Title

A tool for federated training of segmentation models on whole slide images

Permalink

https://escholarship.org/uc/item/716097nk

Authors

Lutnick, Brendon
Manthey, David
Becker, Jan U
et al.

Publication Date

2022

DOI

10.1016/j.jpi.2022.100101

Journal of Pathology Informatics 13 (2022) 100101

Contents lists available at ScienceDirect

Journal of Pathology Informatics

journal homepage: www.elsevier.com/locate/jpi

Technical Note

A tool for federated training of segmentation models on whole slide images

Brendon Lutnick [a], David Manthey [b], Jan U. Becker [c], Jonathan E. Zuckerman [d], Luis Rodrigues [e], Kuang-Yu Jen [f], Pinaki Sarder [a,*]

[a] Department of Pathology and Anatomical Sciences, SUNY Buffalo, Buffalo, NY, USA
[b] Kitware Incorporated, Clifton Park, NY, USA
[c] Institute of Pathology, University Hospital Cologne, Cologne, Germany
[d] Department of Pathology and Laboratory Medicine, University of California at Los Angeles, Los Angeles, CA, USA
[e] University Clinic of Nephrology, Faculty of Medicine, University of Coimbra, Portugal
[f] University of California, Davis School of Medicine, Sacramento, CA, USA

ARTICLE INFO

Keywords:
Computational pathology
Cloud computing
Federated learning
Renal pathology
Interstitial fibrosis and tubular atrophy

ABSTRACT

The largest bottleneck to the development of convolutional neural network (CNN) models in the computational pathology domain is the collection and curation of diverse training datasets. Training CNNs requires large cohorts of image data, and model generalizability is dependent on training data heterogeneity. Including data from multiple centers enhances the generalizability of CNN-based models, but this is hindered by the logistical challenges of sharing medical data. In this paper, we explore the feasibility of training our recently developed cloud-based segmentation tool (Histo-Cloud) using federated learning. Using a dataset of renal tissue biopsies we show that federated training to segment interstitial fibrosis and tubular atrophy (IFTA) using datasets from three institutions is not found to be different from a training by pooling the data on one server when tested on a fourth (holdout) institution's data. Further, training a model to segment glomeruli for a federated dataset (split by staining) demonstrates similar performance.

Introduction

As the practice of digitizing histological slides has become common practice,[1] the field of computational pathology has exploded. Modern image analysis technologies (such as deep learning[2]) are increasingly being applied to examine whole-slide images (WSIs). The maturation of convolutional neural networks (CNNs)[3] (a specialized subset of deep learning) for the analysis and segmentation of natural images has led to widespread adoption of this technology in the field of computational pathology. CNNs have shown promising results for state of the art computational pathology image analysis tasks including tissue segmentation,[4–8] disease classification,[9–12] and outcome prediction.[13,14] Training these networks is enhanced by access to diverse WSI datasets, as greater data variability is known to enhance model robustness.[15] For histological tissue, stained and scanned digitally as WSIs, the institution where data is prepared often has a large effect on the quality and appearance of the tissue.[16] Institution-specific factors such as tissue preparation and staining protocol, as well as any demographic biases can have a large effect on the resulting WSIs. Practically this means gathering training data from multiple institutions. However, sharing medical data across institutions can be complicated by regulatory challenges,[17] limiting the scope of collaboration and therefore the generalizability of computational pathology tools.

Federated learning was recently proposed as an efficient solution for decentralized training of models without sharing data.[18,19] Training a network on a federated dataset uses multiple rounds of local training performed on hardware located at the data source, where the learned network parameters are shared and averaged between each round to avoid divergence between training sites. At the core of federated learning is federated averaging (FedAvg),[20] which is simply a weighted average of the network weights across training sites, performed at pre-selected intervals (Fig. 1). FedAvg has been practically shown to achieve convergence with proper hyperparameter tuning.[21] Originally proposed for smartphone natural language processing tasks where data sharing is limited by a limited network bandwidth and privacy concerns, federated learning has recently gained the interest of computational researchers in the medical field.[22] Computational pathology datasets are a perfect candidate for federated learning where both file sizes of WSIs (gigapixels) and regulatory limitations hinder data sharing.

To show the feasibility of federated learning on pathology data in the real world, we have created a pipeline for federated segmentation on WSIs capable of deployment across multiple institutions. This pipeline is deployed in the cloud for easy access for data viewing and annotation by each site's constituents. This companion work to our recently published Histo-Cloud segmentation tool[8] shows the feasibility for training Histo-Cloud in

[*] Corresponding author.
E-mail address: pinakisa@buffalo.edu (P. Sarder).

http://dx.doi.org/10.1016/j.jpi.2022.100101
Received 20 February 2022; Received in revised form 6 April 2022; Accepted 9 April 2022
Available online 21 May 2022

