Evoland
LAND MONITORING EVOLUTION

T2.21 Source Code

Work Package 2 - Method development

Authors: Wanda De Keersmaecker, Kasper Bonte (VITO), Conrad Albrecht, Nassim Ait Ali Braham (DLR), Jordi Inglada, Ekaterina Kalinicheva (CESBIO), Julien Michel (UT3), Janik Deutscher, Martin Puhm, Heinz Gallaun (JR)

Date: 31.03.2025

Funded by the European Union

Full Title	**Evolution of the Copernicus Land Service portfolio** (integrating novel EO data and latest machine learning algorithms to continuously monitor the status, dynamics and biomass of the land surface)			
Project number	101082130		**Acronym**	EVOLAND
Start date	01.01.2023		**Duration**	36 months
Granting authority	European Health and Digital Executive Agency (HaDEA)			
Project Coordinator	Vlaamse Instelling voor Technologisch Onderzoek N.V. (VITO)			
Date of delivery	**Contractual**	M28	**Actual**	M28
Type	OTHER		**Dissemination level**	PU - Public
Lead beneficiary	VITO			
Lead author	Ruben Van De Kerchove (VITO)		**Email**	ruben.vandekerchove@vito.be
Other authors	Wanda De Keersmaecker, Kasper Bonte (VITO), Conrad Albrecht, Nassim Ait Ali Braham (DLR), Jordi Inglada, Ekaterina Kalinicheva (CESBIO), Julien Michel (UT3), Janik Deutscher, Martin Puhm, Heinz Gallaun (JR)			
Reviewer(s)	Tim Ng (VITO)			
Keywords	Method development, work package 2, source code			

Document Revision History				
Version	**Issue date**	**Stage**	**Changes**	**Contributor**
0.1	10.02.2025	Draft	Review of D2.20	ALL
1.0	31.03.2025	Final	Review and finalization	VITO

1 Weakly supervised learning

1.1 DLR code

Example of code

Module Imports

```
from lightning.pytorch.trainer import Trainer
from evoland.data import DataDownloader, patchify, TreeCoverDataModule
from evoland.models import SemanticSegmentationModule
import wget, os
```

Model Inference Demo

```
import lightning.pytorch
```

```
model = SemanticSegmentationModule(
    num_classes=2,
    model="fcn",
    backbone="custom_resnet18",
    weights=None,
    in_channels=12,
    ignore_index=255,
    loss='ce',
)
```

```
Setting up SimpleSegmentationModel
```

```
type(model), isinstance(model, lightning.pytorch.LightningModule)
```

```
(evoland.models.forest_segmentation.SemanticSegmentationModule, True)
```

```
datamodule = TreeCoverDataModule(
    root         = ".",
    img_dir_name = IMAGE_DIR,
    batch_size   = 1,
    num_workers  = 0,
    patch_size   = PATCH_SIZE,
    use_masks    = False,
)
```

```
type(datamodule), isinstance(datamodule, lightning.pytorch.LightningDataModule)
```

```
(evoland.data.treecover.TreeCoverDataModule, True)
```

```
MODEL_PATH = 'custom_resnet18_l2a.ckpt'
trainer = Trainer(
    accelerator='cpu',
    devices=1,
)
output = trainer.predict(
    model=model,
    datamodule=datamodule,
    ckpt_path=MODEL_PATH,
)
```

	Description
Name	EvoLand - software
Description	Python package with modules for AI methods evoland.models and evoland.data for loading corresponding data inputs including helper classes and functions in evoland.utils. An overview for installation (mamba/conda), documentation (Pdoc), and usage (Jupyter notebook) provides https://github.com/Evoland-Land-Monitoring-Evolution/evoland-software/blob/main/README.md
Version	0.1 (interim version)
Access	EvoLand GitHub: https://github.com/Evoland-Land-Monitoring-Evolution/evoland-software
License	Apache-2.0

