Federated Whole Prostate Segmentation in MRI with Personalized Neural Architectures

Holger R. Roth[0000-0002-3662-8743], Dong Yang[0000-0002-5031-4337],
Wenqi Li[0000-0003-1081-2830], Andriy Myronenko[0000-0001-8713-7031],
Wentao Zhu[0000-0002-7505-9512], Ziyue Xu[0000-0002-5728-6869],
Xiaosong Wang[0000-0002-3840-5658], and Daguang Xu[0000-0002-4621-881X]

NVIDIA, Bethesda, MD, USA[⋆]

Abstract. Building robust deep learning-based models requires diverse training data, ideally from several sources. However, these datasets cannot be combined easily because of patient privacy concerns or regulatory hurdles, especially if medical data is involved. Federated learning (FL) is a way to train machine learning models without the need for centralized datasets. Each FL client trains on their local data while only sharing model parameters with a global server that aggregates the parameters from all clients. At the same time, each client's data can exhibit differences and inconsistencies due to the local variation in the patient population, imaging equipment, and acquisition protocols. Hence, the federated learned models should be able to adapt to the local particularities of a client's data. In this work, we combine FL with an AutoML technique based on local neural architecture search by training a "supernet". Furthermore, we propose an adaptation scheme to allow for personalized model architectures at each FL client's site. The proposed method is evaluated on four different datasets from 3D prostate MRI and shown to improve the local models' performance after adaptation through selecting an optimal path through the AutoML supernet.

Keywords: Federated Learning, AutoML, Domain Adaptation

1 Introduction

The advancements of the last few years in medical image segmentation were dominated by deep learning (DL) approaches. DL mostly eliminated the need for handcrafting image features. However, it has been arguably replaced by the need of domain experts to design application-specific DL models. In particular, the medical image computing field has been dominated by popular hand-engineered network architectures such as 2D and 3D U-Net [21,2], V-Net [20], High-Res-Net [14], DeepMedic [12], and many others. To get a good network design for a particular problem, one promising direction is to automate the time-consuming model designing process via AutoML techniques. As another major challenge in model development, large amounts of data covering sufficient large range of examples are usually required to train accurate and robust

[⋆] Contact: {hroth,daguangx}@nvidia.com

models. To achieve this goal, hospitals and medical institutes often need to collaborate and host centralized databases for the development of clinical-grade DL models. This can become challenging due to data-privacy and various ethical concerns associated with data sharing in healthcare domain. One approach to combat such issues is through federated learning (FL), where only model and/or DL workflow parameters are shared among participating institutes instead of raw medical data. Furthermore, it is well known that global robustness and local accuracy is in many cases conflicting: models trained on large centralized datasets might not always generalize well to the data at a particular imaging site due to various inconsistencies (scanner models, imaging protocols, patient populations, etc.) among the different sites. In this case, domain adaptation (DA) is often needed. In this work, we propose to systematically tackle the three challenges in a unified framework: combining an FL algorithm with AutoML and the capability of global-local model adaptation. In particular, we implement a “supernet” training strategy that can be trained in a federated setting. We believe AutoML and FL technologies are a natural fit for each other because of their complementary nature. By combining the two, we are also able to address the DA problem. For one, FL can circumvent the problem of hosting and accessing large centralized datasets by distributing the learning effort to several clients with their own local data. FL will only communicate the model gradients after a local round of training to a centralized server which aggregates the results and starts the next round of FL. At the same time, AutoML with supernet design allows us to avoid hand-engineering of dataset-specific network architectures and a particular sub-network of the trained supernet can be used as a way of local domain adaptation to handle inconsistencies between the different contributing data sites. Next, we summarize related works.

AutoML: Recently, deep learning is applied for various applications, such as image recognition, semantic segmentation, object detection, natural image generation, etc.. However, for each specific task, particular network architectures often need to be hand-designed. Neural architecture search (NAS) [3] is one of the most common approaches to circumvent such hand-design of architectures in AutoML for DL applications. The goal of NAS is to automatically design neural network architectures without any human heuristics or assumptions. In addition to the model weights, after searching, the model architecture itself is optimized for the task at hand, while often still being generalizable to other datasets [28]. A common concept in (one-shot) NAS and AutoML literature is the “supernet” [18,1,27]. The main idea behind supernet is that we can create a large neural network including several candidate modules at each level of the networks. This supernet can be trained jointly, and from the supernet, specific sub-networks can be chosen by selecting a path through the module candidates. At deployment, the final architecture is selected from the supernet by assigning path weights to select particular module candidates. Additional budgeting constraints, such as latency or number of model parameters, can be added to find optimal architectures for a given application. Recent works can achieve state-of-the-art results on computer vision tasks while being computationally efficient [25].