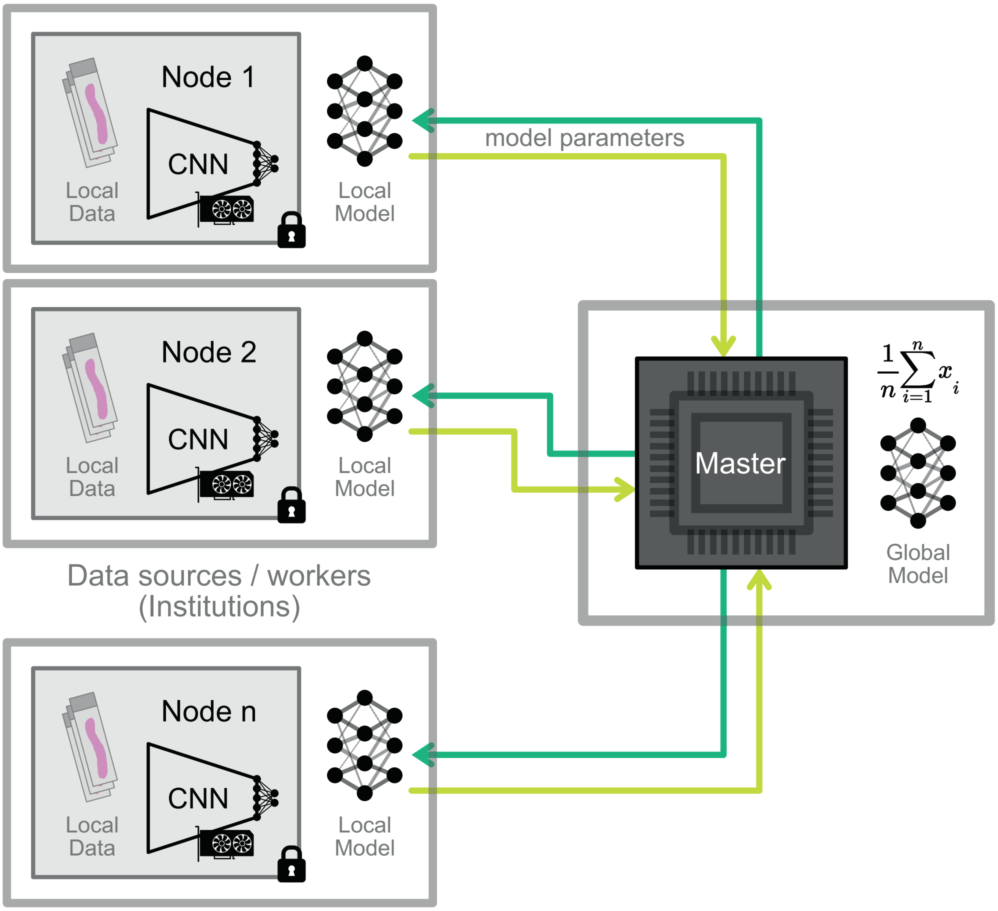

Fig. 1. The federated learning schematic. A schematic example of federated learning. Multiple worker nodes store data and model parameters locally at the institution of origin. The data stored on these worker nodes is never shared, and the nodes perform local training using this data upon the request of the master server. The local models are then shared with the master server which performs parameter averaging, before sending the updated global model back to the worker nodes for further local training. This process is repeated iteratively throughout the training process until model convergence.

a federated setup. Histo-Cloud is a cloud-based tool for segmentation of WSIs. It combines the digital slide archive (DSA)[23] for WSI data management, HistomicsUI for WSI viewing and annotation, and a modified version of the DeepLab V3 + network[5] for WSI segmentation.

Our contributions are:

1) The first tool for federated segmentation of WSIs

2) Hyper-parameter recommendations for fast and reproducible convergance

3) Validation of this tool using three physically separate servers

Results

Federated training was performed on data distributed across three discrete servers (workers). A fourth server acted as the master server, performing parameter averaging and training synchronization; a schematic is available in Fig. 1. In each server, data is stored on an instance of the DSA,[23] and our Histo-Cloud plugin[8] is responsible for network training. This plugin is capable of utilizing hardware acceleration for training, and uses two available GPUs in all three host machines for a total of 6 GPUs. We demonstrate the feasibility of federated segmentation of WSIs with two case studies:

1 - Federated IFTA segmentation (divided by institution):

For the first case study, interstitial fibrosis and tubular atrophy (IFTA) was segmented from WSIs from kidney transplant biopsies with chronic allograft nephropathy stained using periodic acid Schiff (PAS). Three pathologists from different institutions each provided a minimum of 20 PAS stained WSIs. The WSIs per set were uniformly chosen from four IFTA classes defined based on ci/ct scores (0, 1, 2, & 3); ci/ct scoring is a method defined in Banff 2018 criteria for assessing transplant biopsies with ci addressing cortical interstitial fibrosis and ct addressing tubular atrophy.[24] A minimum of 5 slides per class were used for each set. The cases were reviewed to ensure the following selection criteria were met: (1) the amount of early or evolving IFTA with variable intermixed edema was minimized, (2) no active inflammation, (3) no prior history of rejection, and (4) cases were selected to represent the full range of IFTA severity. All types of IFTA, including classic, endocrinization, and thyroidization types, were included in the analysis, without distinguishing between the types. In total, the pathologists from institutions 1–3 provided 20, 48, and 22 slides respectively. A holdout dataset was randomly selected by pooling 1/3rd of the slides from each institution (29 slides total). We trained 5 models using this dataset: The first model was trained across three federated servers, training data for this study was split by institution of origin. For a baseline performance, a second model was trained centrally by pooling all the training data on a single server and using traditional gradient decent. Finally, to compare the performance in a data restricted setting, 3 additional models were trained using data from a single institution alone.

We note that IFTA boundaries are poorly defined, and subject to disagreement between pathologists.[25] Receiver operating characteristic (ROC) curves were used to better capture the performance characteristics of our trained models. These were generated by applying a varying threshold to the network logits for the prediction of IFTA regions. To measure performance, we calculate the area under the curve (AUC) which is a common metric for measuring performance when a ROC curve is available.