Citation	Braham, N., Jancauscas, V., Wang, Y., Albrecht, C. M., "evoland – a Python package for self-supervised AI methodologies for remote sensing employed and developed for the EU Horizon project EvoLand" (2024)

1.2 CESBIO code

1.2.1 MMDC pre-training

	Description
Name	MMDC-SingleDate
Description	Models and training code for the Multi-Modal Data Cube generic embeddings using Sentinel-1 and Sentinel-2 in single-date configuration.
Version	0.1.0
Access	GitHub https://github.com/Evoland-Land-Monitoring-Evolution/MMDC-SingleDate
License	Apache-2.0 license
DOI	https://doi.org/10.5281/zenodo.10876755

1.2.2 MALICE and MALICE Aux pre-training

	Description
Name	malice
Description	Models and training code for Multimodal ALIse Cross-Encoder (MALICE) and MALICE with auxiliairy data embeddings using Sentinel-1 and Sentinel-2 in multi-date configuration.
Version	0.1.0
Access	GitHub https://github.com/Evoland-Land-Monitoring-Evolution/malice
License	Apache-2.0 license

1.2.3 ProsailVAE embeddings extraction with openEO

	Description
Name	openeo_pvae
Description	Sentinel-2 single-date embeddings extraction with prosailVAE model using openEO client
Version	0.1.0
Access	GitHub https://github.com/Evoland-Land-Monitoring-Evolution/openeo_pvae
License	Apache-2.0 license

1.2.4 MALICE embeddings extraction with openEO

	Description
Name	openeo_malice
Description	Sentinel-2 and Sentinel-1 ASC multi-temporal embeddings extraction with MALICE model using openEO client
Version	0.1.0
Access	GitHub https://github.com/Evoland-Land-Monitoring-Evolution/openeo_malice
License	Apache-2.0 license

1.2.5 ProsailVAE pre-training

	Description
Name	prosailvae
Description	Models and training code for ProsailVAE embeddings using Sentinel-2 in single-date configuration.
Version	0.1.0
Access	GitHub https://github.com/Evoland-Land-Monitoring-Evolution/prosailvae
License	Apache-2.0 license

2 Improved resolution

2.1 UT3 code

Example of code

Code Blame 438 lines (369 loc) · 12.8 KB

```
"""
Inference script
"""

import argparse
import logging
import os
import sys
from dataclasses import dataclass

import numpy as np
import onnxruntime as ort  # type: ignore[import-untyped]
import rasterio as rio  # type: ignore[import-untyped]
from affine import Affine  # type: ignore[import-untyped]
from sensorsio.sentinel2 import Sentinel2
from sensorsio.sentinel2_l1c import Sentinel2L1C
from tqdm import tqdm


@dataclass(frozen=True)
class Chunk:
    source_area: rio.coords.BoundingBox
    target_area: rio.coords.BoundingBox


def generate_chunks(
    roi: rio.coords.BoundingBox,
    tile_size_in_meters: float,
    margin_in_meters: float,
) -> list[Chunk]:
```

	Description
Name	sentinel2_superresolution
Description	Generate 5m super-resolved images from Sentinel-2 L2A (Theia) products (bands B02, B03, B04, B05, B06, B06, B08, B8A, B11 and B12) using Single Image Super-Resolution model trained in the frame of the EVOLAND Horizon Europe.
Version	1.0
Access	GitHub https://github.com/Evoland-Land-Monitoring-Evolution/sentinel2_superresolution
License	Apache-2.0
DOI	https://doi.org/10.3390/data7070096
Reference	Michel, J.; Vinasco-Salinas, J.; Inglada, J.; Hagolle, O. SEN2VENµS, a Dataset for the Training of Sentinel-2 Super-Resolution Algorithms. Data 2022, 7, 96. https://doi.org/10.3390/data7070096

