# SAMBA: A Trainable Segmentation Web-App with Smart Labelling

**Ronan Docherty**[1,2], **Isaac Squires**[2], **Antonis Vamvakeros**[2], and **Samuel J. Cooper**[2¶]

**1** Department of Materials, Imperial College London, London SW7 2AZ, United Kingdom **2** Dyson School of Design Engineering, Imperial College London, London SW7 2DB, United Kingdom ¶ Corresponding author

**DOI:** 10.21105/joss.06159

**Software**

- Review
- Repository
- Archive

**Editor:** Kevin M. Moerman

**Reviewers:**

- @pchlap
- @jingpengw

**Submitted:** 02 November 2023
**Published:** 05 June 2024

## Summary

Segmentation is the assigning of a semantic class to every pixel in an image and is a prerequisite for various statistical analysis tasks in materials science, like phase quantification, physics simulations or morphological characterisation. The wide range of length scales, imaging techniques and materials studied in materials science means any segmentation algorithm must generalise to unseen data and support abstract, user-defined semantic classes. Trainable segmentation is a popular interactive segmentation paradigm where a classifier is trained to map from image features to user drawn labels. `SAMBA` is a trainable segmentation tool that uses Meta's Segment Anything Model (SAM) for fast, high-quality label suggestions and a random forest classifier for robust, generalisable segmentations. It is accessible in the browser (https://www.sambasegment.com/), without the need to download any external dependencies. The segmentation backend is run in the cloud, so does not require the user to have powerful hardware.

## Statement of need

Trainable segmentation tools like ilastik (Berg et al., 2019) and Trainable Weka Segmentation (Arganda-Carreras et al., 2017) are popular options for image segmentation in biology and materials science due to their flexibility. They apply common image processing operations like blurs, edge detection, and texture filters to create a feature stack, then train a classifier (usually a random forest ensemble) to map from the stack to user drawn labels on a pixelwise basis. ilastix offers fast segmentations and a responsive live-view, whereas Trainable Weka Segmentation is part of and interoperates with the FIJI ImageJ library (Schindelin et al., 2012). Other trainable segmenters are based on the napari (Ahlers et al., 2023) image viewer, including napari-feature-classifier (Luethi & Hess, n.d.) and the GPU-accelerated napari-apoc (Haase et al., 2023).

Accurate, representative labels are important for segmentation quality, but tooling is limited to polygons and/or pixel brushes which can make labelling boundaries time-consuming. `SAMBA` (Segment Anything Model Based App) uses the recent object-detecting foundation model Segment Anything Model (Kirillov et al., 2023) as a 'Smart Labelling' tool, which offers suggestions for associated objects as the cursor is moved which can then be added as labels. This 'Smart Labelling' can quickly pick out portions of the same phase and is resilient against noise, varying exposure, and certain types of artefacts *etc.* SAM has been applied to cell microscopy annotation and segmentation (Archit et al., 2023), but not to multi-phase segmentation.

`SAMBA` is also unique in being fully web-based, meaning it can be used without downloading a large binary, installing libraries or requiring significant local computational power. An associated gallery page allows users to contribute to and view an open dataset of micrographs, labels, segmentations and experimental metadata. It is designed to be usable by researchers regardless of prior programming or image analysis experience.