Testing these models on the holdout set, we observed that central training and federated training of the IFTA model performed similarly both with *AUC = 0.95*. Performance fell when testing the models trained using a single institutions data, giving *AUC = 0.92, 0.87, & 0.91* respectively. ROC plots of the performance of the 5 models is highlighted in Fig. 2a. An example of IFTA segmentation on a holdout slide using the federated model is shown in Fig. 2c. Here we use the network logits to display the predictions as a probabilistic heatmap which we believe is better for the display of structures with poorly defined boundaries such as IFTA.

A fourth pathologist from a different institution provided an additional 17 slides to be used as an independent testing dataset. When we applied the trained IFTA models to this independent set, we observed a similar trend as the holdout set. Here the federated model performed best with *AUC = 0.90* and the central model also performed similarly well with *AUC = 0.88*. Like the holdout set, performance of the models trained on a single institution was lower than federated or central models, with *AUC = 0.85, 0.81, & 0.84* respectively. ROC plots of the performance of the five models are highlighted in Fig. 2b.

2 - Federated glomeruli segmentation (divided by stain):

As a further test of our method, we designed a second study focused on glomerular segmentation, with a goal of studying the effects of tissue staining on federated training. A training set of 75 human renal WSIs from transplant biopsies, stained with 25 periodic acid Schiff (PAS), 25 hematoxylin & eosin (H&E), and 25 Masson's trichome (TRI). These slides were selected from a single institution and ground truth glomeruli annotation was performed by a single annotator. Slides were divided by stain and uploaded to the 3 training servers (25 slides per server). Like the IFTA study, 5 models were trained: A federated model, a central model using all the data, and 3 models trained on each single stain individually. A holdout set of 30 slides (10 from each stain) was selected to test the model's performance.

Unlike IFTA, glomerular boundaries are well defined and are best displayed by directly using the network predictions. We convert these predictions to contours for display on the slide. Fig. 3c shows an example of glomerular boundaries predicted using the federated model on holdout slides of various stains. We choose to use Matthews correlation coefficient (MCC) to measure the performance of glomerular segmentation, as it is commonly used for binary segmentation tasks. On the holdout data the federated model and the central model had identical performance, both achieving *MCC = 0.91,* outperforming all the models trained using only one stain (*MCC = 0.88* H&E model, *0.56* PAS model, and *0.85* TRI model). A violin plot of the holdout performance as a function of the model used is shown in Fig. 3a.

To further show the model's generalizability, an independent testing set of 58 slides was chosen from a separate institution, and annotated by a

