Volume Segmantics: A Python Package for Semantic Segmentation of Volumetric Data Using Pre-trained PyTorch Deep Learning Models

Oliver N. F. King[1¶], **Dimitrios Bellos**[2], and **Mark Basham**[1,2]

1 Diamond Light Source Ltd., Harwell Science and Innovation Campus, Didcot, Oxfordshire, UK **2** Rosalind Franklin Institute, Harwell Science and Innovation Campus, Didcot, Oxfordshire, UK ¶ Corresponding author

DOI: 10.21105/joss.04691

Software

- Review
- Repository
- Archive

Editor: Øystein Sørensen

Reviewers:

- @jingpengw
- @estenhl

Submitted: 03 August 2022
Published: 05 October 2022

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

Segmentation of 3-dimensional (3D, volumetric) images is a widely used technique that allows interpretation and quantification of experimental data collected using a number of techniques (for example, Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Electron Tomography (ET)). Although the idea of semantic segmentation is a relatively simple one, giving each pixel a label that defines what it represents (e.g cranial bone versus brain tissue); due to the subjective and laborious nature of the manual labelling task coupled with the huge size of the data (multi-GB files containing billions of pixels) this process is often a bottleneck in imaging workflows. In recent years, deep learning has brought models capable of fast and accurate interpretation of image data into the toolbox available to scientists. These models are often trained on large image datasets that have been annotated at great expense. In many cases however, scientists working on novel samples and using new imaging techniques do not yet have access to large stores of annotated data. To overcome this issue, simple software tools that allow the scientific community to create segmentation models using relatively small amounts of training data are required. `Volume Segmantics` is a Python package that provides a command line interface (CLI) as well as an Application Programming Interface (API) for training 2-dimensional (2D) PyTorch (Paszke et al., 2019) deep learning models on small amounts of annotated 3D image data. The package also enables applying these models to new (often much larger) 3D datasets to speed up the process of semantic segmentation.

Statement of need

`Volume Segmantics` harnesses the availability of 2-dimensional encoders which have been pre-trained on huge databases such as ImageNet (Russakovsky et al., 2015). This provides two main advantages, namely (i) it reduces the time and resource taken to train models — only fine-tuning is required and (ii) it prevents over-fitting the models when training on small datasets. These models of various architectures are included from the `segmentation-models-pytorch` repository (Yakubovskiy, 2020). In order to increase the accuracy of the models, augmentations of the data are made during training, both via 'slicing' the 3D data in planes perpendicular to the three orthogonal axes $((x, y), (x, z), (y, z))$ and by using the library `Albumentations` (Buslaev et al., 2020). Additionally, user configuration for training is kept to a minimum by starting with a reliable default set of parameters and by automatically choosing the model learning rate. If adjustments to model architecture, encoder type, loss function or training epochs are required; these can be made by editing a YAML file.

Even though these 2D models are quicker to train and require fewer computational resources than their 3D counterparts (Alvarez-Borges et al., 2022), when predicting a segmentation for a volume, the lack of 3D context available to these models can lead to striping artifacts in the 3D output, especially when viewed in planes other than the one used for prediction. To overcome this, a multi-axis prediction method is used, and the multiple predictions are merged by using maximum probability voting. It is hoped that in the future other merging techniques will be included such as fusion models (Perslev et al., 2019). A schematic of the training and prediction processes performed by the `Volume Segmantics` package can be seen in Figure 1.

