ProX: A REVERSED ONCE-FOR-ALL NETWORK TRAINING PARADIGM FOR EFFICIENT EDGE MODELS TRAINING IN MEDICAL IMAGING

Shin Wei Lim[#*] *Chee Seng Chan*[#†] *Erma Rahayu Mohd Faizal*[#] *Kok Howg Ewe*[‡]

[#]Faculty of Comp. Sci. and Info. Tech., Universiti Malaya, Kuala Lumpur, Malaysia
[‡]Software Validation & Engineering, Intel Corporation

ABSTRACT

The usage of edge models in medical field has a huge impact on promoting the accessibility of real-time medical services in the under-developed regions. However, the handling of latency-accuracy trade-off to produce such an edge model is very challenging. Although the recent *Once-For-All* (OFA) network is able to directly produce a set of sub-network designs with *Progressive Shrinking (PS)* algorithm, it still suffers from training resource and time inefficiency downfall. In this paper, we propose a new OFA training algorithm, namely the *Progressive Expansion (ProX)*. Empirically, we showed that the proposed paradigm can reduce training time up to 68%; while still able to produce sub-networks that have either similar or better accuracy compared to those trained with *OFA-PS* in ROCT (classification), BRATS and Hippocampus (3D-segmentation) public medical datasets.

Index Terms— Medical Image Analysis, Edge A.I.

1. INTRODUCTION

Digital inequality issue in the under-privileged communities is an on-going problem. For instance, in the rural area, it is always lack with telecommunication accessibility. At the same time, for the past few years, we can notice that there is an increase of deploying deep learning solutions via cloud services especially in the medical domain with success stories. However, all these success stories came with a heavy price tag - high computational cost and require advanced technologies. Thus, research in lowering the model complexity, especially in edge models is important to make the solutions available to all. That is to say, we need the edge models to be able to work in those low bandwidth or limited network coverage environments, e.g., health centers in rural area or portable medical devices in a moving vehicle.

Recently, the *Once-For-All* (OFA) Network [1] that based on *Neural Architecture Search* (NAS) framework showed that it is able to train a trillion of Convolutional Neural Network (CNN) subnetworks from a mother network at once. Then,

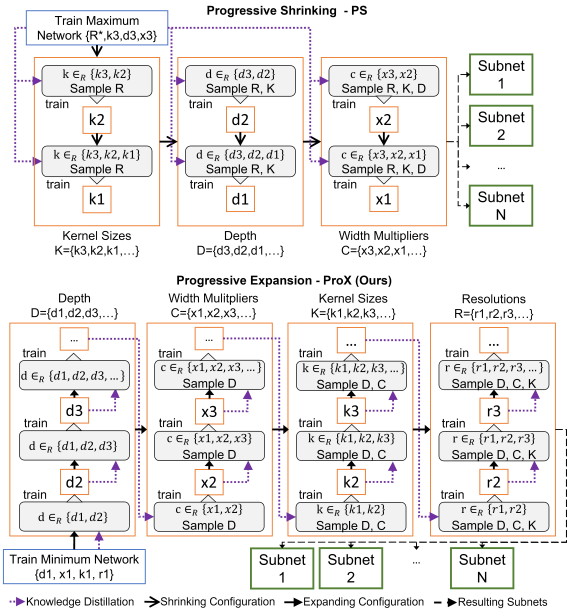

Fig. 1. The *Progressive Expansion (ProX)* (below, our proposed method) and the *Progressive Shrinking (PS)* (above, the original method). *R = {r3,r2,r1}.

it employed Automated Machine Learning (AutoML) algorithms to search for a best-fit (latency-accuracy trade-off) configuration based on different deployment scenarios (CPU latency). Technically, the mother network is trained with *Progressive Shrinking (PS)* algorithm where the training starts from the largest configuration and gradually shrinking the network across 4 dimensions (i.e. input resolution, kernel size, depth and width). Although the searching complexity is reduced under this OFA framework, the training complexity with the *PS* algorithm is still a major concern when preparing the mother network. For example, one of the most obvious drawback of this shrinking paradigm is - (i) **not resource friendly** and (ii) requires **a longer training time**. This is because the training of the network must start from the highest (best) configurations, which in turn requires the ultimate training resource planning i.e., GPUs. Besides that, training with the largest network at start will also expose the model to the risk of **over-parameterizing** and subsequently overfits

