The Journal of Open Source Software

IAMAP: Unlocking Deep Learning in QGIS for non-coders and limited computing resources

Paul Tresson [1¶], **Pierre Le Coz** [1,2], **Hadrien Tulet** [1], **Anthony Malkassian** [3], **and Maxime Réjou-Méchain** [1,2]

1 AMAP, Univ. Montpellier, IRD, CNRS, CIRAD, INRAE, Montpellier, France **2** Forest Restoration Research Unit, Department of Biology, Faculty of Science, Chiang Mai University, Chiang Mai, Thailand **3** Université de la Réunion, UMR PVBMT, St. Pierre, La Réunion, France ¶ Corresponding author

DOI: 10.21105/joss.10329

Software

- Review
- Repository
- Archive

Editor: Neea Rusch

Reviewers:

- @florianfranz
- @AMSANJEEV28

Submitted: 01 December 2025
Published: 25 June 2026

Summary

We introduce IAMAP, a user-friendly plugin for the geographic information system QGIS that allows users to (1) tile raster images and feed them into pre-trained deep learning model to extract image features; (2) reduce feature dimensionality; (3) perform clustering on features or their reduced representations; (4) generate feature similarity maps; (5) calibrate and validate supervised machine learning models to create maps. By enabling non-AI specialists to leverage the high-quality features provided by foundation models without requiring GPU capacity or extensive reference datasets, IAMAP contributes to the democratization of these computationally efficient and energy-conscious methods. Development of the plugin is open-source on GitHub. Documentation and tutorials are available on Read the Docs.

Statement of need

The integration of remote sensing data with deep learning approaches is currently revolutionizing Earth observation sciences, leading to significant qualitative and quantitative improvements in large-scale predictions (Yasir et al., 2023; Yuan et al., 2020; Zhu et al., 2017). However, this revolution comes with a number of challenges.

First, over the past decade, most deep learning applications have been highly data-demanding, requiring extensive manual labeling with typically more than one hundred thousand labeled points (Safonova et al., 2023). In most ecological and environmental science studies, constructing such a large reference dataset, through *e.g.*, ground observations or photo-interpretation, remains a major barrier to the implementation of deep learning approaches.

Second, most deep learning developments have been conducted by and for actors able to leverage significant computing power with the training of a model being GPU-dependent. This creates a significant barrier to users who could otherwise benefit from inference-only deep learning applications.

Finally, the use of state of the art models is still mostly confined to users able to code in Python, even considering different wrapper libraries that can ease the implementation of models.

State of the field

Recently, deep learning state of the art has seen an evolution towards the use of large foundation models, trained on large scale datasets in an unsupervised way, and capable of very good few-shot performance (*i.e.*, without expensive supervised training of the network) (Ericsson et al., 2021). The main difference between a pre-trained self-supervised learning (SSL) model and a pre-trained supervised model lies in their training objectives: SSL models are not constrained by predefined labels and are therefore free to explore and encode the intrinsic structure and diversity of the data, often resulting in more general and transferable representations. In contrast, supervised models are explicitly optimized to perform a specific user-defined task, which can lead to highly specialized representations that may overlook other meaningful features in the data. As such, SSL foundation models can perform well even in low-shot or zero-shot tasks, *i.e.*, using the model as is, with few or no training data. Consequently, SSL models are considered particularly promising for remote sensing tasks, as demonstrated by recent works and initiatives (Cong et al., 2022; Jakubik et al., 2023; Marsocci et al., 2024; Xiong et al., 2024).

With the democratization of deep learning, some developers have already worked on the integration of deep learning models in geographic information systems such as the open source and widely used QGIS software (QGIS Development Team, 2025). However, at the time of writing, these solutions mostly focus on fine-tuning models or using a model in inference only (*e.g.*, see Aszkowski et al., 2023; Zhao et al., 2023). Then, they are only usable by users with access to high-end computing power, extensive dataset, or interested in a task for which a specific model was already trained.

We have therefore developed a plugin that allows users to use a variety of deep learning models and manipulate their features in a user-friendly fashion.

Software design

This plugin aims at allowing two main tasks:

1. feeding a tiled raster through a pre-trained deep learning model, typically via the *Timm* and *Hugging Face* libraries (Wightman, 2019; Wolf et al., 2020).
2. manipulating the produced features with common machine learning tools, provided by the *Scikit-Learn* library.

These tasks are separated into five modules depending on the type of models and operations the user wants to perform: deep learning feature extraction (see Figure 2), dimension reduction, clustering, similarity computation, or fitting a machine learning model in a supervised way (see Figure 1).

