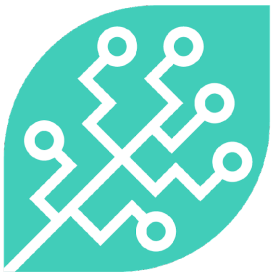

MAMBO

MODERN APPROACHES TO THE MONITORING OF BIODIVERSITY

D4.7 [Open-source deep learning framework for habitat extent mapping]

28/02/2024

Lead beneficiary: INRIA

Author/s: Théo Larcher, Alexis Joly

Funded by
the European Union

Prepared under contract from the European Commission
Grant agreement No. 101060639
EU Horizon Europe Research and Innovation Action

Project acronym:	**MAMBO**
Project full title:	**Modern Approaches to the Monitoring of Biodiversity**
Project duration:	01.09.2022 – 31.08.2026 (48 months)
Project coordinator:	Dr. Toke Thomas Høye, Aarhus University (AU)
Call:	HORIZON-CL6-2021-BIODIV-01
Deliverable title:	
Deliverable n°:	D4.7
WP responsible:	France Gerard (CEH)
Nature of the deliverable:	Other
Dissemination level:	Public
Lead beneficiary:	INRIA
Due date of deliverable:	M18
Actual submission date:	M18

Deliverable status:

Version	Status	Date	Author(s)
0.1	Toc	17/01/2024	Alexis Joly (Inria)
0.2	First Draft	6/02/2024	Alexis Joly (Inria)
1.0	First complete version	21/02/2024	Alexis Joly (Inria), Théo Larcher (Inria)
1.1	Revised version	28/02/2024	Alexis Joly (Inria), Théo Larcher (Inria), France Gerard (CEH)

Table of Contents

Table of Contents **4**
Summary **5**
1 Introduction **5**
2 MALPOLON Framework v1.0 **6**
2.1 Overview 6
2.2 Source code repository 8
2.3 Documentation 9
3 Tests and community **9**
References **10**

Summary

This document provides a description of the first version of the open-source framework developed by Inria as part of task T4.1 of the MAMBO project on habitat extent monitoring. The overall approach employed in MAMBO to monitor habitat extent comprises two steps: (i) mapping plant species composition of habitats, (ii) inferring habitat types (EUNIS) from the species compositions. This first version of the framework is mainly focused on the first step, i.e., the mapping of the species composition. This step is based on what we call deep-learning based species distribution models (DeepSDMs). DeepSDMs allow modelling the distribution of thousands of species in a single joint model, taking as input very-high resolution and complex data such as remote-sensing images or time-series. The open-source framework developed, called MALPOLON, aims to facilitate DeepSDM training and sharing for users with general Python language skills, but no advanced skills (e.g. modeling ecologists). It is freely available on github, along with extensive documentation and examples of use in various scenarios.

1 Introduction

As illustrated in Figure 1, the overall approach employed in MAMBO to monitor habitat extent comprises two steps: (STEP1) mapping plant species composition of habitats, (STEP2) inferring habitat types (EUNIS) from the species compositions. The MALPOLON framework, described in this