[*]Author performed the work while at Intel Corp.
[†]Corresponding author - cs.chan@um.edu.my

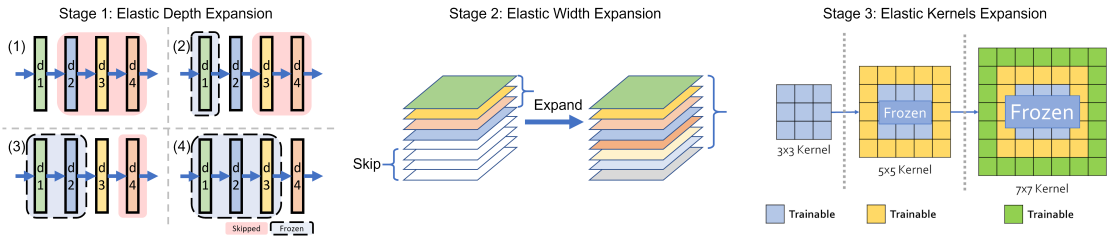

Fig. 2. (Stage 1): Elastic Depth Expansion; (Stage 2): Elastic Width Expansion; (Stage 3): Elastic Kernels Expansion

the models, especially for medical imaging datasets which are usually smaller in size and prone to noises [2, 3]. Not the least, the mother network trained with *PS* has **limited scalability** because the candidate architecture configurations are bounded to $[arch_{min}, arch_{max}]$. That is, if the network needs to be expanded (when more training resources or datasets are acquired), the network will have to repeat the training again.

In this paper, we propose a novel training algorithm for OFA, namely the *Progressive Expansion (ProX)* as an alternative to the *PS* training method as shown in Fig. 1. In brief, our solution is a reverse paradigm to Cai et al. [1] where we train the OFA mother-net from the smallest configurations first, and then gradually expand it, as oppose to training the largest network first and shrink the configurations. Empirically, we show that our proposed training paradigm is (i) shorter in training time (Fig. 3) while maintaining the model accuracy (Fig. 4); and (ii) possible to prevent over-parameterization on smaller size and lower dimension medical datasets (Fig. 6).

2. RELATED WORK

In order to reduce the model complexity, solutions such as network pruning [4–7], bit quantization [8–10], and NAS [11, 12] have been employed. NAS methods are more flexible in comparison to the other two because NAS can search for any possible configurations within the search space; while the other two involve only in decreasing the parameters of the network. However, the search time for conventional NAS is rather slow due to the train-search process coupling, which is inefficient when deploying models to various devices. The recent OFA Network was introduced to tackle the time constraint by decoupling the train-search process. With OFA, subnetworks can be sampled directly from the mother network without re-training. However, the *PS* algorithm in OFA requires a higher training resource, a longer training time and prone to overparameterization. There are several works extending the OFA techniques or using similar training styles such as the *CompOFA* [13] and *3D-NAS* [11], but these methods are still bounded to training the largest configuration network first, which have similar drawbacks as aforementioned.

In contrast, this paper is motivated by several expanding paradigm works. For example, *EfficientNets* [14] was one of the examples of growing the network using compound scaling. Model expansion are also used in *Lifelong Learning* studies such as the *Progressive Neural Network* [15] and the *Dynamically Expandable Networks* [16]. In ProX, we follow this paradigm and extended the use of expansion concept into training the OFA mother network.

3. *ProX* TRAINING PARADIGM

This section details the proposed *Progressive Expansion (ProX)* training paradigm as shown in Fig. 1. In brief, *ProX* is very similar to *PS*, but refined in terms of order and inner operations as described next. Herein, each stage is referred to depth, width, kernel and resolution respectively as illustrated in Fig. 2, while each phase is referred to the inner operations in the respective stage.

3.1. Elastic Depth Expansion

First, we treat the convolution layers in the OFA network as a section of depths, e.g., if the OFA mother-net is initialized with 5 sequentially connected sections where each section contains 4 layers, then the mother-net will have a minimum 5 to a maximum 20 layers of convolution blocks. Fig. 2 (Stage 1) shows the depth expansion steps within a depth section. The depth expansion step is done simultaneously across all sections in the mother network. Technically, we adopted *Forward Thinking* depth training [17, 18], but redesigned it from training a single network originally to training a OFA mother network. In order to do that, initially, we will train the OFA mother-net with the smallest number of layers and all the remaining layers are skipped. That is to say, the training starts with depth, $d = 1$ for all the depth sections. Next (phase 2), the depth candidate set will be expanded to $\{d1, d2\}$. In this phase, for instance, if $d2$ is sampled, the weights of $d1$ will be frozen to reduce the training computation cost. The process will iterate until the final layer is trained. Subnetworks with different depth configurations can now be directly sampled from the mother network.