Federated Learning: FL enables collaborative and decentralized DL training without sharing raw patient data [19]. Each client in FL trains locally on their own data and then submits their model parameters to a server that accumulates and aggregates the model updates from each client. Once a certain number of clients have submitted their updates, the aggregated model parameters are redistributed to the clients for local model update, and a new round of local training starts. While out of the scope of this work, FL can also be combined with additional privacy-preserving measures to avoid potential reconstruction of training data through model inversion if the model parameters would be leaked to an adversary [13]. Several works have shown the applicability of FL to medical imaging tasks [22,13,26]. Recent work that combines NAS approaches with FL has been proposed for the mobile phone applications [31]. As such, its focus is on reducing the computational requirements on the local edge devices, making its setting quite different from the "cross-silo" FL [10] medical image segmentation investigated here, where the focus is on model performance and personalization. The closest work in motivation to ours is [7] which focuses on the non-I.I.D. setting but is restricted to using toy datasets for classification tasks, like CIFAR-10, and differs in its implementation details.

Domain Adaptation: Domain adaptation aims to tackle data inconsistencies among different domains, or between training data and unknown data. In its simplest form, fine-tuning, also known as transfer learning [23], can help to adapt a pre-trained model to a particular target domain. More recent approaches for DA typically involve some form of adversarial learning to introduce a specific loss that can minimize the feature-level differences among different domains [11] or through gradient back-propagation using adversarial training [4,5]. An alternative approach is coming from the "image translation" field where generative adversarial networks (GAN) are utilized to translate the image of one domain to mimic another domain. An important part of these approaches is the application of some form of cycle-consistency which is essential to train on un-paired data [9,32,30] The common concept of adversarial training suggests that the gradients from external constraints will help balance various domains and change the model's feature representations.

Contributions: Our proposed approach here is similar in that we will ultimately adapt the model's internal feature representations through the selection of an adapted sub-network of the trained supernet, but without the need to use computationally expensive adversarial learning schemes. Our contributions can be summarized as follows:

1. We show that we can successfully train models through federated learning with comparable or better performance to models trained on centrally hosted data.
2. We extent federated learning by introducing an AutoML approach for supernet model training.
3. We show that finding an optimal path through the supernet can act as a form of local domain adaptation and bring performance gains for each individual client.

2 Method

Here we describe the technical details of the FL and AutoML approach utilized in this work. The proposed method can be separated into two steps: 1) FL with AutoML supernet training and 2) local model adaptation by finding the best path through the supernet with respect to the local data. Both FL and AutoML procedures presented are designed for 3D medical image segmentation tasks.

Client-Server-Based Federated Learning: In its typical form, FL utilizes a client-server setup. Each client trains the same model architecture locally on their own data. Once a certain number of clients finished local training, the updated model weights (or their gradients) are sent to the server for aggregation. After aggregation, the new weights on the server are re-distributed to the clients to execute the next round of local model training. After several FL rounds, the models at each client are converged. Each client can be allowed to select their local best model by monitoring a certain performance metric on a local hold out validation set. In our experiments, we implement the `FederatedAveraging` algorithm proposed in [19]. While there exist variants of this algorithm to address particular learning tasks, in its most general form, FL tries to minimize a global loss function $\mathcal{L}$ which can be a weighted combination of K local losses $\{\mathcal{L}_k\}_{k=1}^{K}$ that each is computed on a client k's local data. Hence, FL can be formulated as the task of finding the model parameters ϕ that minimize L given some local data X.

$$\min_{\phi} \mathcal{L}(X;\phi) \quad \text{with} \quad \mathcal{L}(X;\phi) = \sum_{k=1}^{K} w_k \, \mathcal{L}_k(X_k;\phi), \tag{1}$$

where $w_k > 0$ denote the weight coefficients for each client k, respectively. Note, that the local data X_i is never shared among the different clients. Only the model weights are accumulated and aggregated on the server as shown in Algorithm 1.

