The Journal of Open Source Software

VST: A Python-based deep learning tool for segmenting electron microscopy samples

Yuyao (Daniel) Huang[1], **Nickhil Jadav**[1], **Georgia Rutter**[1], **Mihnea Bostina**[1,2], **Duane Harland**[3], **and Lech Szymanski**[1]

1 University of Otago, New Zealand **2** Okinawa Institute of Science and Technology Graduate University, Japan **3** AgResearch Group, New Zealand Institute for Bioeconomy Science Ltd, New Zealand

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Richard Liu

Reviewers:

- @ibro45
- @ductho-le
- @ClementCaporal

Submitted: 03 December 2025
Published: unpublished

Summary

Volume Segmentation Tool (VST) is a Python based deep learning tool designed specifically to segment three-dimensional VEM biological data without extensive requirements for cross disciplinary knowledge in deep learning. The tool is made accessible through a user-friendly interface with visualisations and a one-click installer.

Recognising the current rapid expansion of the VEM field, we have built VST with flexibility and instance segmentation in mind, hoping to ease and accelerate statistical analysis of large datasets in biological and medical research contexts. VST is composed of two main parts: the PyTorch (Paszke et al., 2019)-based deep learning core that performs semantic/instance segmentation on volumetric grey scale image datasets, and a user interface that operates on top of it, responsible for constructing CLI commands to the core components for tasking. The general pipeline of VST is shown in Figure 1. We had put in efforts to ensure VST could automatically handle issues associated with large dataset sizes, instance segmentation, anisotropic voxels and imbalanced classes.

