| CurSeg  | 95.7  | 0.98 |
| IncSeg  | 95.7  | 60.9 |
| finetune| 71.8  | 56.8 |
| LwfSeg  | 95.5  | 36.9 |
| AeiSeg  | 95.8  | 73.3 |

---

**Fig. 2:** Comparison between the conventional and incremental methods for 2 holdout sets for the proposed IRs (cf. Table I) for Cur ← Femur and Inc ← Tibia.
Fig. 3: Comparison between the conventional and incremental methods for 5 holdout sets for 2 extreme IRs.

Fig. 4: Comparison between resized image patches and patches extracted at original physical resolution of 0.4 mm, for a single holdout set.

TABLE IV: Average Dice and MSD metrics for the experiments in Fig. 4. Dice score differences between resized and patches above 10% are marked in bold.

| Method      | Dice Score [%] | Mean Surface Distance [mm] |
|-------------|----------------|----------------------------|
|             | IR100 | IR01 | IR100 | IR01 | IR100 | IR01 |
| CurSeg      |        |      |       |      |       |      |
| Resized     | 0.96  | 0.97 | 0.75  | 0.64 | -     | -    |
| Patches     | 0.97  | 0.97 | 1.07  | 1.27 | -     | -    |
| IncSeg      |        |      |       |      |       |      |
| Resized     | -     | 0.95 | 0.66  | 1.26 | -     | -    |
| Patches     | -     | 0.95 | 1.07  | 2.13 | -     | 5.16 |
| LwfSeg      |        |      |       |      |       |      |
| Resized     | 0.96  | 0.96 | 0.82  | 0.64 | 2.05  | 3.07 |
| Patches     | 0.96  | 0.96 | 0.86  | 0.71 | 1.77  | 2.93 |
| AeiSeg      |        |      |       |      |       |      |
| Resized     | 0.96  | 0.97 | 0.94  | 0.71 | 0.80  | 1.00 |
| Patches     | 0.94  | 0.97 | 0.94  | 0.78 | 1.33  | 1.74 |

scenarios (IR100 & IR01). However, proposed incremental methods with resized images outperform their patch-based counterparts in the Inc data when there is an extreme imbalance ratio as seen in Table IV.

We showed the robustness of our methods to the order of structures. In Table V one can see that the behavior of the compared methods with respect to different amount of current vs. incremental dataset imbalance do not change when anatomical structures are picked in the reverse order. We have retrospectively noticed that the 10 out of 11 fail cases of LwfSeg come from the same holdout set. This can be attributed to the incremental dataset of a single volume (IR01) heavily mismatching the distribution of the test set.

Considering the number of convolutional layers in a “head”, we also experimented with narrower alternatives; i.e., a convolutional layer of kernel size $1 \times 1$ at the end of the shared body for each dataset. This has proven to be insufficient for the segmentation task. Empirically, we found that heads with 2 layers of convolutional layers with each having $n_{fil}$ filters to be satisfactory for the experimented dataset, corroborating our earlier results in [9] on a different dataset of different imaging sequences.

Memory footprint is an essential concern in our framework, since it implements a solution for lifelong learning In our TensorFlow implementation, each new head requires an additional 73 KB for parameter storage, which is the only additional
| Method | Dice Score [%] | Mean Surface Distance [mm] |
|--------|----------------|---------------------------|
|        | IR100          | IR01                      | IR100         | IR01                        |
| Cur*   | 95.2           | 95.8                      | 0.04          | 0.79                        |
| Inc*   | 96.1           | 58.9                      | 0.72          | 8.08 (10)                   |
| LwfSeg | 60.8 (1)       | 91.4 (11)                 | 0.71 (1)      | 1.60 (11)                   |
|        | IR100          | 95.3                      | 0.90          | 6.06 (11)                   |

footprint for LwfSeg. For AeiSeg, an additional ≤ 21 MB was required to store the exemplar set of each class assuming nrep = 100.

V. CONCLUSIONS

We have proposed a segmentation framework for lifelong learning with class-incremental annotations. We propose LwfSeg for knowledge distillation to retain segmentation performance of previously trained class(es) when expanding a deep neural network to new class(es). Thanks to knowledge transfer from pretrained classes through shared body weights, we observe that the segmentation performance of the incremental classes may even exceed conventional methods, especially when the incremental annotations are only a few. We also propose an extension, AeiSeg, for successfully maintaining performance of previously trained classes despite the domain shift caused by the incremental dataset. We have shown that our proposed methods outperform the conventional methods trained on the incremental classes in any incremental dataset ratio, indicating the successful knowledge transfer from previous to incremental dataset. We have furthermore observed that for some increment ratios (e.g., IR17 in Table II and IR100 in Table III), our methods improve the segmentations of even the previously-trained class/structure (Cur). This is indeed an exciting observation, showing the knowledge gained from the incremental dataset can also further improve the already well-trained class performance.

ACKNOWLEDGMENT

This work was funded by the Swiss National Science Foundation and a Highly Specialized Medicine grant (HSM2) of the Canton of Zurich.

REFERENCES

[1] M. Sloan and N. P. Sheth, “Projected volume of primary and revision total joint arthroplasty in the united states, 2030-2060,” Tech. Rep. 16, Annual Meeting of the American Academy of Orthopaedic Surgeons, New Orleans, Louisiana, 03 2018.
[2] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” vol. 24 of Psychology of Learning and Motivation, pp. 109 – 165, Academic Press, 1989.
[3] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental classifier and representation learning,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5533–5542, July 2017.
[4] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in NIPS Deep Learning and Representation Learning Workshop, 2015.
[5] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, “Suggestive annotation: A deep active learning framework for biomedical image segmentation,” in MICCAI, pp. 399–407, 2017.
[6] F. Ozdemir, Z. Peng, C. Tanner, P. Fuernstahl, and O. Goksel, “Active learning for segmentation by optimizing content information for maximal entropy,” in MICCAI DLMIA, pp. 183–191, 2018.
[7] D. S. Hochbaum, “Approximation algorithms for NP-hard problems,” pp. 94–143, PWS Publishing Co., 1997.
[8] C. F Baumgartner, L. M. Koch, M. Pollefeys, and E. Konukoglu, “An exploration of 2D and 3D deep learning techniques for cardiac MR image segmentation,” in Statistical Atlases and Computational Models of the Heart, ACDC and MMWHS Challenges, pp. 111–119, 2018.
[9] F. Ozdemir, P. Fuernstahl, and O. Goksel, “Learn the new, keep the old: Extending pretrained models with new anatomy and images,” in MICCAI, pp. 361–369, 2018.
[10] L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou, “Empirical analysis of the hessian of over-parametrized neural networks,” 2018.
[11] Z. Li and D. Hoiem, “Learning without forgetting,” in ECCV, pp. 614–629, 2016.
[12] T. Heimann, B. J. Morrison, M. A. Styner, M. Niethammer, and S. Warfield, “Segmentation of knee images: a grand challenge,” in MICCAI Workshop on Medical Image Analysis for the Clinic, pp. 207–214, 2010.
[13] O. Goksel, O. A. Jimenez-del Toro, A. Foncubierta-Rodriguez, and H. Miller, “Overview of the visceral challenge at isbi 2015,” in IEEE International Symposium on Biomedical Imaging (ISBI), 2015.
[14] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in International Conference on Machine Learning (ICML), pp. 1050–1059, 2016.
[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.