3.2. Elastic Width Expansion

In the previous stage, the mother net was trained initially with an initial width c and the width multiplier candidate set $\{x1\}$.

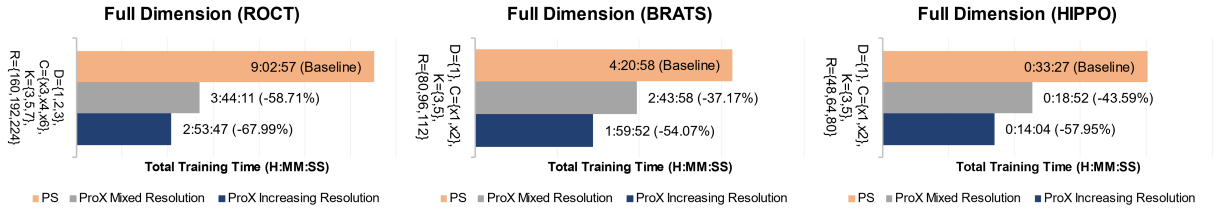

Fig. 3. Comparison of the training time for full OFA mother network trainings in 3 different datasets.

In other words, the network with the smallest width $c \times 1$ was trained and the unused layers were skipped previously. Following that, in the second stage, we will train this depth expanded OFA mother net with the expanded width multiplier candidate set $\{x1, x2\}$. If $x2$ is randomly sampled in an iteration, all layers within first $c \times 2$ channels will be trained as shown in Fig. 2 (Stage 2). In contrast to the width manipulation operation in *SlimmableNet* [19], we do not iterate all the width candidates at once, instead we gradually expand the width candidate set from the minimum width and randomly sample a width from the candidate set on every iteration.

3.3. Elastic Kernels Expansion

In *ProX*, a larger kernel is formed by stacking smaller kernels inside it as depicted in Fig. 2 (Stage 3). Our idea is a small kernel can be transformed into a larger kernel by padding the trainable weights around it as to [20]. That is to say, technically, in this third stage, with the network already trained with the smallest possible kernel size (usually 3x3 in the previous Elastic Depth and Width Expansion stage), so for the first expansion, the kernel candidate set is expanded to $\{3\text{x}3, 5\text{x}5\}$ where the 3x3 kernel will be padded with trainable zero tensors to make it a 5x5 kernel. To focus on reusing and preserving the content of the smaller kernels, the weights of the inner kernels (3x3) will be frozen during the training. In other words, only the outer part of the kernel is trainable. The same operation will be repeated for a 7x7 or larger kernels, if any. After training is completed, all kernels with varying sizes, expanding from the innermost to the outermost, can be directly sampled for inference.

3.4. Elastic Resolution Expansion

Generally, input image with a higher resolution or more pixel count needs more Floating Point Operations (FLOPs) to complete the convolution. For instance, convolving a 32x32 and 64x64 image will require 9,000 FLOPs and 38,440 FLOPs respectively with a 3x3 kernels. In *ProX*, with the OFA network already trained with the smallest image size possible since Stage 1, e.g., 160x160 in this paper but not limited to. For this final stage, to support a higher resolution like 192x192, we could expand the resolution candidate set to $\{160, 192\}$. The dataset generator will resize and generate images with randomly chosen resolution from the candidate set. The operations is same as to dealing with 3D volumes.

3.5. *ProX* OFA Model

In summary, the training algorithm of *ProX* is re-arranged in the order of depth, width, kernels and resolution sequentially as shown in Fig. 1. It starts with the network initialized with the lowest[1] depth, width, kernels and resolution. At then, *ProX* expands the weights of the network in a **forward** (expanding depth), followed by **upward** (expanding channel width on each depth) and **outward** (expanding kernels on each channel) logic, respectively and finally increasing the image resolution. Beside that, our training will also be aided with *Knowledge Distillation* [21], where each model from the previous phase is a teacher network to partially supervise the learning of the expanded dimensions. This is opposed to *PS* where the teacher network is a maximum network.