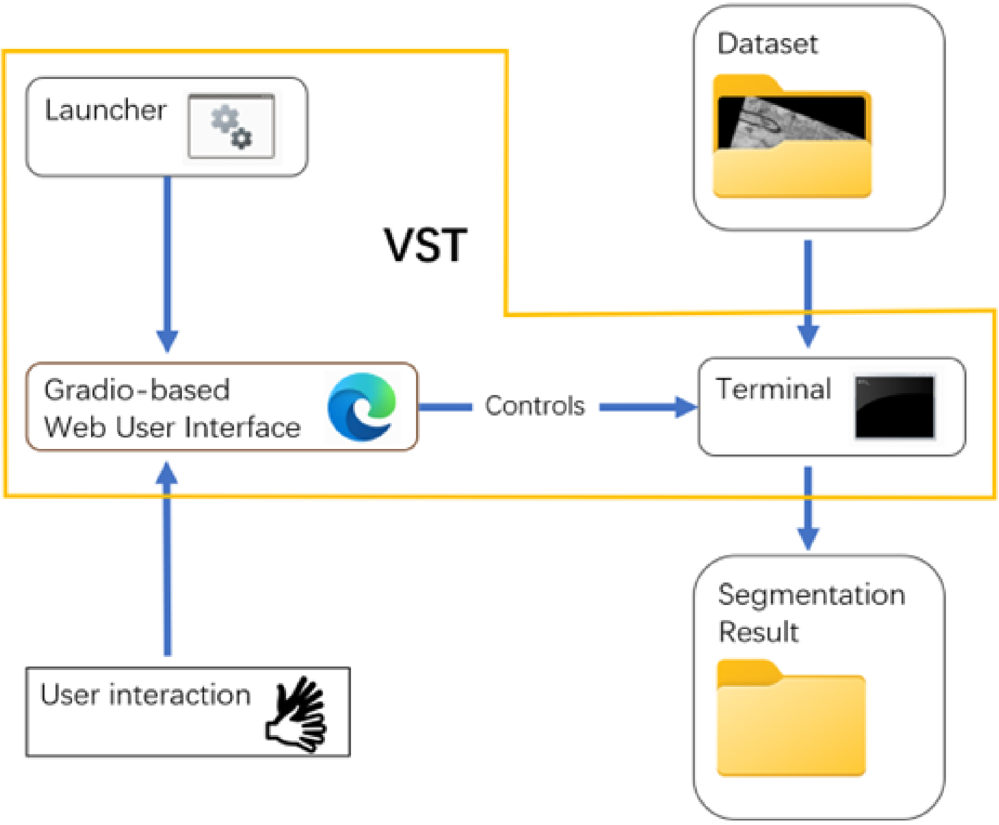

Figure 1: Schematic diagram for VST

Statement of need

Volume Electron Microscopy (VEM) enables the capture of 3D structure beyond planar samples, which is crucial for understanding biological mechanisms. With automation, improved resolution, and increased data storage capacity, VEM has led to an explosion of large three-dimensional datasets. Large datasets offer the opportunity to generate statistical data, but analysing them often requires assigning each voxel (3D pixel) to its corresponding structure, a process known as image segmentation. Manually segmenting hundreds or thousands of image slices is tedious and time-consuming. Computer-aided, especially Machine Learning (ML) based segmentation is now a routinely used method, with Trainable Weka Segmentation (Arganda-Carreras et al., 2017) and Ilastik (Berg et al., 2019) being two leading options. Emerging methods for EM image segmentation are often based on Deep Learning (DL) (Mekuč et al., 2020) because this approach has potential to outperform traditional ML in terms of accuracy and adaptivity (Erickson, 2019; Minaee et al., 2021).

Many earlier DL tools developed are highly specific to single sample types, like in connectomics (Kamnitsas et al., 2017; W. Li et al., 2017), MRI (Milletari et al., 2016) or X-ray tomography (A. Li et al., 2022), they use a subject-optimised design at the cost of adaptability to non-target datasets. Dedicated DL segmentation tools for generalised VEM data are gradually becoming available but each have short-comings. One example, CDeep3M (Haberl et al., 2018), which uses cloud computing. Although easy to use, it was designed for anisotropic data (where the z-resolution is much lower than xy-resolution) which creates limitations when applied to isotropic data (Gallusser et al., 2022). Another example is DeepImageJ (Gómez-de-Mariscal et al., 2021), which runs on local hardware and integrates easily with the ImageJ suit (Schneider et al., 2012). However, it only supports pre-trained models and does not have the functionality to train new ones. ZeroCostDL4Mic (Von Chamier et al., 2021) utilises premade notebooks running on Google Colab, but it requires user interaction during the entire segmentation process, which can take hours and thus is inconvenient. A more recent and advanced example is nnU-Net (Isensee et al., 2021), which auto-configurates itself based on dataset properties and has a good support for volumetric dataset, but it focuses exclusively on semantic segmentation and lacks a user friendly interface.

In short, there is a lack of tools that can handle a wide range of VEM data well for generating both semantic and instance segmentation, while at the same time been easy to use, scalable and can be run locally. Which is what motivated us to develop VST - an easy-to-use and adaptive DL tools specifically optimised for generalised VEM image segmentation.

Software design

The core principles of VST lie in the user-friendliness and scalability. The software comes with a one-click installer and full documentation on all user-accessible features, with the aim to enable accessibility for domain experts without machine learning expertise. In terms of scalability, VST uses Zarr(Abernathey, n.d.), a framework for distributed storage, which allows just-in-time, chunked access for datasets much larger than the user's system memory. Which is a common situation within VEM, where datasets of hundreds or thousands of gigabytes scales are present.

In terms of design, the software provides a graphical interface over a set of scripts handling various aspects of the deep learning and inference workload. The internal training framework utilises PyTorch (Paszke et al., 2019), the interface compiles terminal commands and activates the Python scripts as needed (Figure 1). For the DL model underneath, VST uses a heavily modified U-Net (Ronneberger et al., 2015), a proven DL architecture for segmentation tasks that is known for its fast convergence and adaptability. The size, depth and other details of the model are configured automatically based on the characteristics of the user's dataset.

Much of VST's internal logic is optimised for single-class semantic and instance segmentation and cannot easily be transferred to the multi-class case or 2D image segmentation. This trade-off was made to keep workloads manageable, simplify the codebase, and maximise the ease of use for those with minimal machine learning expertise.

Research impact statement

VST has been used in postgraduate projects at the University of Otago in New Zealand for segmentation of the entire mitochondrial complement of tumorsphere (Jadav et al., 2023), as well as poorly demarked cell remnants within wool fibres. VST's competitive performance to nnU-Net (Huang et al., 2025), an MIT open-source licence, and comprehensive documentation make it ready for use by the wider community.

The graphical user interface

VST's GUI is supported by the Gradio package (Abid et al., 2019) and hosted on the user's browser.

The GUI is divided into three sections: Main, Activations Visualisation and Extras.

The main section (Figure 2) contains settings regarding training and using segmenting networks. Two segmentation modes are supported: semantic segmentation, in which the foreground objects are separated from the background, and instance segmentation, in which individual foreground objects are separated from each other as well. User can either train a new network, load an existing network and use it for predictions on new data, or train one and use it immediately.