Algorithm 1 Client-server federated learning with `FederatedAveraging` [19,13]. T is the number of federated learning rounds and n_k is the number of `LocalTraining` iterations minimizing the local loss $\mathcal{L}_k(X_k;\phi^{(t-1)})$ for a client k.

```
procedure FEDERATED LEARNING
    Initialize weights: $\phi^{(0)}$
    for $t \leftarrow 1 \cdots T$ do
        for client $k \leftarrow 1 \cdots K$ do                ▷ Executed in parallel
            Send $\phi^{(t-1)}$ to client $k$
            Receive $(\Delta\phi_k^{(t)}, n_k)$ from client's LocalTraining($\phi^{(t-1)}$)
        end for
        $\phi_k^{(t)} \leftarrow \phi^{(t-1)} + \Delta\phi_k^{(t)}$
        $\phi^{(t)} \leftarrow \frac{1}{\sum_k n_k} \sum_k (n_k \cdot \phi_k^{(t)})$
    end for
    return $\phi^{(t)}$
```

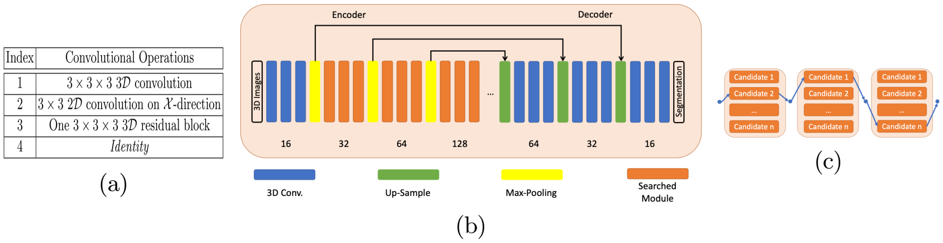
AutoML with Supernet: In order to allow for personalized neural architectures, we designed a supernet $\mathcal{S}$ consisting of various DL module candidates $\mathcal{M}$ suitable for 3D medical imaging tasks shown in Fig. 1a. Each candidate $\mathcal{M}$ is a subgraph

$s \in \mathcal{S}$, denoted as $\mathcal{N}(s,w)$ with model weights w. These modules are optimized at multiple resolution levels to capture different levels of image features useful for the segmentation task. In general, we follow the popular encoder-decoder structure which has been successfully applied to many medical imaging tasks [21,2,20] as shown in Fig. 1b with skip connections that concatenate features of the encoder with their corresponding layer in the decoder path. During training, we choose arbitrary paths m from the module candidates $\mathcal{M}$ following a uniform sampling scheme (see Fig. 1c) to define a sub-network s sampled from the supernet $\mathcal{S}$ as in Eq. 2.

$$W_{\mathcal{S}} = \underset{W}{\operatorname{argmax}}\ \mathcal{L}(\mathcal{N}(\mathcal{S},W)) \tag{2}$$

In this work, we choose the combination of Dice loss [20] and cross entropy loss as our loss function which is commonly used for segmentation tasks in medical imaging [8]. Dice loss' major advantage is its ability to work well in segmentation tasks with an unbalance in the amount of foreground/background regions. Once the supernet is trained, we can find a sub-network s_0 by identifying a locally optimal path through the supernet, effectively adapting the model to the target domain. During adaptation, the model parameters ϕ stay fixed and only the path weights are optimized for one epoch on the local validation set. This results in an optimal path $m_0 \in \mathcal{M}$ that defines our locally adapted sub-network $s_0 \in \mathcal{S}$ as Eq. 3.

$$s_0 = \underset{s \in \mathcal{S}}{\operatorname{argmax}}\ \mathcal{L}_{\text{val}}(\mathcal{N}(s,w_s)) \tag{3}$$



Index	Convolutional Operations
1	$3 \times 3 \times 3$ $3D$ convolution
2	3×3 $2D$ convolution on $\mathcal{X}$-direction
3	One $3 \times 3 \times 3$ $3D$ residual block
4	*Identity*

(a)

(b)

(c)

Fig. 1: (a) Searched network module candidates, (b) our supernet architecture $\mathcal{S}$, and (c) a potential path m selecting several module candidates $\mathcal{M}$. Note, *Identity* is only used if input/output sizes are the same.

3 Experiments & Results

Our proposed method is evaluated on the task of 3D whole prostate segmentation in T2-weighted MRI. In particular, MRI has challenges of data inconsistencies due to variations in different imaging protocols and scanners used at each data contributing site, potentially causing drastic variations in contrast and intensity values.

