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Published: unpublished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 is a web-based tool that allows scientists and clinicians to perform volumetric analysis of structural Magnetic Resonance Imaging (MRI) using pre-trained deep learning models, without the need for technical expertise or setup of AI solutions. It is the first browser-based front-end MRI segmentation tool that can process full brain volumes in a single pass, using the lightweight and reliable Meshnet model (Fedorov et al., 2017) modified to incorporate volumetric dilated convolutions (Yu & Koltun, 2016) for increased accuracy and modest computational requirements. In addition to offering extensibility through an open-source framework, Brainchop also addresses privacy concerns by performing client-side processing and leveraging hardware acceleration across different brands and architectures.

Statement of needs

Accurate segmentation of brain tissue from MRI volumes is a crucial step in various brain imaging analysis pipelines, with applications ranging from surgical planning and measurement of brain changes to visualization of anatomical structures. However, few professionals possess both domain expertise and machine learning skills, as well as the necessary computational infrastructure, to take advantage of the benefits of artificial intelligence (AI)-assisted neuroimaging. This limitation is particularly pronounced in rural areas and developing countries. To address these challenges, we developed brainchop, an in-browser machine learning platform for volumetric neuroimaging that offers high accessibility, scalability, low latency, ease of use, lack of installation requirements, and cross-platform operation while also preserving end-user data privacy. Unlike traditional web applications, brainchop operates within the browser environment, avoiding the need for server-side processing and the associated privacy and portability issues.

Pipeline

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

JOSS
The Journal of Open Source Software

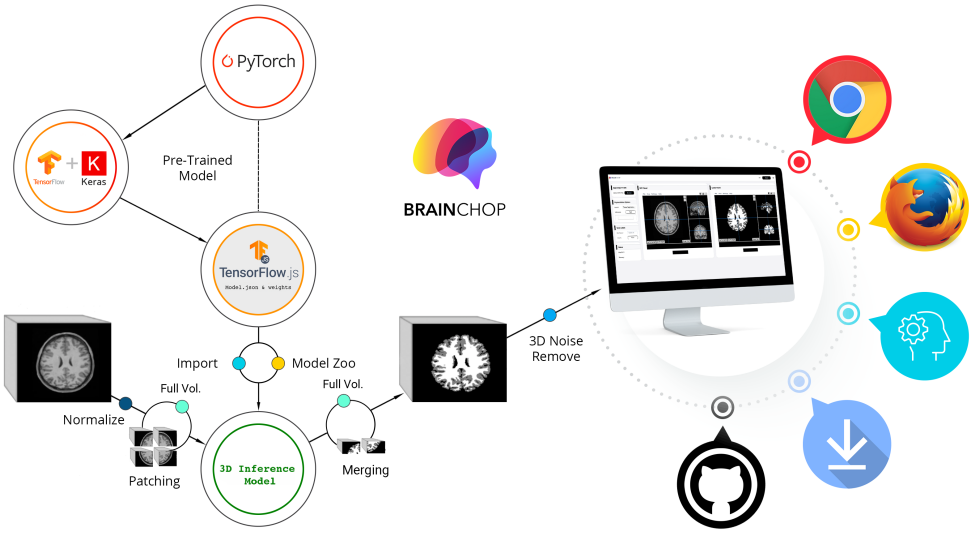

Figure 1: Brainchop high-level architecture.

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).

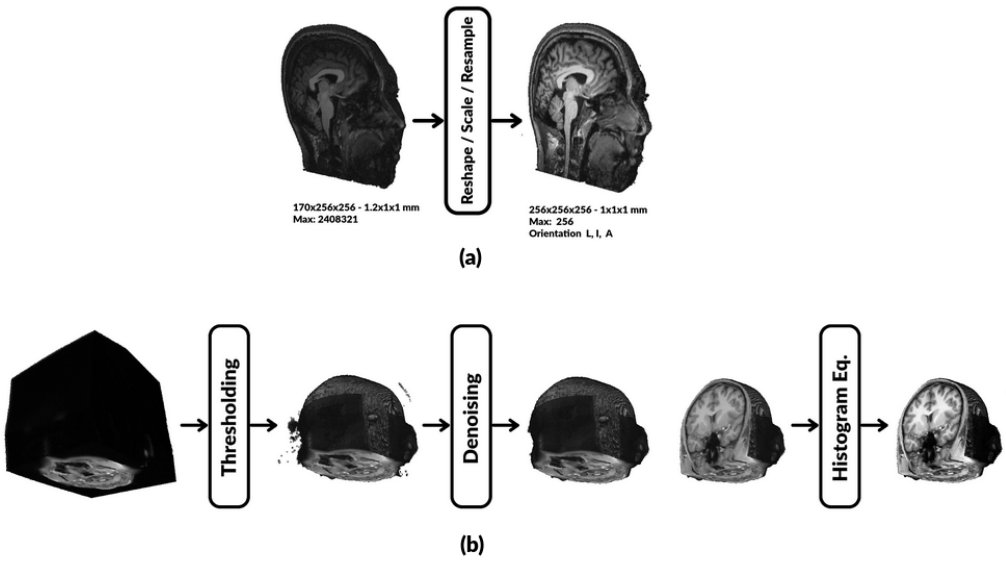

Figure 2: Brainchop preprocessing pipeline (a) Conform operation (b) MRI enhancement operations.

The rest of the preprocessing pipeline is to remove input noisy voxels and enhance input volume intensities to improve 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.