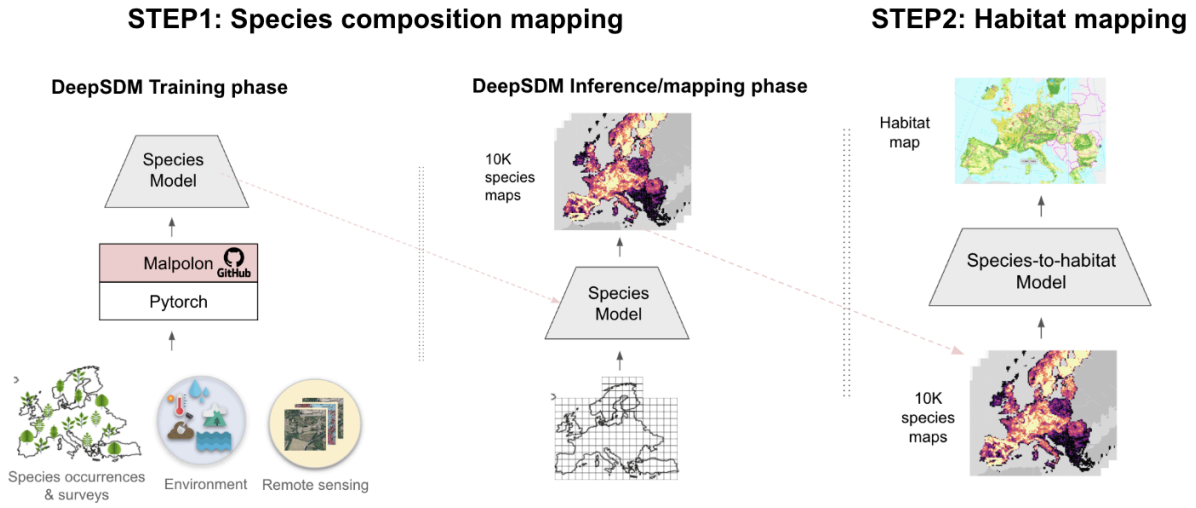

Figure 1 - **Global pipeline of MAMBO habitat extent monitoring**

deliverable, focuses on the first step. As illustrated on Figure 2, It allows training various types of SDMs based on deep learning using a variety of input variables such as environmental rasters (bioclim, land cover, human footprint, etc.), remote sensing images (e.g. Sentinel2 images) or time series (e.g. landsat data time series). The framework is built

on top of the PyTorch deep learning framework and enables compatibility with TorchGeo, a PyTorch domain library providing datasets, samplers, transforms, and pre-trained models specific to geospatial data. MALPOLON allows using various types of neural network architectures ranging from simple Multi-Layer Perceptron (MLP) to complex models such as vision transformers (in particular thus included in popular TIMM models catalogue).

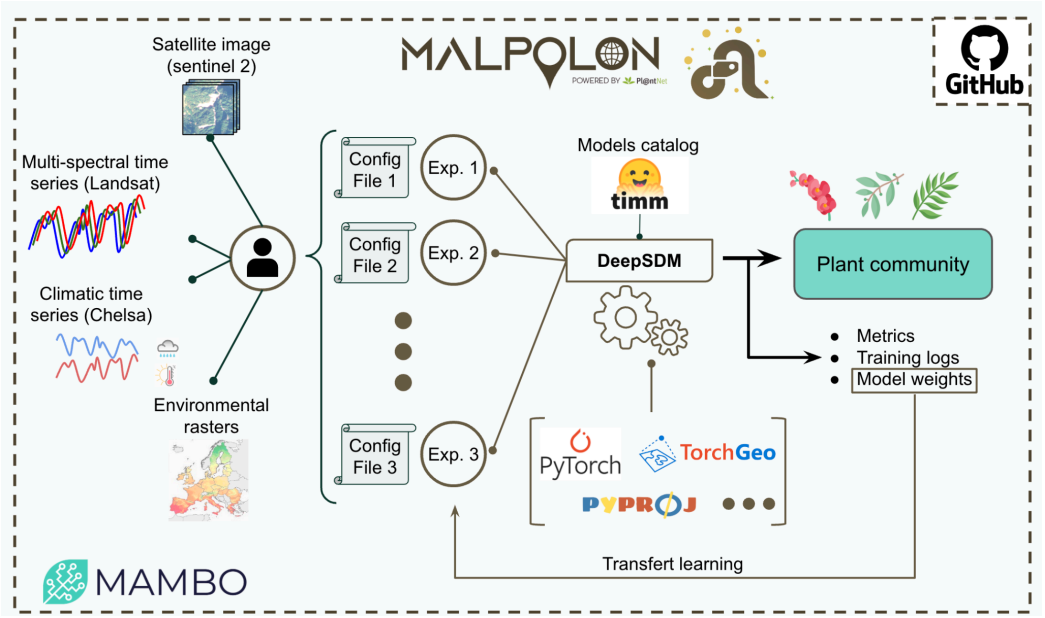

Figure 2 - **Graphical abstract of MALPOLON open source software**

One interest of Deep SDM models is that they can predict suitability scores for a very large number of species using a single model rather than a multitude of individual single-species models. They are capable of capturing complex interactions among the input environmental covariates even if they are composed of heterogeneous and complex modalities (as discussed for instance in [1,2,3]).