## Design and usage

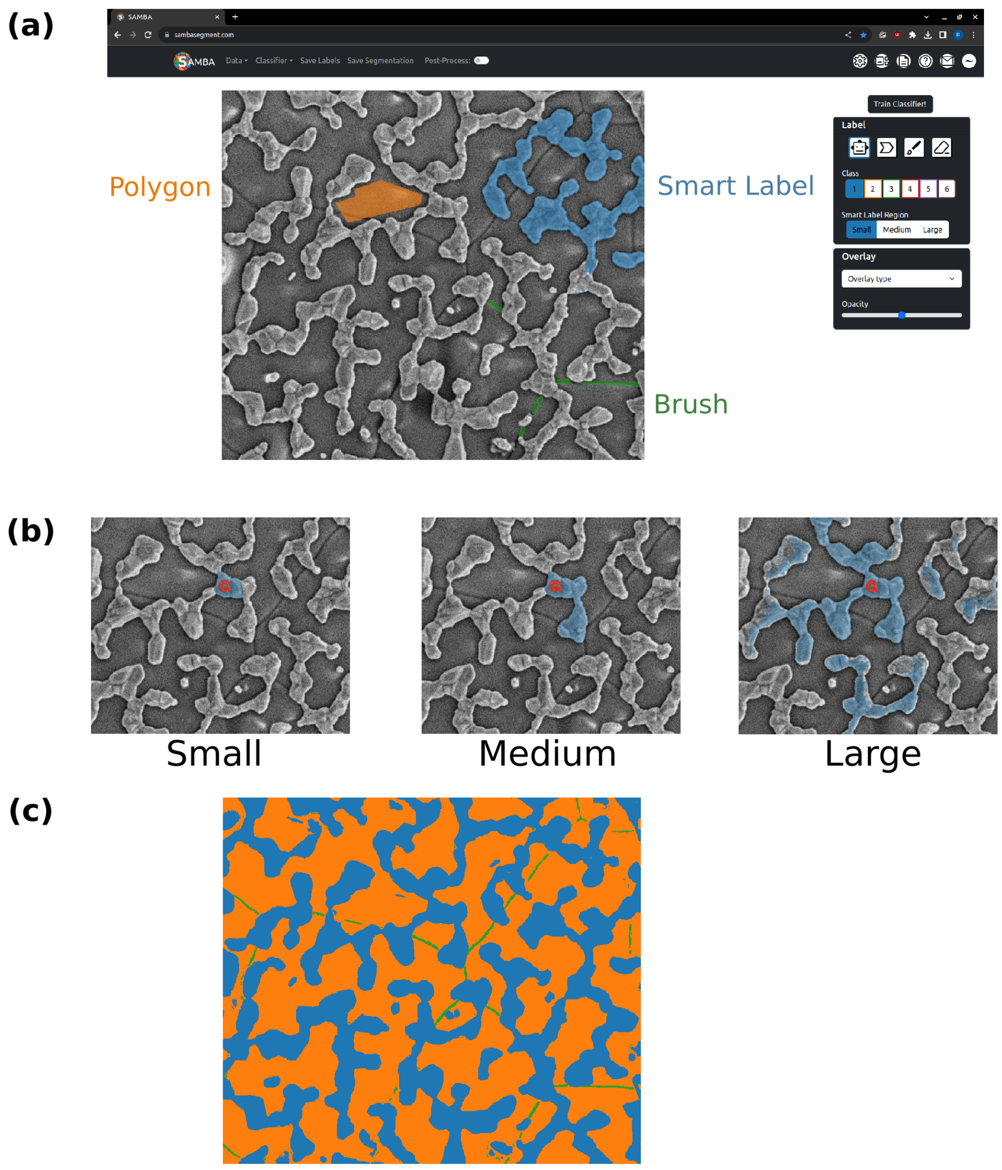

**Figure 1: (a)** screenshot of the SAMBA website, displaying the different labelling options including SAM powered 'Smart Labelling'. **(b)** shows how changing the Smart Label region sizes affects the suggested label at the same mouse position (red), giving the user the flexibility to focus on different length scales. **(c)** an example output segmentation of the tool, which can be saved as `.tiff` for later analysis.

SAM is a *promptable*, open-source macroscopic object segmentation model, trained on 1 billion masks over 11 million images. It feeds a prompt (click, bounding box, mask or text) alongside an embedding of an image generated by a pretrained autoencoder (Dehghani et al., 2023) through a lightweight decoder network to produce a segmentation for the prompt (*i.e,* the object at the mouse cursor). The decoder is lightweight enough that it can be run in the browser in real-time when exported via the `ONNX` framework (Bai et al., 2019), though the embedding must still be generated by a large autoencoder model running on a computer with `PyTorch`. Despite its impressive zero-shot performance, `SAM` is primarily an object-detection model and cannot be used for the semantic phase segmentation desired in materials science. However, its understanding of shape, texture and other lower-level features apply well to instances of certain phases.

This provides the motivation for and structure of `SAMBA`: a Python web server supplies the image embedding (and later performs random forest segmentation) for the user's image, then browser-based labelling is performed by client-side object segmentations as label suggestions. Labels are confirmed with a left click, and a right click switches length scales of the smart labelling - this was achieved by exposing `SAM`'s hidden multi-mask output. Normally, `SAM` produces three masks per prompt - each representing a different length scale and with an associated internal quality estimate. Giving the user the choice of length scales allows greater flexibility, which is useful given `SAM` is not designed for micrographs, and therefore its internal estimate of quality may not always match the user's.

Standard drawing tools like a variable width brush, polygon, and eraser allow the user to create labels when `SAM` struggles, for example with very high frequency spatial features. Previous labels are not overwritten (unless erased), so new labels can be added on top for boundary fine-tuning or rapid whole-image labelling. These labels can then be downloaded for use in offline workflows, like training task-specific neural networks.