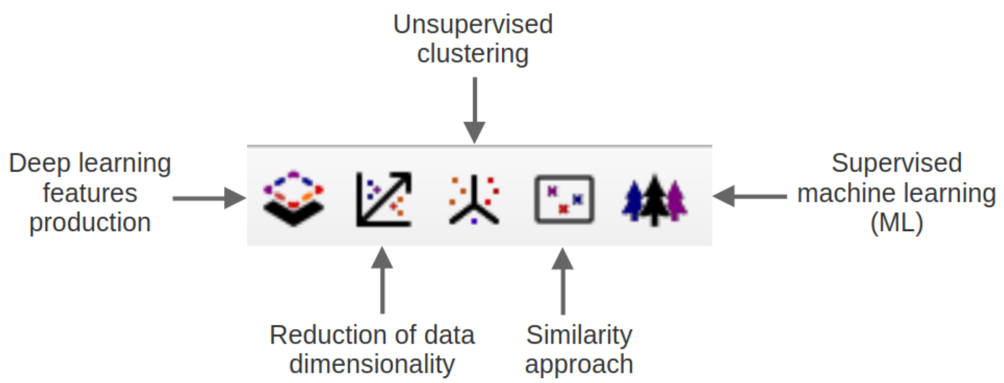

Figure 1: The five main modules of the IAMAP plugin.

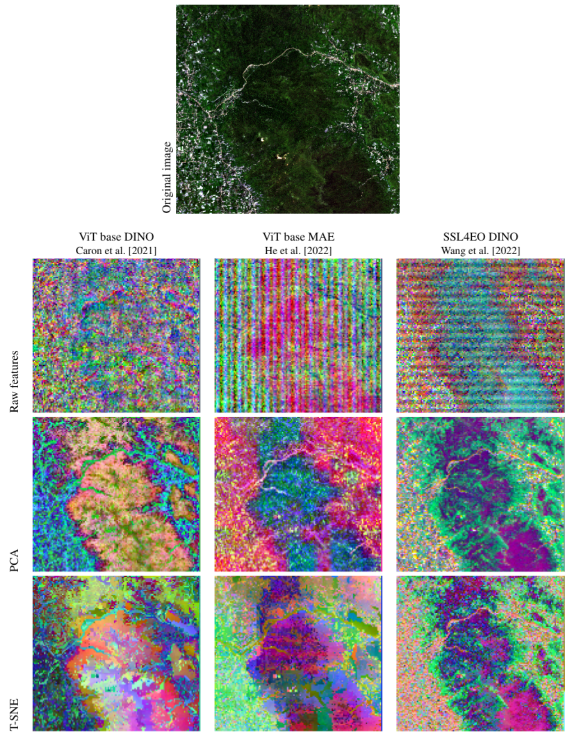

Figure 2: A Sentinel-2 image of a forested landscape in Thailand (Khao Banthat Wildlife Sanctuary; Lat 7.53°, Lon 99.82°) processed by different backbones. The top row represents the first three feature dimensions output by the models (which may not be the most informative). The second row shows a 3D PCA of the features mapped to the red, green, and blue channel respectively. The third row shows a projection using a 3D t-SNE.

- **The deep learning feature extraction module** allows users to feed a raster into a ViT-like backbone (Vision Transformer; see (Dosovitskiy et al., 2020)) . The user can choose a raster loaded in QGIS, a pre-trained model that is downloaded from Timm/Hugging Face or pass local pre-trained weights. The raster is then tiled according to user defined rules and fed into the model using a light fork of the *TorchGeo library* (Stewart et al., 2025). Resulting features are saved into a GeoTIFF file and can then be used for further analysis.
- **The dimension reduction module** uses *Scikit-Learn* API to feed a raster into a model and collect the resulting raster. All algorithms available in the *Scikit-Learn* `decomposition` and `cluster` modules that have common APIs (namely, a `fit()`, a `transform()`, or a `fit_transform()` method) are available. The parameters can be fed following a JSON format.
- **The clustering module** works in a similar manner with algorithms available in the *Scikit-Learn* `cluster` module sharing common APIs (namely, a `fit()`, a `predict()`, or a `fit_predict()` method).
- **The similarity module** computes the similarity between a raster and user-defined template pixels (via a vector layer).
- **The machine learning module** allows the user to fit a supervised machine learning model available in the *Scikit-Learn* `ensemble` and `neighbors` modules. Hyperparameters (*e.g.* validation scheme, labelled data…) can be defined by the user.

A core constraint of the development is for the plugin to be accessible for a non-coding user without a GPU and with limited internet access. As such, all design and development is done on CPU on a laptop to assess usability with no GPU. Various optimizations are available, such as quantization, scheduled pauses, and saving progress to disk. We used code from the Deepness plugin (Aszkowski et al., 2023) to provide automated dependency-installation dialog.