2 MALPOLON Framework v1.0

2.1 Overview

MALPOLON is a Python based framework designed to be both accessible to users with minimal knowledge of Python or PyTorch; and highly customizable by advanced users interested in re-defining classes[1] and methods to perfectly fit their datasets and use case.

The framework is currently operable by command line only and consists of 2 main sections:

1. **Experiments**: a directory containing different use case scenarios with pre-written plug-and-play example files, "*main scripts*", which the users can use to run training and inference loops either on the provided data samples or their own datasets.
2. **Engine**: a directory containing all the classes necessary to interact with the various datasets and models used by the main scripts.

[1] A Python class is a blueprint for creating objects, providing a structure that defines the attributes and behaviors common to those objects.

In order to make the framework accessible for non-expert users, all experiments are parametrized by a corresponding configuration file which follows a simple and intuitive categories hierarchy structure and requires little to no programming language knowledge. These parameters allow the user to indicate their models where to find their input data, customize their model's hyperparameters, choose between training and inference mode and much more. An exhaustive list of the available parameters is given in the repository documentation for each experiment.

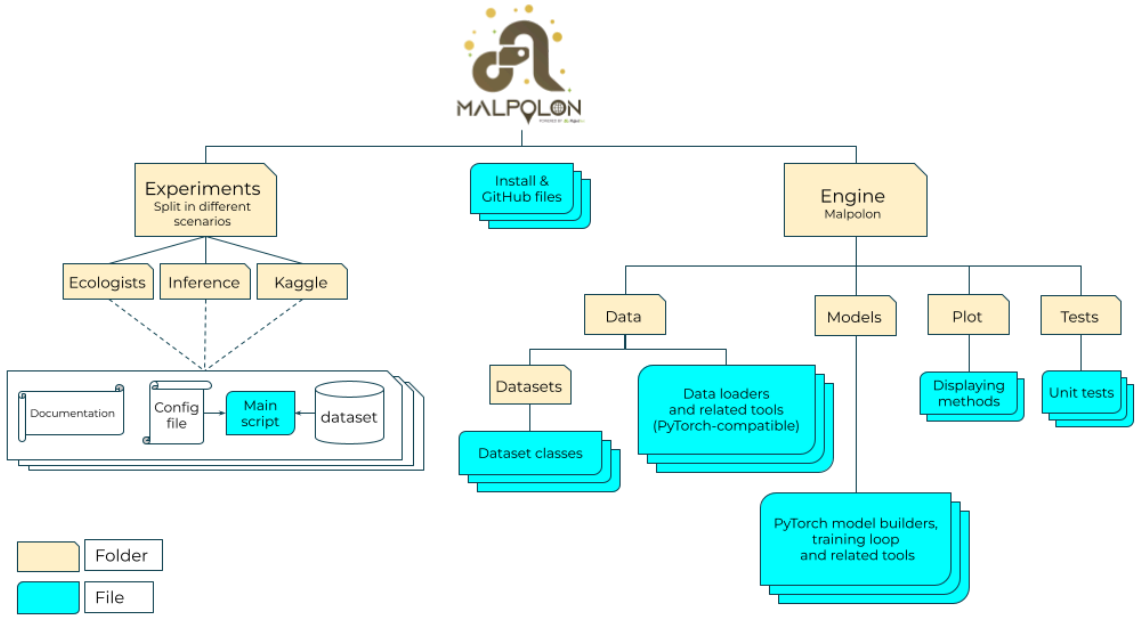

Figure 3 - **Simple meta-class diagram of MALPOLON**

To address the needs of more advanced users, the framework is highly modular, with well documented (cf. section 2.3) and identifiable classes. As such, very generic "base classes" are provided for each aspect of a deep learning pipeline. Based on top of these classes and following the encapsulation principle, MALPOLON provides dataset-specific classes which enables us to provide users with functional experiment examples.