Datasets: We utilize prostate MRI datasets from four different publicly available data sources. MSD-Prostate[1] [24], PROMISE12 [2] [17], NCI-ISBI13[3], and ProstateX[4] [16]. For each dataset, we perform three random splits into training, validation, and testing sets at roughly 70%, 10%, and 20% of the total number of cases of each dataset. The resulting number of cases for each dataset are shown in Table 1. We average results across the testing splits of each random split. For reference, we show the results on a centralized dataset where all four datasets have been combined. We also compare the performance for models trained locally and through federated learning an each dataset's testing split. The performance of a standard 3D U-Net [2] which is a subgraph of our supernet (when all candidates are type 1) is shown for a baseline comparison. We resample each image to a constant resolution of 0.5 mm × 0.5 mm × 1.0 mm and normalize all non-zero image intensities by subtracting their mean and dividing by their standard derivation on a per-image basis.

Implementation: Both U-Net and the supernet are trained using randomly cropped patches of size $160 \times 160 \times 32$ from the input images and labels. We used a mini-batch size of 18 by selecting 3 random crops from any 6 random input image and label pairs. As the optimizer for training the supernet, we chose *NovoGrad* which has typically faster convergence speed than the more commonly used Adam optimizer [6]. The learning rate for supernet training was set to $1e^{-3}$. For finding the optimal path for the final sub-network we use the Adam optimizer with a learning rate of $1e^{-3}$. Augmentation techniques like random intensity shifts, contrast adjustments, and adding Gaussian noise are applied during training to avoid overfitting to the training set. Our supernet has $3^3 \times 4^6 = 110{,}592$ possible path combinations. Therefore, it is trained 10× longer than 3D U-Net to give it the opportunity to train most paths well. Both 3D U-Net baseline and the supernet are implemented with PyTorch[5] using components from MONAI[6] and NVFlare [7] for FL communication. All models are trained on NVIDIA V100 GPUs with 16 GB memory. We monitor convergence on randomly chosen paths sampled from a uniform distribution during each validation to determine when the supernet is sufficiently trained across clients. The number of training iterations is chosen such that the likelihood of a path being selected during the entire training is at least ¿1.

Results: Table 1 show the performance for assuming local, centrally hosted, and federated datasets. Table 2 shows the better generalizibilty of supernet models trained in the FL. We show the performance of the proposed supernet training approach and its adaption to the local dataset distribution via path optimization, together with a baseline implementation of 3D U-Net using the same augmentation, optimization and hyperparameters to be comparable. Visualization of the results before and after

[1] http://medicaldecathlon.com

[2] https://promise12.grand-challenge.org

[3] http://doi.org/10.7937/K9/TCIA.2015.zF0vlOPv

[4] https://prostatex.grand-challenge.org

[5] https://pytorch.org

[6] https://monai.io

[7] https://pypi.org/project/nvflare

model adaptation are shown in Fig. 2. In descending order, most commonly chosen operations were 3D conv., 3D residual block, 2D conv., followed by identity.

Table 1: Results for centralized dataset and each dataset trained locally and in federated learning. We show the performance of a baseline 3D U-Net and our proposed supernet (SN) approach. The average Dice of the local model's scores is shown (excluding the scores on centralized data). The highest scores for each dataset are marked in **bold**.

Cases	**Central**	**NCI**	**PROMISE12**	**ProstateX**	**MSD**
Training	172	45	35	69	23
Validation	23	6	5	9	3
Testing	48	12	10	20	6
Total	243	63	50	98	32

Avg. Dice [%]	**Central**	**NCI**	**PROMISE12**	**ProstateX**	**MSD**	**Avg. (loc.)**
U-Net (loc.)	89.72	90.15	83.59	90.73	86.17	87.66
U-Net (fed.)		90.59	85.73	90.35	88.40	88.77
SN (loc.)	90.11	90.42	83.51	90.76	86.76	87.86
SN (fed.)		90.57	85.61	**91.07**	88.39	88.91
SN (loc.) + adapt.	**90.15**	90.50	83.46	90.78	87.83	88.14
SN (fed.) + adapt.		**90.68**	**86.15**	90.65	**88.74**	**89.06**