Research impact statement

The plugin is used internally and has been presented in several webinars with strong positive feedback from potential users. It has also been used for research by Malkassian et al. (a preview of the results is available in the documentation).

AI usage disclosure

AI was only used to help generate small functions and "boilerplate" code on common tasks that could immediately be tested.

Availability

Development of the plugin is open-source on GitHub. Documentation is available on Read the Docs. The plugin is developed in continuous integration. We plan to publish the plugin in the official QGIS Plugin repository to further ease the installation process.

Acknowledgements

The authors would like to thank all people who have tested this software during development and have provided meaningful feedback.

Conflict of interest

The authors declare no conflict of interest.

References

Aszkowski, P., Ptak, B., Kraft, M., Pieczyński, D., & Drapikowski, P. (2023). Deepness: Deep neural remote sensing plugin for QGIS. *SoftwareX*, *23*, 101495. https://doi.org/10.1016/j.softx.2023.101495

Cong, Y., Khanna, S., Meng, C., Liu, P., Rozi, E., He, Y., Burke, M., Lobell, D., & Ermon, S. (2022). SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. *Advances in Neural Information Processing Systems*, *35*, 197–211. https://doi.org/10.52202/068431-0015

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., & others. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv Preprint arXiv:2010.11929*. https://arxiv.org/pdf/2010.11929/100

Ericsson, L., Gouk, H., & Hospedales, T. M. (2021). How well do self-supervised models transfer? *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5414–5423. https://doi.org/10.1109/cvpr46437.2021.00537

Jakubik, J., Roy, S., Phillips, C. E., Fraccaro, P., Godwin, D., Zadrozny, B., Szwarcman, D., Gomes, C., Nyirjesy, G., Edwards, B., Kimura, D., Simumba, N., Chu, L., Mukkavilli, S. K., Lambhate, D., Das, K., Bangalore, R., Oliveira, D., Muszynski, M., … Ramachandran, R. (2023). Foundation Models for Generalist Geospatial Artificial Intelligence. *Preprint Available on arXiv:2310.18660*. https://arxiv.org/pdf/2310.18660

Marsocci, V., Jia, Y., Bellier, G. L., Kerekes, D., Zeng, L., Hafner, S., Gerard, S., Brune, E., Yadav, R., Shibli, A., Fang, H., Ban, Y., Vergauwen, M., Audebert, N., & Nascetti, A.

(2024). *PANGAEA: A global and inclusive benchmark for geospatial foundation models*. https://arxiv.org/abs/2412.04204

QGIS Development Team. (2025). *QGIS geographic information system*. Open Source Geospatial Foundation. https://qgis.org

Safonova, A., Ghazaryan, G., Stiller, S., Main-Knorn, M., Nendel, C., & Ryo, M. (2023). Ten deep learning techniques to address small data problems with remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, *125*, 103569. https://doi.org/10.1016/j.jag.2023.103569

Stewart, A. J., Robinson, C., Corley, I. A., Ortiz, A., Lavista Ferres, J. M., & Banerjee, A. (2025). TorchGeo: Deep learning with geospatial data. *ACM Transactions on Spatial Algorithms and Systems*, *11*(4), 1–28. https://doi.org/10.1145/3707459

Wightman, R. (2019). PyTorch image models. In *GitHub repository*. https://github.com/rwightman/pytorch-image-models; GitHub. https://doi.org/10.5281/zenodo.4414861

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P. von, Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … Rush, A. M. (2020). *HuggingFace's Transformers: State-of-the-art natural language processing*. https://arxiv.org/abs/1910.03771

Xiong, Z., Wang, Y., Zhang, F., Stewart, A. J., Hanna, J., Borth, D., Papoutsis, I., Saux, B. L., Camps-Valls, G., & Zhu, X. X. (2024). Neural plasticity-inspired multimodal foundation model earth observation. *arXiv Preprint arXiv:2403.15356*. https://arxiv.org/pdf/2403.15356

Yasir, M., Jianhua, W., Shanwei, L., Sheng, H., Mingming, X., & Hossain, M. (2023). Coupling of deep learning and remote sensing: A comprehensive systematic literature review. *International Journal of Remote Sensing*, *44*(1), 157–193. https://doi.org/10.1080/01431161.2022.2161856

Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., Xu, H., Tan, W., Yang, Q., Wang, J., Gao, J., & Zhang, L. (2020). Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, *241*, 111716. https://doi.org/10.1016/j.rse.2020.111716

Zhao, Z., Fan, C., & Liu, L. (2023). *Geo SAM: A QGIS plugin using Segment Anything Model (SAM) to accelerate geospatial image segmentation* (Version 1.1.0). Zenodo. https://doi.org/10.5281/zenodo.8191039

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*(4), 8–36. https://doi.org/10.1109/mgrs.2017.2762307