The framework currently covers and provides experiment examples for 3 scenarios:

a. "Ecologists"

In this scenario, users possess a dataset of their own on which they want to train a model while being able to easily customize the training process and the model architecture.

b. "Inference"

In this scenario, users possess both a test dataset and a trained model. They want to simply test the performances of their model with relevant metrics.

c. "Kaggle"

In this scenario, users are participants of a GeoLifeCLEF challenge and look for plug-and-play examples of data loaders and/or training examples, specifically developed for the challenge dataset.

- Typical sequence of the “Ecologists” scenario

The typical and recommended way users should use the framework to train a model is to use (or duplicate) an experiment’s main script, update (or create) its associated configuration file, and choose a data loader that is best adapted to their data. If none are satisfactory, they should re-define the missing functions of the generic class (or any of the dataset-specific classes) and call them in MALPOLON’s data module.
Then users should choose a model by inputting a correct name in their configuration file (list of available models can be found on TIMM’s documentation).
Finally, they should parametrize their model through the configuration file, with custom or default values.
Metrics, logs and model’s weights are saved in a unique output folder.

- Typical sequence of the “Inference” scenario

The typical and recommended way users should use the framework to perform inference with a trained model on a test dataset, is to use (or duplicate) an experiment’s main script and update the dataset path as well as the model’s weight path. Similarly to the “Ecologists” scenario, users may import their own modifications to data loaders and other classes before running the inference pipeline. Metrics, logs and predictions are then outputted in a unique output folder.

- Typical sequence of the “Kaggle” scenario

This scenario is suited for participants of Kaggle competitions organized by INRIA, namely GeoLifeCLEF. In this scenario, data loading scripts are provided which users can use to plug in their deep learning pipeline (PyTorch compatible).

2.2 Source code repository

The source code is open-source and hosted on a GitHub repository :
https://github.com/plantnet/malpolon/tree/main

The framework ensures that both the code and the experiment datasets are approaching FAIR standards by the following means:

- **Findable**: the source code is hosted on a public GitHub repository, easily findable on a search engine, taking advantage of the visibility of the Pl@ntNet organization, to which the MALPOLON repository is associated.

- **Accessible**: the source code and experiment datasets are open and available either directly on GitHub, on Kaggle or on a self-hosted Seafile platform. In the 2 former cases, instructions and links are provided on the GitHub repository to guide the user as to where to find those data and how to download them.

- **Interoperable:** the source code is written at 99.9% in Python, compatible with version 3.10 and over, and is based upon standard libraries such as PyTorch, Numpy,

TorchGeo or Matplotlib. The code quality is being checked by linters so that it approaches near perfect PEP8 standards; and tested by unit tests so that it is reproducible and stable.
The datasets are stored using well known formats widely used in the remote-sensing, ecology and IA domains such as JPEG2000 for satellite patches, TIF for environmental and geospatial satellite rasters, and CSV for plant species observations.

- **Reusable:** the source code and datasets come with extensive documentation and datasets descriptions, either on the GitHub itself or on the linked Kaggle and Seafile pages. Plug-and-play examples are provided to give users an easy way of familiarising themselves with the framework. All data and source code are registered under the MIT License, and contribution and usage guidelines have been provided so that any user can contribute in improving the framework via pre-formatted Pull Requests.

2.3 Documentation

The framework repository front page contains a description of the project and its partners with instructions on how to install and use the framework for different use cases.

A code documentation already being hosted on GitHub Pages at this link. It is set to be updated every time new content is pushed to the main branch.

3 Tests and community

3.1 Users

Users and testers of the framework currently consist of PhD students, postdocs, permanent researchers and engineers.