After labelling is complete, the Python web server trains a random forest ensemble to map the feature stack (computed in the background) to the labels. The feature stack creation involves a Python reimplementation of most of the image processing operations outlined in Trainable Weka Segmentation. This is achieved predominantly using the `scikit-image` library (Walt et al., 2014). Once completed, the resulting segmentation of the full image is returned to the user as a variable opacity overlay, allowing for additional labels to refine the output. An optional overlay highlights pixels the classifier is least certain about, which could be useful starting points for new user labels. Improvements in this 'Human-In-The-Loop' (HITL) feedback (*i.e,* feature-weighted uncertainty highlights) are interesting avenues for future work. The trained classifier is a `scikit-learn` object (Pedregosa et al., 2011) - it can be downloaded and applied to future data either on `SAMBA` or in other Python workflows.

`SAMBA` has a clean, responsive user interface implemented in React and Typescript. It works for a range of common image file types (`.png`, `.jpeg`, `.tiff`), as well as large `.tiff` images and `.tiff` stacks. Local hosting is simple and advisable if the user is handling sensitive data or wishes to benefit from GPU acceleration. A desktop version that supports larger files, 3D segmentation and has deeper `PyTorch` integration is planned. `SAMBA` has an associated gallery, which users can optionally publish to once a segmentation is complete. It is hoped this will provide the foundation for a varied, open source library of high-quality image data and encourage further development of generalist machine learning models in materials science. A user manual, instructions for contributing and setup instructions for local hosting is available in the public repository.

## Acknowledgements

This work was supported by funding from the EPSRC Centre for Doctoral Training in Advanced Characterisation of Materials (EP/S023259/1 recieved by RD) and the Royal Society (IF\R2\222059 received by AV as a Royal Society Industry Fellow).

The authors would like to thank other members of the TLDR group for their testing and feedback.

## References

Ahlers, J., Althviz Moré, D., Amsalem, O., Anderson, A., Bokota, G., Boone, P., Bragantini, J., Buckley, G., Burt, A., Bussonnier, M., Can Solak, A., Caporal, C., Doncila Pop, D., Evans, K., Freeman, J., Gaifas, L., Gohlke, C., Gunalan, K., Har-Gil, H., … Yamauchi, K. (2023). *napari: a multi-dimensional image viewer for Python* (Version v0.4.18). Zenodo. https://doi.org/10.5281/zenodo.8115575

Archit, A., Nair, S., Khalid, N., Hilt, P., Rajashekar, V., Freitag, M., Gupta, S., Dengel, A., Ahmed, S., & Pape, C. (2023). Segment anything for microscopy. *bioRxiv*. https://doi.org/10.1101/2023.08.21.554208

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. https://doi.org/10.1093/bioinformatics/btx180

Bai, J., Lu, F., Zhang, K., & others. (2019). *ONNX: Open Neural Network Exchange*. GitHub. https://github.com/onnx/onnx

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., Eren, K., Cervantes, J. I., Xu, B., Beuttenmueller, F., Wolny, A., Zhang, C., Koethe, U., Hamprecht, F. A., & Kreshuk, A. (2019). Ilastik: Interactive machine learning for (bio)image analysis. *Nature Methods*. https://doi.org/10.1038/s41592-019-0582-9

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Riquelme, C., Minderer, M., Puigcerver, J., Evci, U., … Houlsby, N. (2023). Scaling vision transformers to 22 billion parameters. In *ArXiv e-prints*. https://doi.org/10.48550/arXiv.2302.05442

Haase, R., Lee, D., Pop, D. D., & Žigutytė, L. (2023). *haesleinhuepf/napari-accelerated-pixel-and-object-classification: 0.14.1* (Version 0.14.1). Zenodo. https://doi.org/10.5281/zenodo.10071078

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything. In *ArXiv e-prints*. https://doi.org/10.48550/arXiv.2304.02643

Luethi, J., & Hess, M. (n.d.). *napari-feature-classifier: An interactive classifier plugin to use with label images and feature measurements*. https://github.com/fractal-napari-plugins-collection/napari-feature-classifier

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., & Cardona, A. (2012). Fiji: An open-source platform for biological-image analysis. *Nature Methods*, *9*(7), 676–682. https://doi.org/10.1038/nmeth.2019

Walt, S. van der, Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & contributors, the scikit-image. (2014). Scikit-image: Image processing in python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453