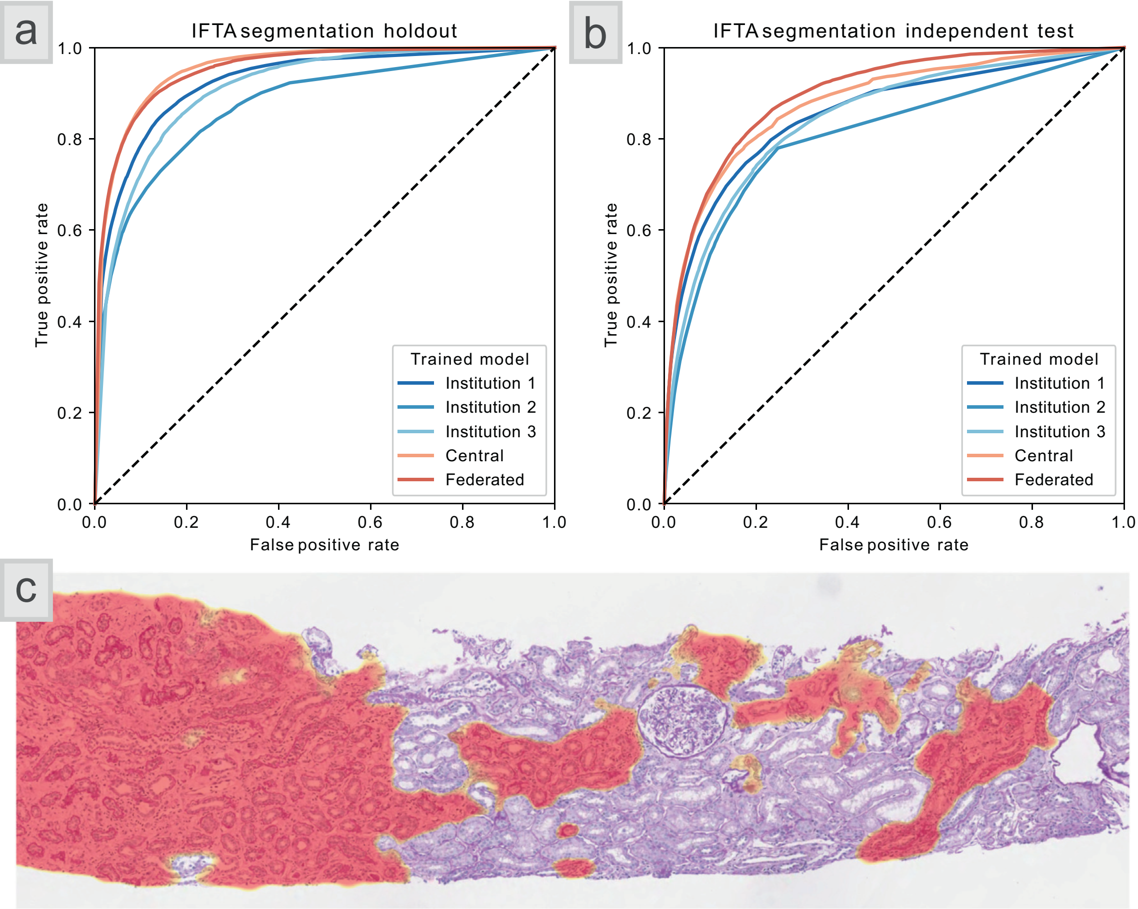

Fig. 2. Multi-institute IFTA segmentation performance, data split by institution. The segmentation performance of the trained IFTA models. This data was split by institution across 3 servers for federated training. Due to the subjective nature of IFTA boundaries, we use ROC curves and AUC to measure the segmentation performance. [a] ROC curves showing each models performance on a dataset of 29 holdout WSIs which were randomly selected from the same data as the training set. We observed that central training and federated training of the IFTA model performed similarly both with *AUC = 0.95*. Performance fell when testing the models trained using a single institutions data, giving *AUC = 0.92, 0.87, & 0.91* respectively. [b] ROC curves showing each models performance on an independent test set of data containing 17 WSIs. This dataset was from an institution which did not provide any training data, and was annotated by an independent pathologist. Similar to the holdout set, the central and federated models outperformed the models trained on a single institution's data. Interestingly the federated model performed best with *AUC = 0.90* and the central model also performed well with *AUC = 0.88*. The institutions 1, 2, & 3 had *AUC = 0.85, 0.81, & 0.84* respectively. [c] An example of IFTA segmentation using the federated model on a slide from the holdout dataset. The prediction of IFTA is shown here using a heatmap, which reflects the network confidence in IFTA segmentation.