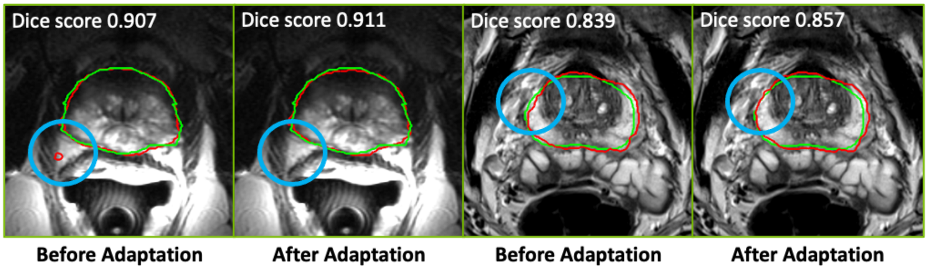


Fig. 2: The visual comparison of results from supernet before and after adaptation. Left two figures show results from PROMISE12, and right two figures are from MSD. Green contours denote ground truth (boundaries), and red contours are predictions from neural networks. We can see that adaptation is not only able to remove false positives, but also capable to correct false negative regions (highlighted by circles).

4 Discussion & Conclusions

It can be observed from Table 1 that the supernet training with local adaptation in FL *(SN (fed.) + adapt.)* achieves the highest average Dice score on the local datasets. At the same time, the adapted models also show the best generalizability (see Table 2). This illustrates the viability of supernet training with local model

Table 2: Generalizability of the supernet models trained locally SN (loc.), in federated learning **SN (fed.)**, and after local adaptation **SN (fed.) + adapt**. Here, client sites A, B, C, and D stand for NCI, PROMISE12, ProstateX, and MSD, respectively. We show the local average performance of the models (**Loc.**, same as in Tab. 1) and the generalizability (**Gen.**), i.e. the Dice score on other clients' test data. The total average generalizability score is denoted as $\overline{\textbf{Gen}}$. and the highest scores for each dataset are marked in **bold**.

	SN (loc.)					**SN (fed.)**					**SN (fed.) + adapt**				
	Test site					*Test site*					*Test site*				
Train site	**A**	**B**	**C**	**D**	**Gen.**	**A**	**B**	**C**	**D**	**Gen.**	**A**	**B**	**C**	**D**	**Gen.**
A	**90.4**	81.5	87.7	81.9	83.7	**90.6**	86.1	90.9	85.5	87.5	**90.7**	86.0	90.6	86.2	87.6
B	84.6	**83.5**	85.3	87.3	85.7	87.4	85.6	89.7	88.3	88.5	88.2	**86.2**	90.3	**89.5**	89.3
C	84.9	72.8	**90.8**	84.3	80.7	89.7	**86.4**	**91.1**	87.7	87.9	90.0	86.0	**90.7**	87.7	87.9
D	77.3	66.0	82.9	**86.8**	75.4	83.1	82.3	86.9	**88.4**	84.1	84.5	83.9	87.3	88.7	85.2
	Loc.	***87.9***		$\overline{\textbf{Gen}}$.	*81.4*	***Loc.***	***88.9***		$\overline{\textbf{Gen}}$.	*87.0*	***Loc.***	***89.1***		$\overline{\textbf{Gen}}$.	***87.5***

adaption to the client's data. We furthermore observe a general improvement of the local supernet models' performance when trained in an FL setting versus local training. This means that in particular supernet model training can benefit from the larger effective training set size made available through FL without having to share any of the raw image data between clients. Overall, we achieve average Dice scores comparable to recent literature on whole prostate segmentation in MRI [16,17,20] and can likely be improved with more aggressive data augmentation schemes [29,8]. Further fine-tuning of the network weights (not the supernet path weights) is likely going to give performance boost on a local client but is also expected to reduce generalizability of the model. Methods of fine-tuning that do not reduce the robustness to other data sources (i.e. generalizability) gained through FL (e.g. learning without forgetting [15]) is still an open research question and was deemed to be out of scope of this work.

In conclusion, we proposed to combine the advantages of both federated learning and AutoML. The two techniques are complementary and in combination, they allow for an implicit domain adaptation through the finding of locally optimal model architectures (sub-networks of the supernet) for a client's dataset. We showed that the performances of federated learning are comparable to the model's performance when the dataset is centrally hosted. After local adaptation via choosing the optimal path through the supernet, we can see an additional performance gain on the client's data. In the future, it could be explored if there is a set of optimal sub-networks that could act as an ensemble during inference to further improve performance and provide additional estimates such as model uncertainty. Furthermore, one could adaptively change the path frequencies used during supernet training based on sub-network architectures that work well on each client in order to reduce communication cost and speed-up training.