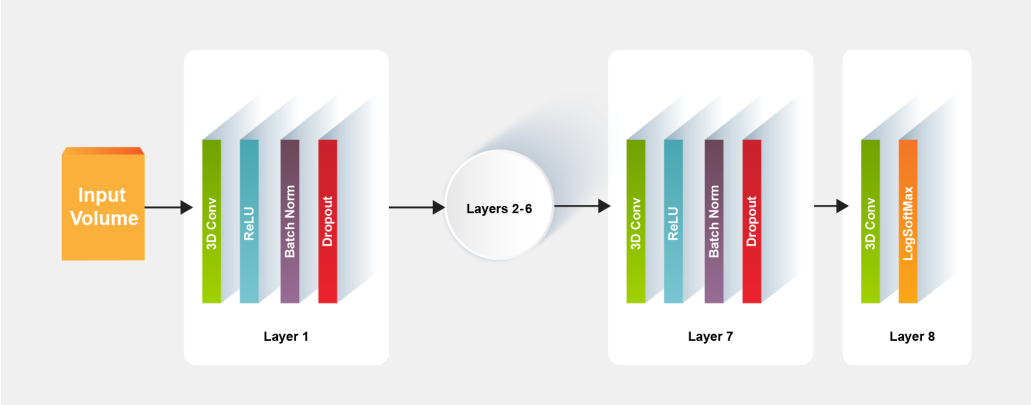

Figure 3: MeshNet architecture.

While MeshNet model has fewer parameters compared to the classical segmentation model U-Net, it 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-volumes 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.

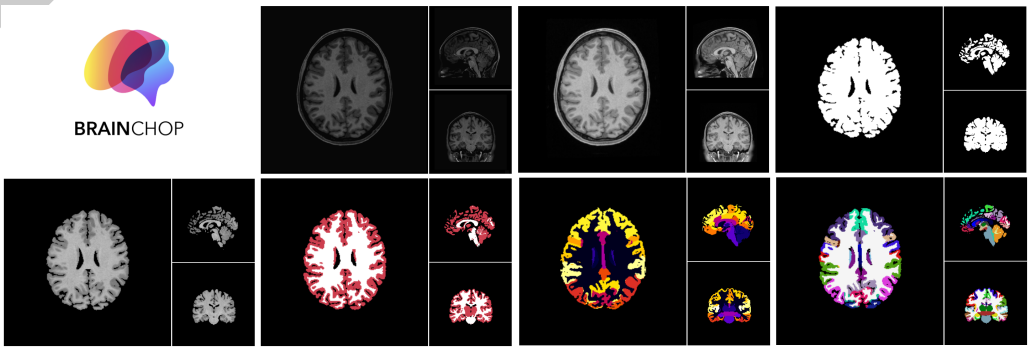

Figure 4: Brainchop outputs.

Normally 3D noisy regions can 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 is 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 the capability of Region of Interest (ROI) selection as shown in Figure 5.

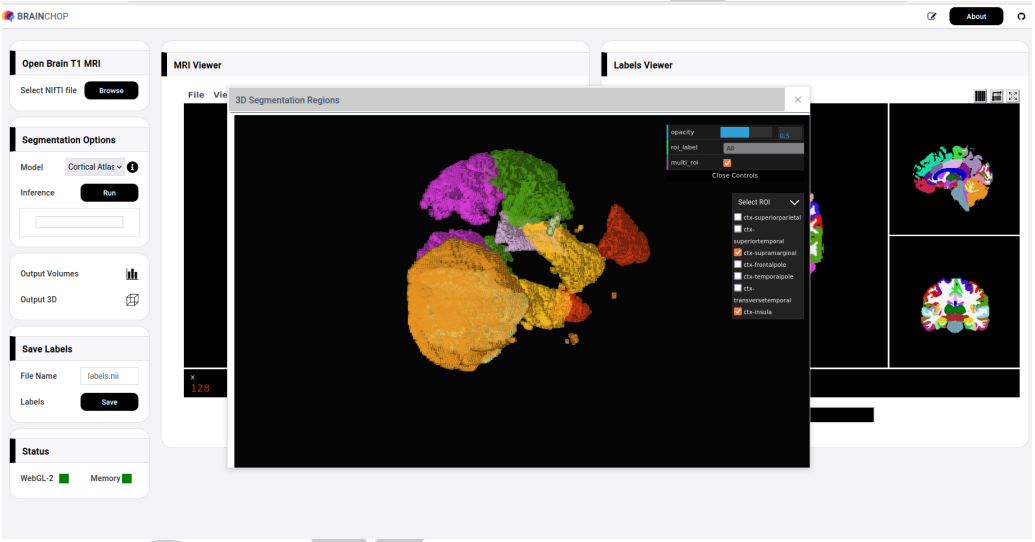

Figure 5: Brainchop rendering segmentation output in 3D.

Code availability

The source code is publicly accessible in a GitHub repository (https://github.com/neuroneural/brainchop) with detailed step-by-step documentation. Built-in models also are provided with brainchop live demo.

Author contributions

We describe contributions to this paper using the CRediT taxonomy (Brand et al., 2015). Writing – Original Draft: M.M., F.H., and S.P.; Writing – Review & Editing: M.M., and S.P.; Conceptualization and methodology: M.M., and S.P.; Software and data curation: M.M., F.H., and S.P.; Validation: M.M., and S.P.; Resources: S.P.; Visualization: M.M., and F.H.; Supervision: S.P.; Project Administration: M.M., and S.P.; Funding Acquisition: S.P.

Acknowledgments

The authors would like to thank Kevin Wang and Alex Fedorov for discussions and pre-trained Meshnet models.

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

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

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). 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.1901.05350

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