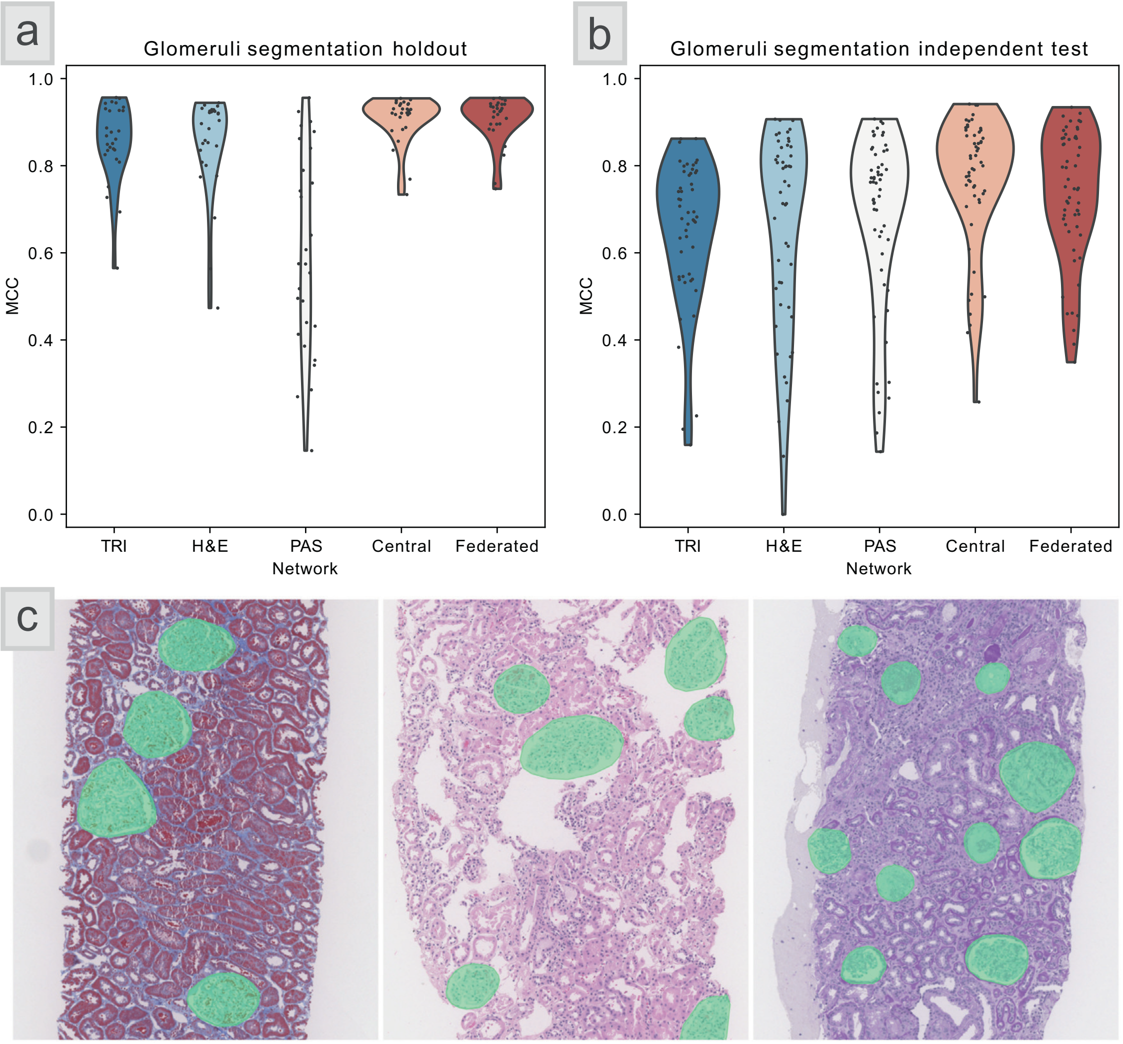

Fig. 3. Glomeruli segmentation performance, data split by stain. The segmentation performance of the models trained for glomeruli segmentation. This data was split by stain across 3 servers for federated training. Because glomeruli have well-defined boundaries, we use MCC to calculate the segmentation performance, without varying the network prediction thresholds. [a] A violin plot showing the performance of each model on a dataset of 30 holdout WSIs which were randomly selected from the same data as the training set. We observed that central training and federated training of the models performed similarly both with MCC = 0.91. Performance fell when testing the models trained using a single stain, giving MCC = 0.85, 0.88, & 0.56 for TRI, H&E, & PAS stains respectively. [b] A violin plot showing the performance of each model on a dataset of 58 holdout WSIs which were selected from an independent institution. This dataset also included WSIs stained with Jones, which was not used for training. The federated model (MCC = 0.80) was outperformed by the central model (MCC = 0.83). However, the federated model still outperformed the models trained using a single stain alone (MCC =0.65 TRI, 0.69 H&E, & 0.78 PAS). [c] Examples of glomeruli segmentation using the federated model on 3 slides from the holdout dataset. From left to right the slides are stained with trichrome, H&E, and PAS.

separate annotator. This data included PAS, H&E, and TRI stains, as well as Jones stain which was not present in the training set. As expected, the segmentation performance was reduced on the independent test set. The federated model (*MCC = 0.80*) was outperformed by the central model (*MCC = 0.83*). However, the federated model still outperformed the models trained using a single stain alone (*MCC = 0.69* H&E model, *0.78* PAS model, and *0.65* TRI model). Interestingly when examining the performance of each model on the individual validation slides, the federated and centrally trained models both achieve a similar maximum performance threshold and a minimum performance which is higher than any model trained on a singular stain. This trend can be seen in the violin plot of the holdout performance as a function of the model used is shown in Fig. 3b.

Discussion

Numerous examples of federated learning on medical data exist,[26–29] however at this time computational pathology research on federated learning using WSIs is limited to a paper by Lu et al.[30] Lu et al trained a weakly supervised, multi instance learning model for subtyping breast cancer and renal cell carcinoma and predicting survival, while exploring the effects of differential privacy[31] on model performance. Setting aside the complexities of network hyperparameter tuning, we argue that federated learning is a data organization and synchronization problem at its core. While current applications in the literature describe the hyperparameters used for training, their data management and synchronization strategies lack details. Often federated learning research is performed locally in one machine, relying on simulated data sites.[18] For example, the details of the federated setup used by Lu et al[30] are not well-described, and it is unclear if the federated training was simulated or actually performed across physically distinct servers. While simulation results are valid for method development, we argue that the complexities of managing data and coordination across multiple training sites are a large logistical hurdle for real world applications of federated learning.

Our experiments (IFTA & glomeruli segmentation) show that not only does federating training for WSI segmentation converge, but the resultant model outperforms training done with a single dataset (institution or

Table 1
Federated learning performance and training time.

Experiment	Training WSIs	Dataset	MCC		Training time (hours)
			Holdout	Independent	
IFTA	61	Federated	0.75	0.44	32.5
		Central	0.75	0.45	19.5
Glomeruli	30	Federated	0.91	0.79	27.7
		Central	0.91	0.82	12.4

The training time and performance of federated and central training are compared. We report the performance using Matthews correlation coefficient (MCC), and separate the performance on the holdout, and independent test datasets for each model.

stain). Furthermore, the federated model performs on par with a model trained traditionally with multiple datasets gathered at a central location. Most importantly, these experiments demonstrate the feasibility of training and coordinating federated segmentation models, managing datasets distributed across physically separate servers, and training in a reasonable time. The training times for federated and central learning are reported in Table 1.

Compared to training a model centrally, the parameter sharing and averaging of federated learning adds additional time to the training process. However, as with all forms of federated learning there is a tradeoff between the training time and model convergence which can be tuned with the frequency of the parameter averaging. In our experiments we found that averaging every 1000 steps (with a total of 40,000 training steps) produced a model which converged with a reasonable time penalty. In the future, we would like to study the frequency of federated averaging with respect to model convergence.

We are not the first to propose federated segmentation, Yi et al proposed SU-Net,[32] a federated network for brain tumor segmentation, which performed similarly to DeepLab[5] for non-federated training. The first to train the DeepLab V3+[5] architecture in a federated setup was Michieli et al,[33] who used the VOC2012 dataset[34] and simulated federated training on a single machine. In contrast, our federated approach offers comparable segmentation performance to the traditional training of DeepLab on gigapixel sized medical images (WSIs). While there is a time penalty for conducting federated training, we believe that this tradeoff is worth the added performance when comparing the performance to training done using a single dataset alone, and plan to further optimize our pipeline for speed in the future.

Working efficiently with WSIs using CNNs requires a substantial amount of engineering effort, and the backbone of our code used for training was custom built to extract and process regions of interest from WSIs efficiently. We believe the ability to easily manage and annotate WSI data at each federated site using the DSA[23] greatly enhances the real world applications of our method.