3 Continuous Monitoring

3.1 JR code

Example of code

run_nrt.py

```python
import argparse
from datetime import datetime, date, timedelta
import os
import shutil
import sys

from nrt.monitor.ewma import EWMA
from nrt.monitor.cusum import CuSum
from nrt.monitor.mosum import MoSum

import numpy as np


def recode_timestamp(ga, output_file, writer):
    """
    Recode the detection timestamp used by the nrt module.

    By default the nrt module uses a timestamp format based on the number of
    days since 1970-01-01, which is not human-readable and difficult to interpret.
    This function recodes the values to the format
    1000*two-digit-year + day-of-year. Two examples: 2004-01-05 -> 4005,
    2020-02-01 -> 20032

    Args:
    ga (rasterioutils.GeoArray): Array representation of the original nrt output. The date
        info is expected to be in a layer called 'timestamp'.
    output_file (str): Path of the recoded output to be created.
    writer (rasterioutils.ArrayWriter): Writer instance

    """
    output_arr = np.zeros(ga.array.shape[2:], np.uint16)
    ref_date = date(1970, 1, 1)
    for delta_days in np.unique(ga.timestamp):
        if delta_days > 0:
            alarm_date = ref_date + timedelta(delta_days.item())
            output_arr[ga.array[0, 0, :, :] == delta_days] = int(alarm_date.strftime('%y%j'))
```

	Description
Name	ts-modeling-benchmarking
Description	This repository represents the implementation of a benchmarking suite able to run several open-source 3rd party algorithms for time series modelling and unsupervised change detection.
Version	1.0
Access	GitHub https://github.com/Evoland-Land-Monitoring-Evolution/ts-modeling-benchmarking
License	Apache-2.0

3.2 VITO Code

Example of code

```
else:
    LSC_PATH = _DATASET_PATH_HPC
    HIST_PATH = _HIST_PATH_HPC


def compute_pixel_coordinates(bounds, shape):
    """
    Compute the y and x coordinates for every pixel in an image.

    Args:
    bounds (tuple): A tuple containing (xmin, ymin, xmax, ymax).
    shape (tuple): A tuple containing the image shape (rows, columns).

    Returns:
    tuple: Two arrays containing y and x coordinates for every pixel.
    """
    xmin, ymin, xmax, ymax = bounds
    rows, cols = shape

    x_res = (xmax - xmin) / cols
    y_res = (ymax - ymin) / rows

    if x_res != y_res:
        raise ValueError("Different resolution for y and x axis are not "
                         "supported. Bounds and shape are not consistent "
                         "with the same resolution on both axis.")

    res_half = x_res / 2

    xx = np.linspace(xmin + res_half,
                     xmax - res_half,
                     cols)

    yy = np.linspace(ymax - res_half,
                     ymin + res_half,
                     rows)

    return yy, xx
```

	Description
Name	Evoland-lsc
Description	The first version of the python code for training LSC models (both the U-Net and CatBoost model) and inference is provided
Version	1.0
Access	https://github.com/Evoland-Land-Monitoring-Evolution/evoland-lsc
License	Creative Commons Legal Code CC0 1.0 Universal

