The Journal of Open Source Software

Brainchop: In-browser MRI volumetric segmentation and rendering

Mohamed Masoud [1¶], **Farfalla Hu** [1,2], **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, GA, USA **2** Georgia State University, Department of Computer Science, Atlanta, GA, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Brainchop brings high fidelity pre-trained deep learning models for volumetric analysis of structural magnetic resonance imaging (MRI) right to the browsers of scientists and clinicians with no requirement on their technical skills in setting up AI solutions. All of this in an extensible open-source framework.

Our tool is the first front-end MRI segmentation tool on the web that supports full brain volumetric processing in a single pass inside a browser. This property is powered by the lightweight and reliable deep learning model Meshnet (Fedorov et al., 2017). Our modified Meshnet model enables voxel-based segmentation of the entire brain at once by using volumetric dilated convolutions (Yu & Koltun, 2016), which leads to increased accuracy with modest computational requirements. High-quality client-side processing solves the privacy problem, as the data does not need to leave the client. Moreover, browser-based implementation can take advantage of available hardware acceleration regardless of the brand or architecture.

Statement of needs

Extracting brain tissue from MRI volumes and segmenting it either into gray and white matter or into a more elaborate brain atlas is an essential step in many brain imaging analysis pipelines. Surgical planning, measuring brain changes, and visualizing its anatomical structures are just a few clinical applications commonly dependent on MRI segmentation. However, not many places in the world have clinicians and scientists who are simultaneously skilled in their domain and are experts in machine learning and computational infrastructure to be able to take advantage of the benefits enabled by AI-assisted neuroimaging. Yet, the value and potential of modern AI-assisted solutions are difficult to overestimate, especially for specialists in rural areas and developing countries. Web applications could address the many challenges faced by practitioners, but the constrained browser environment is a challenge for high-quality AI models. The common fall-back to server-side processing violates data privacy and usually limits the portability by locking-in on a single type of hardware to run AI on servers. We developed brainchop to address these challenges by creating in-browser machine learning pipelines for volumetric neuroimaging. The high accessibility, scalability, low latency, ease of use, lack of installation requirements, and cross-platform operation are just a few of the unique and enabling features that brainchop can provide while also preserving end-user data privacy.

Pipeline

In order to deploy the PyTorch MeshNet model in the browser, there is a need to convert it first to a workable tensorflow.js (tfjs) (Smilkov et al., 2019). The tool has a pre-processing pipeline, full-volume and sub-volume inference options, 3D input/output rendering, and post-processing capability as illustrated in Figure 1 for the brainchop high-level architecture.

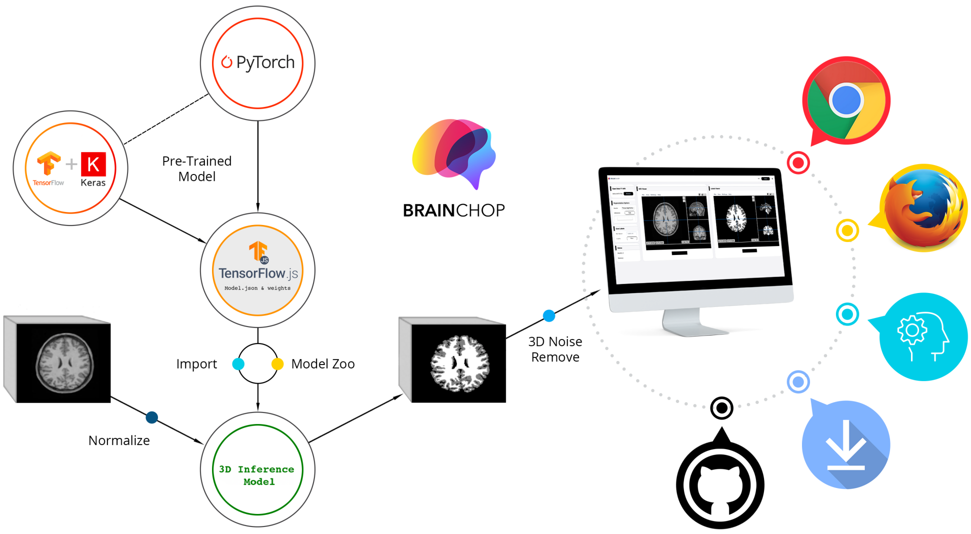

Figure 1: Brainchop high-level architecture.

Preprocessing

Brainchop is designed to support T1 weighted MRI volume segmentation. The input is read in Nifti format ("NIfTI Reader," 2021). T1 image needs to be in shape 256x256x256, scaled, and resampled to 1mm isotropic voxels as a preprocessing step for proper results. This preprocessing can be made in brainchop by using mri_convert.js which uses pyodide (Pyodide dev. team, 2022) to deploy the conform function used by FastSurfer (Henschel et al., 2020) for reshaping, scaling, and re-sampling MRI T1 raw image data as shown in Figure 2 (a).