4. EXPERIMENT SETUP AND RESULTS

To test the effectiveness of our proposed method, we have chosen the two most common tasks in medical imaging - (i) medical image classification and (ii) medical image segmentation tasks. For the classification task, we implemented the OFA network on the *MobileNet* architecture [22]; while for image segmentation task we implemented the OFA network on *3D-UNet* [23]. The former experiment is conducted on the *Retinal Optical Coherence Tomography* (ROCT) [24] dataset, while the latter experiment is conducted on *3D Brain Tumor Segmentation* (BRATS) and *3D Hippocampus* head-and-body segmentation dataset from the *Medical Decathlon Dataset* [25]. In our experiments, both PE and *ProX* methods were trained using the same number of epochs for a fair comparison. All models were trained and tested on 2*NVIDIA Tesla P100 PCIe 16 GB GPUs with Adam optimizer. Cross entropy loss was used for the classification task, meanwhile Dice loss was used for the segmentation tasks. For the knowledge distillation, we used the mean squared loss between the direct outputs of the teacher and student nets.

[1]The *lowest* configuration here is for training a full OFA *ProX* model, when setting up experiments for separate dimension as in Fig. 5 and 6, the minimum configurations are different with varying constant variables

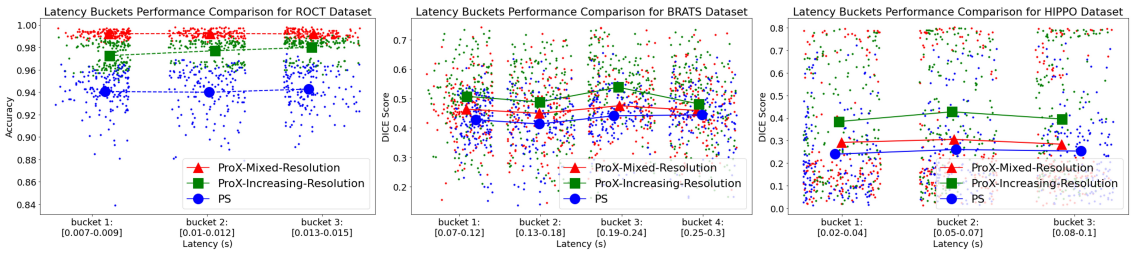

Fig. 4. Subnetworks sampling grouped by latency buckets for ROCT (left), BRATS (middle) and Hippocampus (right) datasets.

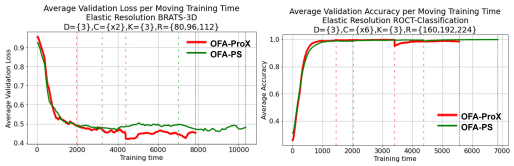

Fig. 5. *ProX* resolution converged better during training in segmentation (BRATS) while *PS* resolution is better in classification (ROCT).

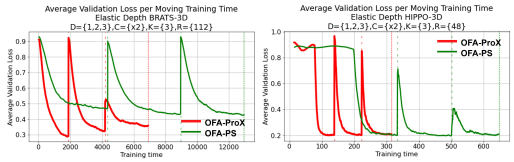

Fig. 6. *ProX* can save training time and resource by allowing early inspect of overparameterization.

4.1. Training Time vs. Model Accuracy

Fig. 3 shows that *ProX* manages to reduce the training time, ranging from 37% to 68% compared to *PS*. For instance in the ROCT datasets, *ProX* (with and without mixing resolution) can reduce the training time from 9 hours to just roughly 3.75 and 3 hours respectively, while similar improvements can be observed in other datasets. In terms of model accuracy, to better represent the actual distribution and scoring performance boundary of the subnetworks at different latencies, we created a bucket of latency groups with different latency ranges as comparison. For each training method, we sample 150 subnetworks with latencies correspond to the buckets and plot the accuracy/Dice score performance (Fig. 4). It is observed that with a much shorter training time, *ProX* can produce subnetworks with better accuracy than *PS*.