Upon training, it automatically opens a TensorBoard interface (Pang et al., 2020) to provides various real time visualisations for the training process.

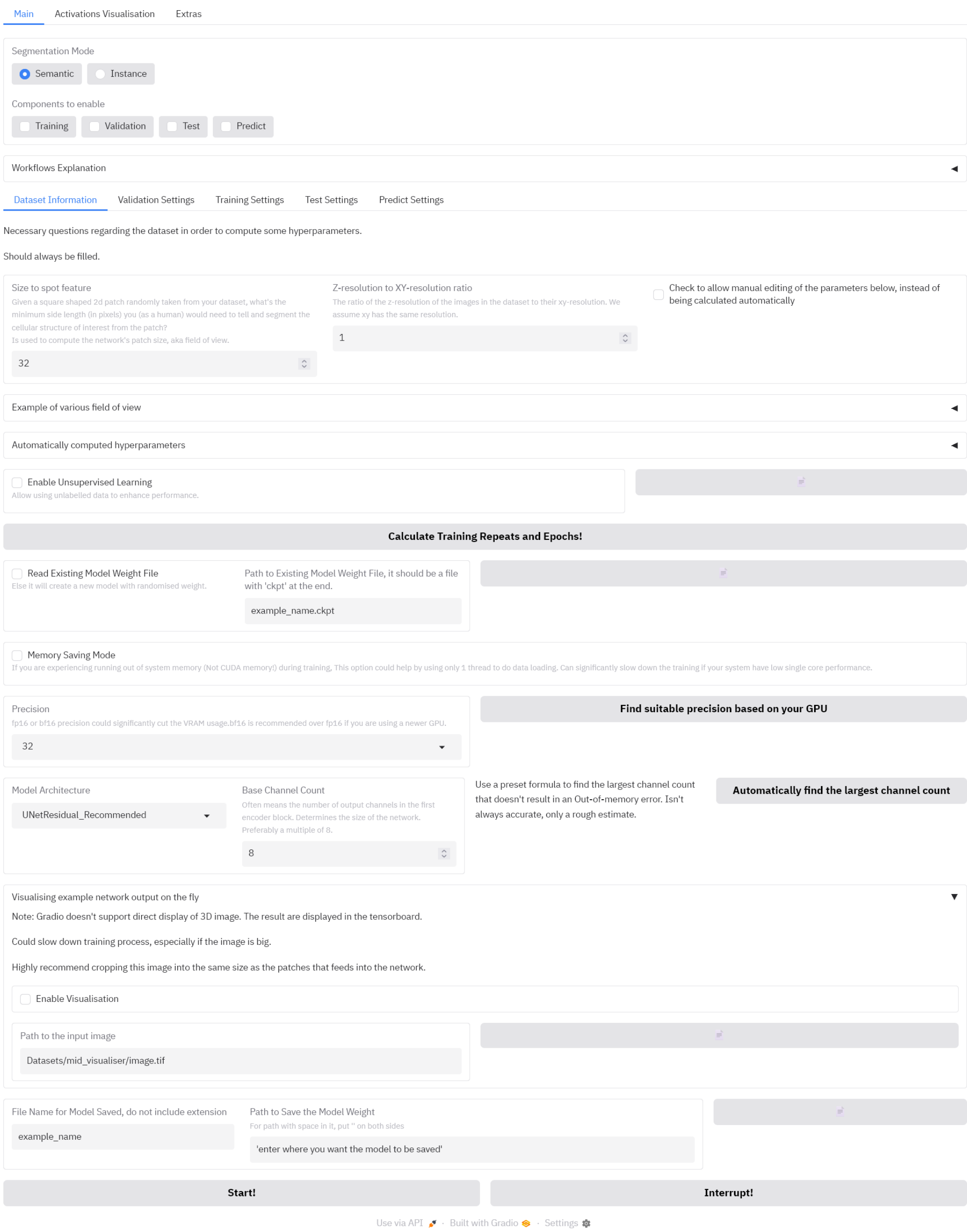

Figure 2: The main interface of VST

The activations visualisation section requires a trained network and an example image. Given that image, it plots the activation across each channel through all layers of the network.

The extra section contains two functionalities: exporting the TensorBoard log to an Excel table, calculating segmentation metrics between (potentially) generated labels and ground truth labels.