Throughout our training process, the newest segmentation model is available for testing at each data site, and could theoretically be used in a human-in-the-loop approach to aid in the annotation of new WSIs similar to our previously described H-AI-L approach.[7] Newly added WSIs will automatically be incorporated into the training set at the beginning of each round of training.

Approaches such as peer-to-peer federated learning[35] and swarm learning[36] offer data synchronization strategies that do not require a central coordinating (master) server. While the lack of centralized training coordination may be beneficial for some tasks, we argue that for federated medical image segmentation, it is likely that only one group will be responsible for model development. Therefore, the ability to control and monitor training and adjust hyperparameters on one master server is ideal. The typical setup of federated learning in a medical setting will involve orders of magnitude fewer training sites than a task such as speech recognition, which has millions of potential training sites such as mobile phones. Multi-institute federated studies using medical images will require careful central coordination with recruitment and opt in by participating sites. The hardware for performing the training (at least for medical image analysis) is specialized, requiring IT setup and support at each institution. Our Histo-Cloud tool (used for training) is easy to setup making it ideal for this purpose.

Methods

An open-source version of our code is available here: https://github.com/SarderLab/federated_learning.

Data acquisition

This study was approved by the Institutional Review Boards at the University at Buffalo, University of California Davis, University of California Los Angeles, University of Coimbra, and University Hospital Cologne. All methods were performed in accordance with the relevant federal guidelines and regulations. All patients provided informed consent.

Segmentation plugin

This work is heavily based upon our previously published Histo-Cloud tool,[8] where we modified the DeepLab V3+ architecture[5] to work natively on WSIs and developed a series of plugins for running segmentation training and prediction in the cloud. This work was based on the Digital Slide Archive (DSA)[23] an open source slide viewer and repository developed by Kitware Inc. Specifically these plugins were developed for accessibility in HistomicsUI, the slide viewing component of the DSA. Functions of the DSA can be controlled using a REST web API,[37] this includes the ability to trigger jobs by running the installed plugins as well as upload and download data stored in the DSA. Achieving federated learning using this system was straightforward. We use the requests python library[38] to send REST calls to the federated workers, which all have the DSA and Histo-Cloud installed and are hosting the respective training datasets.

Server coordination

In this pipeline, each site/institution has a worker node server with the DSA installed where training data is uploaded and annotated. Using the DSA, data permissions can be set so that only eligible users from the institution can access this data. A central (master) server manages the training cycle, uploading the global model parameters to each worker via the DSA REST API before requesting each to run local training. Training Jobs are submitted to each worker (training site) using the Histo-Cloud training plugin.[8] The training job scheduling is handled by the DSA internally using *slicer_cli_web*, which uses Celery[39] for task queue management, and RabbitMQ[40] as a message broker. The job status is monitored by the master server until completion. Upon job completion the master server requests and downloads the resultant saved local model parameters from each worker node. These parameters are averaged by the master server and the global model is updated accordingly. The next round of training is then initiated: the global model is uploaded to each worker and is trained further before being downloaded and averaged. If training fails on one of the participating workers, then it is excluded from the rest of the training round, but participates in future training rounds.

Data management

The training WSI data is uploaded to the DSA worker servers, where it was annotated by expert pathologists. Training data is placed in a folder created on each worker for easy access by the Histo-Cloud training plugin. A separate folder was created for the models produced by training and uploaded after federated averaging. The ID of these folders is known by the master server so it can submit training jobs specifying the data and models to be used for training.

Training process

Training rounds involve parameter upload, training, parameter download, and federated parameter averaging across the worker and master nodes. The following pseudocode describes the training process:

```
INIT global model to ImageNet parameters
WHILE global training steps is less than total steps
    FOR each client, in parallel do
        UPLOAD global model parameters to clients
        CALL TrainNetwork plugin involving
            INIT network parameters with global model parameters
            TRAIN for number of steps in round
        UPDATE global training steps
        DOWNLOAD trained local model
    END FOR
    COMPUTE average model parameters (FedAvg)
    SET global model to FedAvg parameters
END WHILE
```

Training setup

For training, we used three physically distinct Linux servers running Ubuntu 18.04.5 LTS, with the DSA installed. The three servers had different hardware configurations, notably the graphics processing units (GPUs) were different across the servers. All computers had 2 GPUs that were produced by the Nvidia corporation and included:

1) Titan X Pascale (12GB VRAM) & GeForce GTX 1080 (8GB VRAM) – batch size 4.
2) GeForce RTX 2080 Ti (11GB VRAM) & GeForce GTX 1080 (8GB VRAM) – batch size 4.
3) 2X Quadro RTX 5000 (16 GB VRAM) – batch size 12.

For training, we used both available GPUs on each server and adjusted the batch size for each server to accommodate the individual VRAM (GPU memory) capacity of each.