4 Biomass mapping

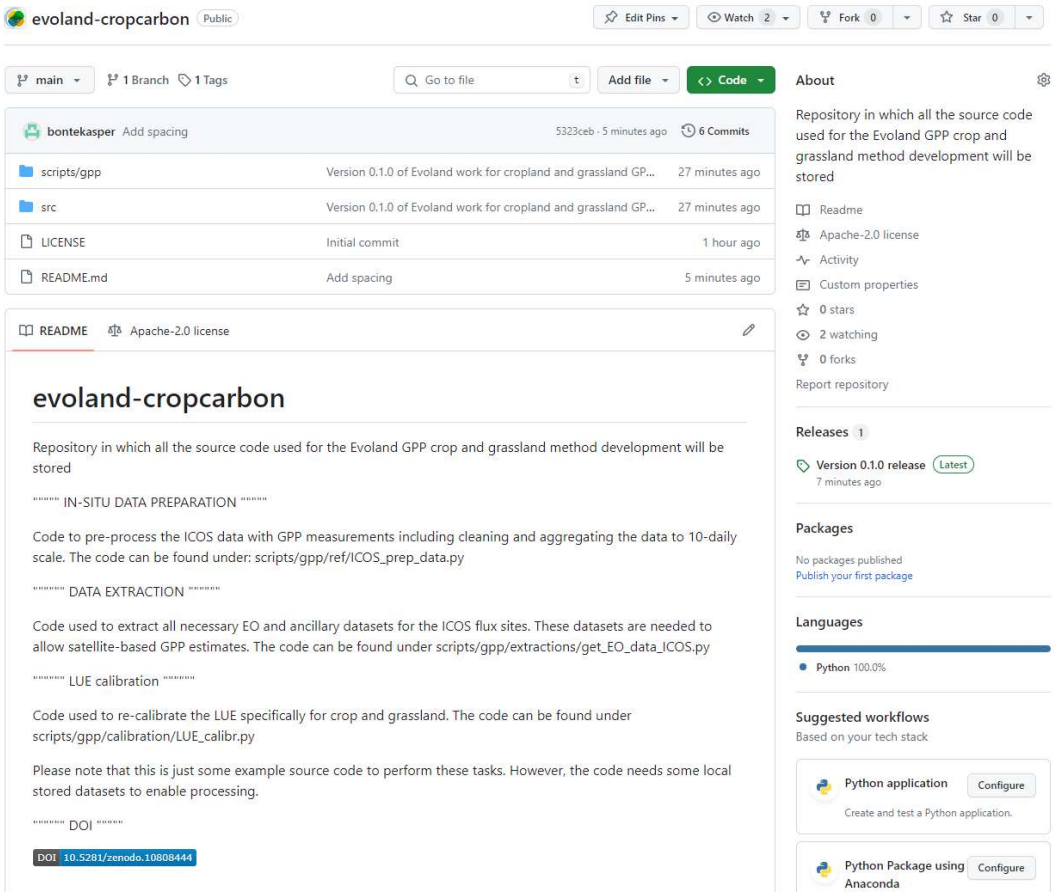
4.1 JR code

	Description
Name	RF1_Sweden_FCH_Predict.py
Description	The script uses a trained Random Forest Model for Forest Canopy Height (FCH) estimation and a multiband geotiff-File which includes various features. The script will output a GeoTIFF file with the predicted FCH. The script requires the following Python packages: - rasterio - numpy - pandas - joblib
Version	1.0
Access	https://github.com/Evoland-Land-Monitoring-Evolution/ForestBiomass
License	Creative commons attribution 4.0 international license

	Description
Name	RF1_Sweden_AGB_Predict.py
Description	The script uses a trained Random Forest Model for Above Ground Woody Biomass (AGB) estimation and a multiband geotiff-File which includes various features.The script will output a GeoTIFF file with the predicted AGB. The script requires the following Python packages: - rasterio - numpy - pandas - joblib
Version	1.0
Access	https://github.com/Evoland-Land-Monitoring-Evolution/ForestBiomass
License	Creative commons attribution 4.0 international license

5.1 VITO code

Example of code



	Description
Name	evoland-cropcarbon
Description	First version of the GPP crop and grassland monitoring workflow for EvoLand. In this repository all code used to prepare the in-situ data for calibration and to retrieve the needed EO and ancillary datasets is provided. In this release, the recalibration of the LUE for crop and grassland is included.
Version	0.1.0
Access	https://github.com/Evoland-Land-Monitoring-Evolution/evoland-cropcarbon
License	Apache-2.0 license
DOI	https://zenodo.org/doi/10.5281/zenodo.10808443

Evoland
LAND MONITORING EVOLUTION

EvoLand Consortium

vito Remote Sensing

International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

CLS
COLLECTE LOCALISATION SATELLITES

CESBIO

SINERGISE

JOANNEUM RESEARCH

cnes

DLR

GAFAG
an e-GEOS (ASI / Telespazio) Company

evenflow