The Journal of Open Source Software

MeshFL: A Decentralized MeshNet Framework for 3D Brain MRI Segmentation

Mohamed Masoud [1¶], **Pratyush Reddy** [1], **Sandeep Panta** [1], **and Sergey Plis** [1,2]

1 Tri-institutional Center for Translational Research in Neuroimaging and Data Science (TReNDS), Georgia State University, Georgia Institute of Technology, Emory University, Atlanta, United States of America **2** Department of Computer Science, Georgia State University, Atlanta, United States of America ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Chris Vernon

Reviewers:

- @donishadsmith
- @paquiteau

Submitted: 23 January 2025
Published: unpublished

Summary

Advances in federated learning paved the way for privacy-preserving collaborative training of machine learning models on decentralized datasets. This is particularly useful in neuroimaging, where sensitive data, such as brain MRI scans, cannot be easily shared across institutions. MeshFL ("MeshFL," 2025) is an open-source framework designed to facilitate distributed training of deep learning models for 3D brain MRI segmentation while maintaining data privacy. Built upon NVFlare (NVIDIA, 2023), MeshFL employs federated learning principles to train MeshNet models (Fedorov et al., 2017) across multiple data sites, enabling high-accuracy segmentation of white and gray matter regions. With Dice scores of ~0.92 for training and ~0.9 for validation, MeshFL demonstrates that decentralized training can achieve performance comparable to centralized setups.

Statement of Need

In neuroimaging, collaborative machine learning is often hindered by the sensitive nature of patient data and the computational demands of training large 3D models. Traditional centralized learning approaches require aggregating data in one location, which is impractical for datasets governed by strict privacy laws. Federated learning addresses this limitation by enabling model training without sharing raw data between sites (McMahan et al., 2017),(Rieke et al., 2020).

The model choice is determined by the need to limit the bandwidth and reduce the possibility of data leakage through the gradients shared during training. MeshNet's parameter size in our use case is 22.2 KB, making it a lightweight and efficient choice for federated learning.

Existing federated learning frameworks often lack specific adaptations for neuroimaging tasks. MeshFL fills this gap by offering:

- A tailored framework for 3D brain MRI segmentation using the MeshNet model.
- Integration with NVFlare for federated training workflows (NVIDIA, 2023).
- Support for heterogeneously distributed data across sites.

MeshFL provides an easy-to-use yet robust environment for researchers and clinicians, ensuring high model performance while preserving patient privacy.

Implementation

MeshFL leverages NVFlare to implement federated learning workflows, allowing local sites to independently train the MeshNet model on their data and exchange model updates with a central server as shown in Figure 1.

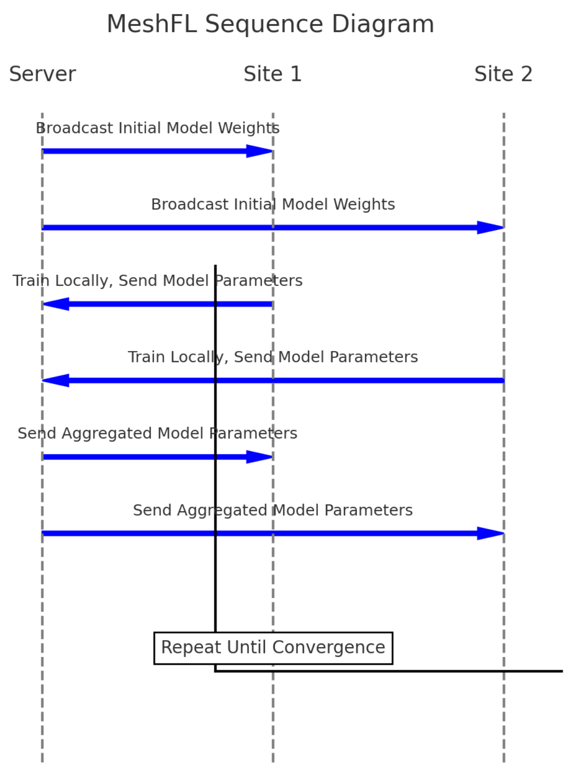

Figure 1: MeshFL Sequence Diagram.

MeshFL key features include:

- **Data Preprocessing:** Automated partitioning of MRI scans into training, validation, and testing sets.
- **Model Training:** The framework utilizes PyTorch for implementing the MeshNet model and optimizing memory usage. Layer checkpointing further reduces memory overhead during training.
- **Aggregation Strategies:** Federated averaging of model weights, where the global model is updated by computing the average of the local weights contributed by each site. Initial model weights are shared across sites for consistent training initialization.
- **Custom Logger:** MeshFL includes a GenericLogger for detailed logging of training progress, gradient application, and Dice score evaluations.
- **Scalability:** Seamless support for multiple sites with varying data distributions and qualities.