References

1. Cai, H., Zhu, L., Han, S.: Proxylessnas: Direct neural architecture search on target task and hardware. arXiv preprint arXiv:1812.00332 (2018)
2. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: MICCAI. pp. 424–432. Springer (2016)
3. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. arXiv preprint arXiv:1808.05377 (2018)
4. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37. pp. 1180–1189. JMLR. org (2015)
5. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
6. Ginsburg, B., Castonguay, P., Hrinchuk, O., Kuchaiev, O., Lavrukhin, V., Leary, R., Li, J., Nguyen, H., Cohen, J.M.: Stochastic gradient methods with layer-wise adaptive moments for training of deep networks. arXiv preprint arXiv:1905.11286 (2019)
7. He, C., Annavaram, M., Avestimehr, S.: Towards non-iid and invisible data with fednas: Federated deep learning via neural architecture search. arXiv preprint arXiv:2004.08546 (2020)
8. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**(2), 203–211 (2021)
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 1125–1134 (2017)
10. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977 (2019)
11. Kamnitsas, K., Baumgartner, C., Ledig, C., Newcombe, V., Simpson, J., Kane, A., Menon, D., Nori, A., Criminisi, A., Rueckert, D., et al.: Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In: International conference on information processing in medical imaging. pp. 597–609. Springer (2017)
12. Kamnitsas, K., Ferrante, E., Parisot, S., Ledig, C., Nori, A.V., Criminisi, A., Rueckert, D., Glocker, B.: Deepmedic for brain tumor segmentation. In: International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries. pp. 138–149. Springer (2016)
13. Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., Baust, M., Cheng, Y., Ourselin, S., Cardoso, M.J., et al.: Privacy-preserving federated brain tumour segmentation. In: International Workshop on Machine Learning in Medical Imaging. pp. 133–141. Springer (2019)
14. Li, W., Wang, G., Fidon, L., Ourselin, S., Cardoso, M.J., Vercauteren, T.: On the compactness, efficiency, and representation of 3d convolutional networks: brain parcellation as a pretext task. In: International conference on information processing in medical imaging. pp. 348–360. Springer (2017)
15. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
16. Litjens, G., Debats, O., Barentsz, J., Karssemeijer, N., Huisman, H.: Computer-aided detection of prostate cancer in MRI. TMI **33**(5), 1083–1092 (2014)

17. Litjens, G., Toth, R., van de Ven, W., Hoeks, C., Kerkstra, S., van Ginneken, B., Vincent, G., Guillard, G., Birbeck, N., Zhang, J., et al.: Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge. Medical image analysis **18**(2), 359–373 (2014)
18. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
19. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., et al.: Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629 (2016)
20. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). pp. 565–571. IEEE (2016)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)
22. Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., Bakas, S.: Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 92–104. Springer (2018)
23. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. TMI **35**(5), 1285–1298 (2016)
24. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019)
25. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114 (2019)
26. Yang, D., Xu, Z., Li, W., Myronenko, A., Roth, H.R., Harmon, S., Xu, S., Turkbey, B., Turkbey, E., Wang, X., Zhu, W., Carrafiello, G., Patella, F., Cariati, M., Obinata, H., Mori, H., Tamura, K., An, P., Wood, B.J., Xu, D.: Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from China, Italy, Japan. Medical Image Analysis **70**, 101992 (2021)
27. You, S., Huang, T., Yang, M., Wang, F., Qian, C., Zhang, C.: Greedynas: Towards fast one-shot nas with greedy supernet. In: CVPR. pp. 1999–2008 (2020)
28. Yu, Q., Yang, D., Roth, H., Bai, Y., Zhang, Y., Yuille, A.L., Xu, D.: C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. In: CVPR. pp. 4126–4135 (2020)
29. Zhang, L., Wang, X., Yang, D., Sanford, T., Harmon, S., Turkbey, B., Wood, B.J., Roth, H., Myronenko, A., Xu, D., et al.: Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. TMI (2020)
30. Zhang, Y., Miao, S., Mansi, T., Liao, R.: Task driven generative modeling for unsupervised domain adaptation: Application to x-ray image segmentation. In: MICCAI. pp. 599–607. Springer (2018)
31. Zhu, H., Jin, Y.: Real-time federated evolutionary neural architecture search. arXiv preprint arXiv:2003.02793 (2020)
32. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)