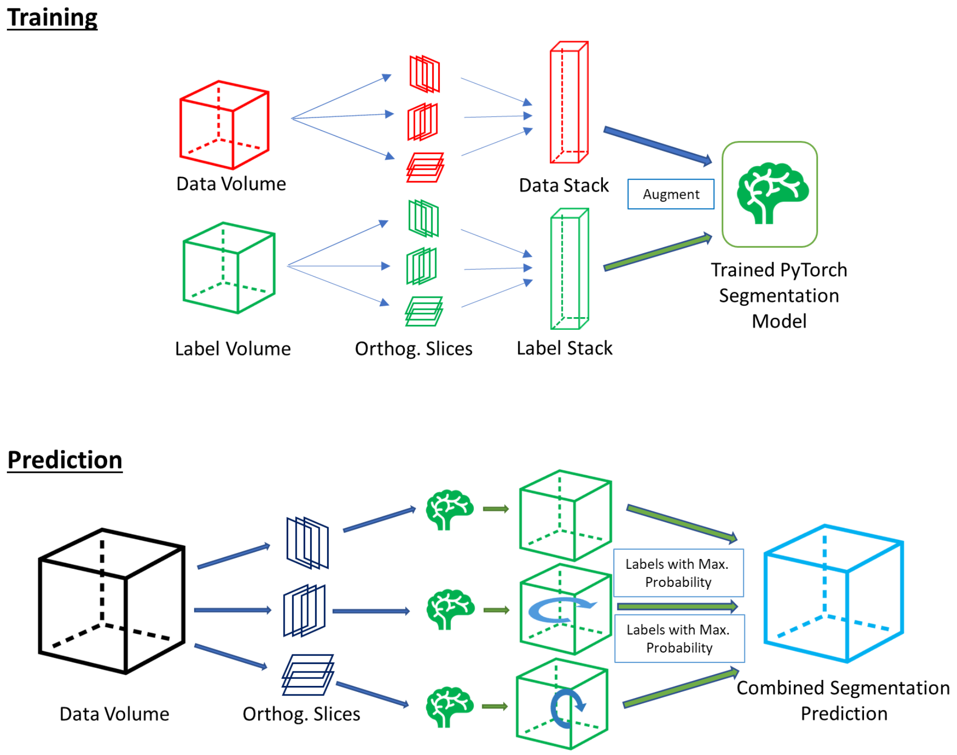
Training
Data Volume
Data Stack
Augment
Label Volume
Orthog. Slices
Label Stack
Trained PyTorch Segmentation Model
Prediction
Data Volume
Orthog. Slices
Labels with Max. Probability
Labels with Max. Probability
Combined Segmentation Prediction

Figure 1: A schematic diagram showing the model training and segmentation prediction processes performed by the `Volume Segmantics` package.

State of the field

Currently there are a number of other software implementations available for segmentation of 3D data. Some of these also use 2D networks and combine prediction outputs to 3D, for example the `Multi-Planar U-Net` package (Perslev & Igel, 2019) and the `CTSegNet` package (Tekawade & Igel, 2020). However, neither of these packages allows the use of pre-trained encoders or multiple model architectures. In addition, the general purpose `pytorch-3dunet` package (Wolny, 2019) exists to allow training a 3D U-Net on image data, again without the time and resource advantages of pre-trained 2D networks.

In the field of connectomics, several packages (Lee & Turner, 2018; Lin et al., 2021; Urakubo et al., 2019; Wu, 2021) enable the segmentation of structures within the brain, often from electron microscopy data, these could in principle be used in a subject and method-agnostic manner similarly to `Volume Segmantics`. One member of this set, the package `pytorch-connectomics`, (Lin et al., 2019) allows training of 2D and 3D networks for segmentation of 3D image data as well as customisable strategies for data augmentation and multi-task and semi-supervised learning. Despite the versatility of this software, its deliberate focus on connectomics which is essential for effectiveness in this complex field, means that there are added levels of complexity for the generalist user. This specialisation also means that there is only one pre-trained 2D model architecture available, and some of the configuration and command-line options are context specific.

Real-world usage

During development of `Volume Segmantics`, the software was used to fine-tune pre-trained U-Net models on small amounts of annotated data in order to investigate the structures that interface maternal and fetal blood volumes in human placental tissue (Tun et al., 2021). In this study, expert annotation of volumes of size 256^3 and 384^3 were sufficient to create two models that gave accurate segmentation of two much larger synchrotron X-ray CT (SXCT) datasets ($2520 \times 2520 \times 2120$ pixels). In a completely different context, SXCT datasets were collected on a soil system in which methane bubbles were forming in brine amongst sand particles. The utility of a pre-trained 2D U-Net was investigated to segment these variable-contrast image volumes in comparison to a 3D U-Net with no prior training (Alvarez-Borges et al., 2022). In this case, the training data ranged in size from 384^3 pixels to 572^3. As well as requiring less time to train than a 3D U-Net, the pre-trained 2D network provided more accurate segmentation results.