Training hyperparameters

The goal of federated averaging is to speed up training by removing the overhead of frequent communication between training sites. This is done by training locally for multiple steps before updating the central model parameters using FedAvg. Practically when optimizing the hyperparameters of our training loop, we found that using 1000 training steps between FedAvg achieved repeatable convergence. We trained for a total of 40 rounds (40,000 steps), using the momentum optimizer[41] with an initial learning rate of $7e^{-3}$. Polynomial decay with a learning power of 0.9 was used to reduce the initial learning rate over the course of training, ending on the value of 0.0 after the final training step. To achieve stability at the start of training, we set the learning rate to $1e^{-4}$ for the first 750 steps. Finally, the gradients on the last layers of the network were scaled up by a factor of 10 to achieve faster convergence. These layers included the ASPP pooling layers and the layers in the decoder as defined by the DeepLab network architecture.[5]

All the models were trained using transfer learning with parameters inherited from a model pre-trained on the ImageNet dataset. Due to the low maximum batch size of 4 on 2 of the training servers, we did not train the batch normalization parameters. Training patches with a size of 512x512 pixels were extracted from the training WSIs with various downsampled resolutions. Here we followed the training protocol in the Histo-Cloud work[8] using training patches randomly downsampled to 1, 2, 3, & 4 times smaller with respect to the native WSI resolution.

Author Contributions

B.L. wrote the manuscript, the code, and performed the experiments. D.M. provided technical advice about using the Digital Slide Archive, particularly the REST API, and edited the manuscript. J.U.B. provided and annotated the independent validation datasets used for the IFTA and glomeruli segmentation. J.E.Z. and L.R. provided and annotated datasets used to train the IFTA segmentation model. K.U.J. provided and annotated a dataset used for IFTA training as well as the glomeruli segmentation training set. P.S. conceived the overall research plan, coordinated with the multi-disciplinary study team on the project and edited the manuscript.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project was supported by NIH-NIDDK grant R01 DK114485 (PS), NIH-OD grants R01 DK114485 02S1 & 03S1 (PS), a pilot grant from the NIH-NIDDK CKD Biomarker Consortium grant U01 DK103225 (PS), via the opportunity pool funding mechanism, namely via glue grant mechanism (PS) of the NIH-NIDDK Kidney Precision Medicine Project (KPMP) consortium grant U2C DK114886 (Contact: Dr. Jonathan Himmelfarb), a multi-disciplinary small team grant RSG201047.2 (PS) from the State University of New York, a pilot grant (PS) from the University of Buffalo's Clinical and Translational Science Institute (CTSI) grant 3UL1TR00141206 S1 (Contact: Dr. Timothy Murphy), a DiaComp Pilot & Feasibility Project 21AU4180 (PS) with support from NIDDK Diabetic Complications Consortium grants U24 DK076169 and U24 DK115255 (Contact: Dr. Richard A. McIndoe), and NIH-OD Human Biomolecular Atlas Project (HuBMAP) consortium grant U54 HL145608 (PS). The project was also supported by European Rare Kidney Disease Network and the Deutsche Forschungsgemeinschaft (BE-3801) (JUB).

References

1. Farahani N, Parwani AV, Pantanowitz L. Whole slide imaging in pathology: advantages, limitations, and emerging perspectives. Pathol Lab Med Int 2015;7:23–33.
2. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–444.
3. Gu J, JasonKuen Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, Chen T, et al. Recent advances in convolutional neural networks. Pattern Recog 2018;77:354–377.
4. Bueno G, Fernandez-Carrobles MM, Gonzalez-Lopez L, Deniz O. Glomerulosclerosis identification in whole slide images using semantic segmentation. Computer Methods Prog Biomed 2020;184:105273.
5. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. Proceedings of the European conference on computer vision (ECCV). 801-818.
6. Ronneberger, O., Fischer, P. & Brox, T. International Conference on Medical image computing and computer-assisted intervention. 234-241 (Springer).
7. Lutnick B, Ginley B, Govind D, McGarry SD, LaViolette PS, Yacoub R, Jain S, Tomaszewski JE, Jen KY, Sarder P. *An integrated iterative annotation technique for easing neural network training in medical image analysis*, 1. 2019:112.
8. Lutnick BR, et al. A user-friendly tool for cloud-based whole slide image segmentation, with examples from renal histopathology. bioRxiv 2021. https://doi.org/10.1101/2021.08.16.456524.2021.2008.2016.456524.
9. Folmsbee, J., Liu, X., Brandwein-Weber, M. & Doyle, S. 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). 770-773 (IEEE).
10. Ginley B, Lutnick B, Jen KY, Fogo AB, Jain S, Rosenberg A, Walavalkar V, Wilding G, Tomaszewski JE, Yacoub R, Rossi GM, Sarder P. Computational segmentation and classification of diabetic glomerulosclerosis 2019;30:1953–1967.
11. Lee S, Amgad M, Mobadersany P, McCormick M, Pollack BP, Elfandy H, Hussein H, Gutman DA, Cooper LAD. Interactive classification of whole-slide imaging data for cancer researchers. Cancer Res 2021;81:1171–1177.
12. Campanella G, Hanna MG, Geneslaw L, Miraflor A, Silva VWK, Busam KJ, Brogi E, Reuter VE, Klimstra DS, Fuchs TJ. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. Nat Med 2019;25:1301–1309.