- **Maxime Ryckewaert** is postdoc researcher at INRIA. He uses the framework in the context of the B-cubed EU project to train plant species distribution models with Pl@ntNet's observations and feed them to a pipeline which generates dynamic maps to plot the predictions. Maxime is also involved in the active development of MALPOLON (in addition to main developper, Théo Larcher).
- **Rémi Pallard** is a senior research engineer at CIRAD who is currently testing the framework before integrating it within the GUARDEN EU project's workflow.
- **Benjamin BOURREL** is a permanent researcher at INRIA who uses the framework to train CNNs on satellite images to predict the abundance of fish species in specific geospatial locations, within the French project Fish-Predict.
- **Gaëtand MORAND** is a research engineer at IRD who uses deep learning models to predict offshore species distributions. He tested Malpolon on multi-label / multi-class classification for abundance estimation of fish species in remote tropical areas, and manually tweaked the framework at the model's *metrics* and *loss* levels to better fit his needs which in turn resulted in official changes within the framework.
- **Camille GARCIN** is a postdoc researcher at INRIA who uses the framework to experiment a new multi-label loss function on the GeoLifeCLEF dataset using MALPOLON.

Future potential users who expressed their interest include:

- The MARBEC research lab: many interactions with David Mouillot and his collaborators/students (Marieke Schultz, Simon Bettinger) about upcoming use of MALPOLON to predict marine species distributions.
- The University of Helsinki: several interactions with Gleb TIKHONOV (postdoc researcher) who works on Joint-SDM and probabilistic machine learning for analysis of community data in ecology.
- The LECA research lab (univ. Grenoble Rhône-Alpes): several interactions with Wilfried Thuiller and his collaborators/students (Sara Si-Moussi, Gabrielle Deschamps) about potential use of MALPOLON.
- The AMAP research lab (CIRAD): several ecologists/modelizers regularly work with SDMs in the lab and expressed their interest in experimenting MALPOLON (e.g. Ghislain Vieilledent, Raphaël Pélissier, Maxime Rejou, Paul Tresson)
- The association Gentiana (Isère, France): several interactions with Nicolas Faure, Alain Poirel and François Munoz who fit SDMs on top of the Infloris dataset (currently containing 700,000 validated, quality occurrences for the whole Isère department).

3.2 Community & dissemination

The MALPOLON framework is already public on GitHub but communication efforts must be intensified to build an active community. For such endeavour, a presentation of the framework is set to take place at CIRAD/AMAP to look for potential users, feedback but also needs that were not initially listed.

MALPOLON will be distributed as one of the resources to be used by participants in the international GeoLifeCLEF 2024 challenge. It will also provide compatibility with the former GeoLifeCLEF 2023 challenge.

In April, INRIA will participate in the European project B-cubed hackathon in Brussels, Belgium with the goal of producing Deep-SDM using MALPOLON, which will serve as dissemination and potentially attract new users.

Citations of MALPOLON currently include a publication under revision from Gaëtand MORAND (see preprint at [4]) and a publication in progress from Benjamin Bourel.

References

[1] Deneu, B., Servajean, M., Bonnet, P., Botella, C., Munoz, F., & Joly, A. (2021). Convolutional neural networks improve species distribution modelling by capturing the spatial structure of the environment.

[2] Estopinan, J., Servajean, M., Bonnet, P., Munoz, F., & Joly, A. (2022). Deep species distribution modeling from sentinel-2 image time-series: a global scale analysis on the orchid family. *Frontiers in Plant Science*, *13*, 839327.

[3] Botella, C., Deneu, B., Gonzalez, D. M., Servajean, M., Larcher, T., Leblanc, C., ... & Joly, A. (2023, December). Overview of GeoLifeCLEF 2023: Species composition prediction with high spatial resolution at continental scale using remote sensing. In *CLEF 2023: Conference and Labs of the Evaluation Forum*.
[4] Morand, G., Joly, A., Rouyer, T., Lorieul, T., & Barde, J. (2023). Predicting species distributions in the open oceans with convolutional neural networks. *bioRxiv*, 2023-08. https://doi.org/10.1101/2023.08.11.551418