AI Usage Disclosure

ChatGPT and DeepSeek were used to generate some Python functions. All generated functions were thoroughly analysed, tested with real-world data, modified and verified to satisfy desired input and output conditions. No generative AI tools were used in the writing of this manuscript, or the preparation of supporting materials.

Acknowledgements

We want to thank to the New Zealand Institute for Bioeconomy Science for their support during the MSc study and University of Otago Postgraduate Publishing Bursary (Master's) for their support with the time of writing.

References

Abernathey, R. (n.d.). *Zarr* (Web Page Jan 20th; Vol. 2026). https://zarr.dev/

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv Preprint arXiv:1906.02569*.

Arganda-Carreras, I., Kaynig, V., Rueden, C., Eliceiri, K. W., Schindelin, J., Cardona, A., & Sebastian Seung, H. (2017). Trainable weka segmentation: A machine learning tool for microscopy pixel classification. *Bioinformatics*, *33*(15), 2424–2426.

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., & others. (2019). Ilastik: Interactive machine learning for (bio) image analysis. *Nature Methods*, *16*(12), 1226–1232.

Erickson, B. J. (2019). Deep learning and machine learning in imaging: Basic principles. In *Artificial intelligence in medical imaging: Opportunities, applications and risks* (pp. 39–46). Springer.

Gallusser, B., Maltese, G., Di Caprio, G., Vadakkan, T. J., Sanyal, A., Somerville, E., Sahasrabudhe, M., O'connor, J., Weigert, M., & Kirchhausen, T. (2022). Deep neural network automated segmentation of cellular structures in volume electron microscopy. *Journal of Cell Biology*, *222*(2), e202208005.

Gómez-de-Mariscal, E., García-López-de-Haro, C., Ouyang, W., Donati, L., Lundberg, E., Unser, M., Muñoz-Barrutia, A., & Sage, D. (2021). DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nature Methods*, *18*(10), 1192–1195.

Haberl, M. G., Churas, C., Tindall, L., Boassa, D., Phan, S., Bushong, E. A., Madany, M., Akay, R., Deerinck, T. J., Peltier, S. T., & others. (2018). CDeep3M—plug-and-play cloud-based deep learning for image segmentation. *Nature Methods*, *15*(9), 677–680.

Huang, Y., Jadav, N., Rutter, G., Szymanski, L., Bostina, M., & Harland, D. P. (2025). A generalist deep-learning volume segmentation tool for volume electron microscopy of biological samples. *Journal of Structural Biology*, *217*(3), 108214.

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, *18*(2), 203–211.

Jadav, N., Velamoor, S., Huang, D., Cassin, L., Hazelton, N., Eruera, A.-R., Burga, L. N., & Bostina, M. (2023). Beyond the surface: Investigation of tumorsphere morphology using volume electron microscopy. *Journal of Structural Biology*, *215*(4), 108035.

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., & Glocker, B. (2017). Efficient multi-scale 3D CNN with fully connected

CRF for accurate brain lesion segmentation. *Medical Image Analysis*, *36*, 61–78.

Li, A., Zhang, X., Singla, J., White, K., Loconte, V., Hu, C., Zhang, C., Li, S., Li, W., Francis, J. P., & others. (2022). Auto-segmentation and time-dependent systematic analysis of mesoscale cellular structure in β-cells during insulin secretion. *Plos One*, *17*(3), e0265567.

Li, W., Wang, G., Fidon, L., Ourselin, S., Cardoso, M. J., & Vercauteren, T. (2017). On the compactness, efficiency, and representation of 3D convolutional networks: Brain parcellation as a pretext task. *International Conference on Information Processing in Medical Imaging*, 348–360.

Mekuč, M. Ž., Bohak, C., Hudoklin, S., Kim, B. H., Kim, M. Y., Marolt, M., & others. (2020). Automatic segmentation of mitochondria and endolysosomes in volumetric electron microscopy data. *Computers in Biology and Medicine*, *119*, 103693.

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, 565–571.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(7), 3523–3542.

Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, *45*(2), 227–248.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & others. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241.

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH image to ImageJ: 25 years of image analysis. *Nature Methods*, *9*(7), 671–675.

Von Chamier, L., Laine, R. F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., Lerche, M., Hernández-Pérez, S., Mattila, P. K., Karinou, E., & others. (2021). Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nature Communications*, *12*(1), 2276.