13. Bychkov D, Linder N, Turkki R, Nordling S, Kovanen PE, Verrill C, Walliander M, Lundin M, Haglund C, Lundin J. Deep learning based tissue analysis predicts outcome in colorectal cancer. Scient Rep 2018;8:1-11.
14. Kolachalama VB, Singh P, Lin CQ, Mun D, Belghasem ME, Henderson JM, Francis JM, Salant DJ, Chitalia VC. Association of pathological fibrosis with renal survival using deep neural networks. Kidney Int Rep 2018;3:464–475.
15. Abels E, Pantanowitz L, Aeffner F, Zarella MD, Laak JVD, Bui MM, Vemuri VN, Parwani AV, Gibbs J, Agosto-Arroyo E, Beck AH, Kozlowski C. Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the Digital Pathology Association. J Pathol 2019;249:286–294.
16. Dimitriou N, Arandjelović O, Caie PD. Deep learning for whole slide image analysis: an overview. Front Med 2019;6:264.
17. Scheibner J, Lenca M, Kechagia S, Troncoso-Pastoriza JR, Raisaro JL, Hubaux JP, Fellay J, Vayena E. Data protection and ethics requirements for multisite research with health data: a comparative examination of legislative governance frameworks and the role of data protection technologies. J Law Biosci 2020;7:lsaa010.
18. Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: Strategies for improving communication efficiency. arXiv preprint 2016.arXiv: 1610.05492.
19. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intel Syst Technol (TIST) 2019;10:1-19.
20. McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. Artificial Intelligence and Statistics. 1273-1282 (PMLR).
21. Li X, Huang K, Yang W, Wang S, Zhang Z. On the convergence of fedavg on non-iid data. arXiv preprint 2019.arXiv:1907.02189.
22. Rieke N, Hancox J, Li W, Milletarì F, Roth HR, Albarqouni S, Bakas S, Galtier MN, Landman BA, Maier-Hein K, Ourselin S, Sheller M, Summers RM, Trask A, Xu D, Baust M, Cardoso MJ. The future of digital health with federated learning. NPJ Digit Med 2020;3:1–7.
23. Gutman DA, Lee S, Nalisnik M, Mullen Z, Beezley J, Chittajallu DR, Manthey D, Cooper LAD. The digital slide archive: a software platform for management, integration, and analysis of histology for cancer research. Cancer Res 2017;77:e75–e78.
24. Roufosse C, Simmonds N, Clahsen-van Groningen M, Haas M, Henriksen KJ, Horsfield C, Loupy A, Mengel M, Perkowska-Ptasińska A, Rabant M, Racusen L, Solez K, Becker JU. A 2018 reference guide to the Banff classification of renal allograft pathology. Transplantation 2018;102:1795–1814.
25. Ginley B, Jen KY, Han SS, Rodrigues L, Jain S, Fogo AB, Zuckerman J, Walavalkar V, Miecznikowski JC, Wen Y, Yen F, Yun D, Moon KC, Rosenberg A, Parikh C, Sarder P. Automated computational detection of interstitial fibrosis, tubular atrophy, and glomerulosclerosis. J Am Soc Nephrol 2021;32(4):837–850.
26. Bercea CI, Wiestler B, Rueckert D, Albarqouni S. FedDis: disentangled federated learning for unsupervised brain pathology segmentation. arXiv preprint 2021.arXiv:2103.03705.
27. Brisimi TS, et al. Federated learning of predictive models from federated electronic health records. Int J Med Inform 2018;112:59–67.
28. Boughorbel S, et al. Federated uncertainty-aware learning for distributed hospital ehr data. arXiv preprint 2019.arXiv:1910.12191.
29. Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning for healthcare informatics. J Healthcare Inform Res 2021;5:1-19.
30. Lu MY, Chen RJ, Kong D, Lipkova J, Singh R, Williamson DFK, Chen TY, Mahmood F. Federated learning for computational pathology on gigapixel whole slide images. arXiv preprint 2020;76:1-13.arXiv:2009.10190.
31. Abadi M., Chu A., Goodfellow I., McMahan H.B., Mironov I., Talwar K., Zhang L. Deep Learning with Differential Privacy, Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 308-318.
32. Yi L., Zhang J., Zhang R., Shi J., Wang G., Liu, X., SU-Net: An Efficient Encoder-Decoder Model of Federated Learning for Brain Tumor Segmentation, International Conference on Artificial Neural Networks. 761-773 (Springer).
33. Michieli U, Ozay M. Prototype guided federated learning of visual feature representations. arXiv preprint 2021.arXiv:2105.08982.
34. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (voc) challenge. Int J Comput Vision 2010;88:303–338.
35. Lalitha A, Kilinc OC, Javidi T, Koushanfar F. Peer-to-peer federated learning on graphs. arXiv preprint 2019.arXiv:1901.11173.
36. Shastry KL, Manamohan S, Mukherjee S, Garg V, Sarveswara R, Händler K, Pickkers P, Aziz NA, Ktena S, Siever C, Kraut M, Desai M, Monnet B, Saridaki M, Siegel CM, Drews A, Nuesch-Germano M, Theis H, Netea MG, Theis F, Aschenbrenner AC, Ulas T, Breteler MMB, Giamarellos-Bourboulis EJ, Kox M, Becker M, Cheran S, Woodacre MS, Goh EL, Schultze JL, German COVID-19 OMICS Initiative (DeCOI). Swarm Learning as a privacy-preserving machine learning approach for disease classification. bioRxiv 2020:64. https://doi.org/10.1101/2020.06.25.171009.
37. Masse M. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces.* O'Reilly Media, Inc. 2011.
38. Chandra RV, Varanasi BS. *Python requests essentials.* Packt Publishing Ltd. 2015.
39. Solem A. Celery - Distributed Task Queue. https://docs.celeryproject.org/en/stable/> 2021.
40. VMware. RabbitMQ. https://www.rabbitmq.com/> 2021.
41. Sutskever, I., Martens, J., Dahl, G. & Hinton, G. International conference on machine learning. 1139-1147 (PMLR).