4.1.1. Mixed Resolution Ablation

Despite observing a faster training loss convergence with *ProX* training, *PS* somehow converged better in the classification tasks (Fig. 5). This result may indicates that randomizing the resolution settings works better for classification tasks. To further understand this, we conducted another set of OFA model besides *PS* and *ProX* in full training to further analyse the impact of using mixed resolution. With this, we introduced *ProX mixed-resolution (MR)* and *ProX increasing-resolution (IR)*. We sampled a total of 11k subnetworks from the OFA-ROCT-MobileNet and 5k each from the OFA-BRATS-3D-UNet and OFA-Hippocampus-3D-UNet, respectively. For ROCT dataset, we observe a mean accuracy scores of 94.16% (*PS*), **99.22% *(ProX-MR)*** and 97.72% (*ProX-IR*). For BRATS datasets, we have mean Dice scores of 0.3792 (*PS*), 0.4235 (*ProX-MR*) and **0.4693 *(ProX-IR)***. Finally, for Hippocampus dataset, we have mean Dice scores of 0.2464 (*PS*), 0.2866 (*ProX-MR*) and **0.4083 *(ProX-IR)***. Together with the latency buckets, these shows that generally *ProX-MR* produces better classifier CNN subnets, while *ProX-IR* is better at segmentation CNN.

4.2. Over-parameterization

Fig. 6 shows the possibility of *ProX* in overparameterization prevention. Each spike in the training graph represents an expand/shrinking process. Visually, it can be seen that as the model stopped to improve when $d > 1$, the *ProX* training can be terminated immediately on depth $d = 1$, avoiding in producing redundancy depth and excessive parameters. As such, together with *ProX* and the help of training monitoring tools like *TensorBoard*, deciding when to terminate the expansion under time and GPU memory concern is possible.

5. CONCLUSIONS

This paper proposes a reversed OFA Network training algorithm, the *Progressive Expansion (ProX)* for medical imaging tasks. *ProX* can achieve up to 68% training time reduction compared to *Progressive Shrinking*, while producing higher quality subnetworks. In future work, we hope to explore the usability of *ProX* on other domains and extend the work to support the open-source Intel OpenVINO™ platform.

6. ACKNOWLEDGEMENT

This research is supported by MyIndustry AI Scholarship Programme, jointly funded by the Intel Corp, Universiti Malaya and Malaysia Digital Economy Corporation (MDEC).

7. REFERENCES

[1] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," in *ICLR*, 2020.

[2] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *ICDMW*, 2016, pp. 241–246.

[3] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, "Deep learning techniques for medical image segmentation: achievements and challenges," *Journal of Digital Imaging*, vol. 32, no. 4, pp. 582–596, 2019.

[4] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[5] J. H. Tan, C. S. Chan, and J. H. Chuah, "End-to-end supermask pruning: Learning to prune image captioning models," *Pattern Recognition*, vol. 122, pp. 108366, 2022.

[6] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *ICLR*, 2019.

[7] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka, "Fuzzy qualitative deep compression network," *Neurocomputing*, vol. 251, pp. 1–15, 2017.

[8] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X. s. Hua, "Quantization networks," in *CVPR*, 2019, pp. 7300–7308.

[9] C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained ternary quantization," in *ICLR*, 2017.

[10] J. H. Tan, C. S. Chan, and J. H. Chuah, "Comic: Toward a compact image captioning model with attention," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2686–2696, 2019.

[11] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *ECCV*, 2020, pp. 685–702.

[12] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy, "Progressive neural architecture search," in *ECCV*, 2018, pp. 19–34.

[13] M. Sahni, S. Varshini, A. Khare, and A. Tumanov, "Compofa: Compound once-for-all networks for faster multi-platform deployment," in *ICLR*, 2021.

[14] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019, pp. 6105–6114.

[15] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[16] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *ICLR*, 2018.

[17] C. Hettinger, T. Christensen, B. Ehlert, J. Humpherys, T. Jarvis, and S. Wade, "Forward thinking: Building and training neural networks one layer at a time," *arXiv preprint arXiv:1706.02480*, 2017.

[18] X. Xiao, T. B. Mudiyanselage, C. Ji, J. Hu, and Y. Pan, "Fast deep learning training through intelligently freezing layers," in *iThings and GreenCom and CPSCom and SmartData*, 2019, pp. 1225–1232.

[19] J. Yu, L. Yang, N. Xu, J. Yang, and T. Huang, "Slimmable neural networks," in *ICLR*, 2019.

[20] S. Han, Z. Meng, Z. Li, J. O. Reilly, J. Cai, X. Wang, and Y. Tong, "Optimizing filter size in convolutional neural networks for facial action unit recognition," in *CVPR*, 2018, pp. 5070–5078.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[23] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2016, pp. 424–432.

[24] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, and F. Yan, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131. e9, 2018.

[25] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, and B. Menze, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019.