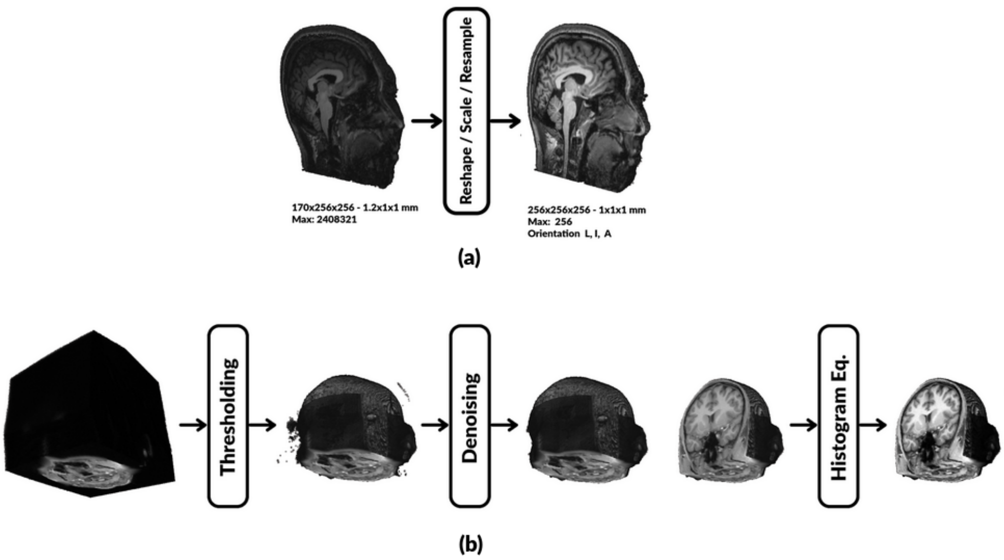

Figure 2: Brainchop preprocessing pipeline (a) Conform operation (b) MRI enhancement operations.

The rest of the preprocessing pipeline is to remove input noisy voxels and enhance input volume intensities as in **??** for improving the segmentation accuracy Figure 2 (b). In addition, brainchop also supports MRI tissue cropping to speed up the inference process and lowering memory use.

Inference Model

The advantage of MeshNet small size is due to its simple architecture and using dilated convolution in which a typical model for the segmentation task can be constructed with few layers as shown in Figure 3.

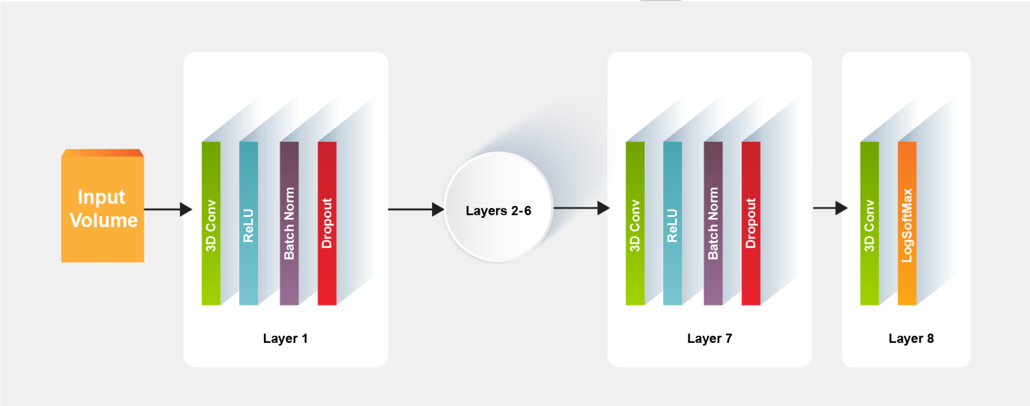

Figure 3: MeshNet architecture.

While MeshNet Model has fewer parameters compared to the classical segmentation model U-Net, it is also can achieve a competitive DICE score as shown in Table 1.

Table 1: Segmentation models performance.

Model	Inference Speed	Model Size	Macro DICE
MeshNet GMWM	116 subvolumes/sec	.89 mb	0.96
U-Net GMWM	13 subvolumes/sec	288 mb	0.96
MeshNet GMWM (full brain model)	0.001 sec/volume	0.022 mb	0.96

Results

Multiple pre-trained models are available with brainchop for full-volume and sub-volume inference including brain masking, gray matter white matter (GMWM) segmentation models, in addition to brain atlas models for 50 cortical regions and 104 cortical and sub-cortical structures as shown in Figure 4.