The architecture of MeshNet, a 3D convolutional neural network, is optimized for volumetric brain MRI segmentation, employing dilated convolutions to capture contextual information while maintaining a compact parameter set (Yu & Koltun, 2016). A CrossEntropyLoss criterion with class weights addresses class imbalance in the dataset.

MeshFL also integrates a learning rate scheduler to enhance training stability. Using OneCycleLR, the scheduler gradually increases the learning rate during the initial phase of training and decreases it afterward, ensuring convergence without disrupting the learning process. This approach prevents spikes in the learning rate and supports optimal parameter updates.

Validation

The performance of MeshFL was validated using the Mindboggle dataset (Klein & Tourville, 2012) on 15 MRI samples labeled for white and gray matter segmentation. Using Dice coefficient as the evaluation metric and CrossEntropy for loss calculation, MeshFL achieved comparable accuracy to centralized training setups while adhering to federated learning constraints. Benchmarks were conducted with uniformly distributed data across sites, ensuring each site received an equal number of samples for training and validation.

Results demonstrated that MeshFL achieved Dice scores of ~0.92 for training and ~0.9 for validation with robust performance comparable to centralized training Figure 2.

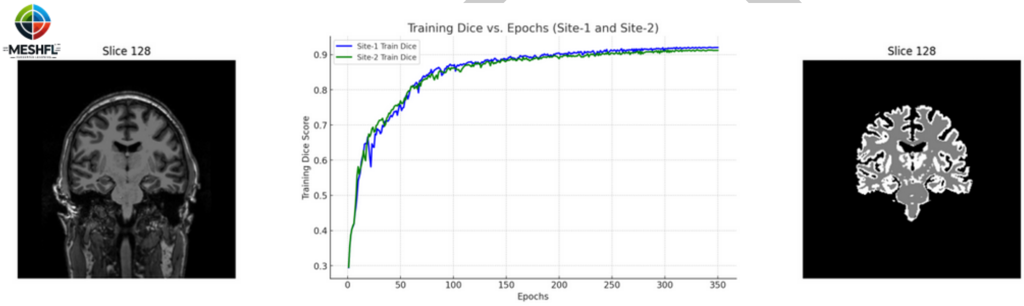

Figure 2: MeshFL Training Performance.

While MeshFL performs volumetric segmentation, slice 128 is shown for illustration, presenting the raw unsegmented slice on the left and the corresponding segmented output on the right, highlighting the segmentation quality achieved.

Code Availability

MeshFL is openly available on GitHub at https://github.com/Mmasoud1/MeshFL. The repository includes documentation, example scripts, and a wiki to guide users through installation and usage. Researchers can reproduce the experiments described here or adapt MeshFL for their applications.

Author Contributions

We describe contributions to this paper using the CRediT taxonomy (Brand et al., 2015). - **Writing – Original Draft:** M.M. - **Writing – Review & Editing:** M.M., S.Panta., and S.Plis. - **Conceptualization and Methodology:** M.M., P.R., and S.Plis. - **Software and Data Curation:** M.M., and P.R. - **Validation:** M.M., S.Plis., and S.Panta. - **Project Administration:** M.M., and S.Panta.

Acknowledgments

This work was funded by the NIH grant R01DA040487. Special thanks to Dylan Martin for his initial demonstration on using NVFlare.

References

Brand, A., Allen, L., Altman, M., Hlava, M., & Scott, J. (2015). Beyond authorship: Attribution, contribution, collaboration, and credit. *Learned Publishing*, *28*(2), 151–155. https://doi.org/10.1087/20150211

Fedorov, A., Johnson, J., Damaraju, E., Ozerin, A., Calhoun, V., & Plis, S. (2017). End-to-end learning of brain tissue segmentation from imperfect labeling. *IEEE International Joint Conference on Neural Networks (IJCNN)*. https://doi.org/10.1109/IJCNN.2017.7966333

Klein, A., & Tourville, J. (2012). Mindboggle: Automated human brain image labeling and morphometry. *Frontiers in Neuroinformatics*, *6*, 2. https://doi.org/10.3389/fninf.2012.00002

McMahan, B., Moore, E., Ramage, D., Hampson, S., & others. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1273–1282. https://arxiv.org/abs/1602.05629

MeshFL. (2025). In *GitHub repository*. GitHub. https://github.com/Mmasoud1/MeshFL

NVIDIA. (2023). NVFlare: NVIDIA federated learning application runtime environment. *GitHub Repository*. https://github.com/NVIDIA/NVFlare

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., & others. (2020). The future of digital health with federated learning. *Npj Digital Medicine*, *3*(1), 119. https://doi.org/10.1038/s41746-020-00323-1

Yu, F., & Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. *arXiv*. https://doi.org/10.48550/arXiv.1511.07122