The API

The API provided with the package allows segmentation models to be trained and used in other contexts. For example, `Volume Segmantics` has recently been integrated into `SuRVoS2` (Pennington et al., 2018, 2022), a client-server application with a GUI for annotating volumetric data. SuRVoS2 can be used to create the initial small region of interest (ROI) labels needed by `Volume Segmantics`, this is achieved by using machine learning models (e.g. random forests) which are trained through 'scribbles' drawn on the data. It is hoped that scientists using our synchrotron facility and beyond will be able to train and use their own deep learning models using this interface to the library. These models can then be used to segment data during their time using the synchrotron and also when back at their home institution. In addition, it is hoped that the scientific community will use and extend `Volume Segmantics` for their own purposes.

Acknowledgements

We would like to acknowledge helpful discussions with Avery Pennington, Sharif Ahmed, Fernando Alvarez-Borges and Michele Darrow during the development of this project. Additional thanks to Luis Perdigao for inspiration and icons for the schematic figure.

References

Alvarez-Borges, F. J., King, O. N. F., Madhusudhan, B. N., Connolley, T., Basham, M., & Ahmed, S. I. (2022). *U-Net Segmentation Methods for Variable-Contrast XCT Images of Methane-Bearing Sand Using Small Training Datasets*. Earth; Space Science Open Archive. https://doi.org/10.1002/essoar.10506807.2

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. *Information*, *11*(2). https://doi.org/10.3390/info11020125

Lee, K., & Turner, N. L. (2018). DeepEM. In *GitHub repository*. GitHub. https://github.com/seung-lab/DeepEM

Lin, Z., Lu, Y., Belhamissi, M., Banerjee, A., Lauenburg, L., Swaroop, K. K., Wei, D., & Pfister, H. (2019). PyTorch connectomics. In *GitHub repository*. GitHub. https://github.com/zudi-lin/pytorch_connectomics

Lin, Z., Wei, D., Lichtman, J., & Pfister, H. (2021). PyTorch connectomics: A scalable and flexible segmentation framework for EM connectomics. *arXiv Preprint arXiv:2112.05754*. https://doi.org/10.48550/arXiv.2112.05754

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pennington, A., King, O. N. F., Luengo, I., & Basham, M. (2018). SuRVoS2. In *GitHub repository*. GitHub. https://github.com/DiamondLightSource/SuRVoS2

Pennington, A., King, O. N. F., Tun, W. M., Ho, E. M. L., Luengo, I., Darrow, M. C., & Basham, M. (2022). SuRVoS 2: Accelerating Annotation and Segmentation for Large Volumetric Bioimage Workflows Across Modalities and Scales. *Frontiers in Cell and Developmental Biology*, *10*. https://doi.org/10.3389/fcell.2022.842342

Perslev, M., Dam, E. B., Pai, A., & Igel, C. (2019). One Network to Segment Them All: A General, Lightweight System for Accurate 3D Medical Image Segmentation. In D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, & A. Khan (Eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019* (pp. 30–38). Springer International Publishing. https://doi.org/10.1007/978-3-030-32245-8_4

Perslev, M., & Igel, C. (2019). Multi-planar U-net. In *GitHub repository*. GitHub. https://github.com/perslev/MultiPlanarUNet

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Tekawade, A., & Igel, C. (2020). CTSegNet. In *GitHub repository*. GitHub. https://github.com/aniketkt/CTSegNet

Tun, W. M., Poologasundarampillai, G., Bischof, H., Nye, G., King, O. N. F., Basham, M., Tokudome, Y., Lewis, R. M., Johnstone, E. D., Brownbill, P., Darrow, M., & Chernyavsky, I. L. (2021). A massively multi-scale approach to characterizing tissue architecture by synchrotron micro-CT applied to the human placenta. *Journal of The Royal Society Interface*, *18*(179), 20210140. https://doi.org/10.1098/rsif.2021.0140

Urakubo, H., Bullmann, T., Kubota, Y., Oba, S., & Ishii, S. (2019). UNI-EM: An Environment for Deep Neural Network-Based Automated Segmentation of Neuronal Electron Microscopic Images. *Scientific Reports*, *9*(1), 19413. https://doi.org/10.1038/s41598-019-55431-0

Wolny, A. (2019). Pytorch-3dunet. In *GitHub repository*. GitHub. https://github.com/wolny/pytorch-3dunet

Wu, J. (2021). Neutorch. In *GitHub repository*. GitHub. https://github.com/flatironinstitute/neutorch

Yakubovskiy, P. (2020). Segmentation models pytorch. In *GitHub repository*. GitHub. https://github.com/qubvel/segmentation_models.pytorch