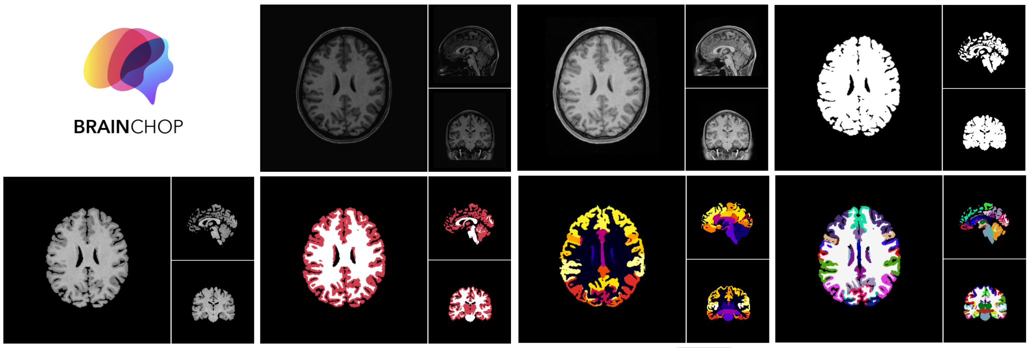

Figure 4: Brainchop outputs.

Normally 3D noisy regions may result from the inference process due to possible bias, variance and irreducible error (e.g. noise with data). To remove these noisy volumes we designed a 3D connected components algorithm to filter out those noisy regions.

Papaya (Papaya dev. team, 2019) viewers is used to visualize the input and output images, and a composite operation also provided to subjectively verify the output image accuracy comparing to the input.

Also, brainchop supports 3D real-time rendering of the input and output volume by using Three.js (Cabello & et, 2010) with capability of Region of Interest (ROI) selection as shown in Figure 5.

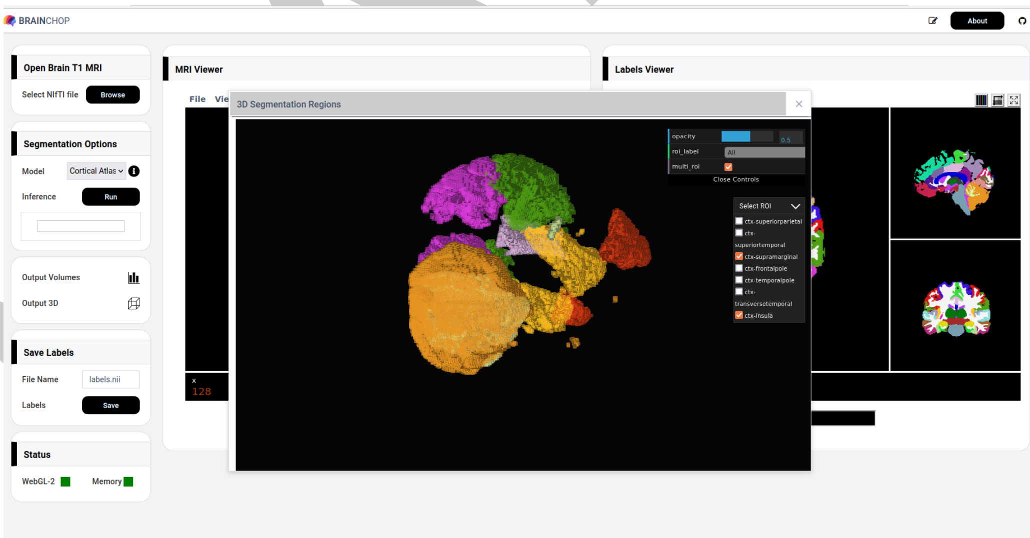

Figure 5: Brainchop rendering segmentation output in 3D.

Detailed step-by-step documentation is provided alongside the source code. Built-in models also are provided with brainchop live demo.

Acknowledgments

This work was funded by the NIH grant RF1MH121885. Additional support from NIH R01MH123610, R01EB006841 and NSF 2112455.

References

Cabello, R., & et, al. (2010). Three.js. In *GitHub repository*. GitHub. https://github.com/mrdoob/three.js

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

Henschel, L., Conjeti, S., Estrada, S., Diers, K., Fischl, B., & Reuter, M. (2020). FastSurfer - a fast and accurate deep learning based neuroimaging pipeline. *Neuroimage*. https://doi.org/10.1016/j.neuroimage.2020.117012

NIfTI reader. (2021). In *GitHub repository*. GitHub. https://github.com/rii-mango/NIFTI-Reader-JS

Papaya dev. team. (2019). Papaya. In *GitHub repository*. GitHub. https://github.com/rii-mango/Papaya

Pyodide dev. team. (2022). *Pyodide* (Version 0.20.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.6430433

Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., & et, al. (2019). TensorFlow.js: Machine learning for the web and beyond. *arXiv*. https://doi.org/10.48550/arXiv:1901.05350v2

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