AI4 copernicus

5th Open Call

Technical Documentation

Reinforcing the AI4EU Platform by Advancing Earth Observation Intelligence, Innovation and Adoption

Table of Contents

List of Tables

List of Figures

List of Terms & Abbreviations

Abbreviation	Definition
AC	Atmospheric Composition
ACD	Amplitude Change Detection
AoI	Area of Interest
AQ	Air Quality
ARD	Analysis Ready Data
AI	Artificial Intelligence
AIS	Automatic Identification System
BoA	Bottom of Atmosphere
C3S	Copernicus Climate Change Service
CAMS	Copernicus Atmosphere Monitoring Service
CD	Change Detection
CDS	Climate Data Store
CSV	Comma-Separated Values
CVA	Change Vector Analysis
DEM	Digital Elevation Model

DIAS	Data and Information Access Services
DL	Deep Learning
DTE	Digital Twin Earth
EC	European Commission
ECTL	Extract, Cleanse, Transform, Load
EO	Earth Observation
EU	European Union
GA	Grant Agreement
GAN	Generative Adversarial Network
GDAL	Geospatial Data Abstraction Library
GHG	Greenhouse gas
GPT	Graph Processing Tool
GRD	Ground Range Detected
IoT	Internet of Things
IW	Interferometric Wide
JRC	Joint Research Centre
JSON	JavaScript Object Notation
km	Kilometer
L1C	Level 1C
L2A	Level 2A
LAI	Leaf Area Index
LSTM	Long Short-Term Memory
ML	Machine Learning
MTC	Multi-Temporal Coherence
NN	Neural Network
OSM	Open Street Map
PM	Particulate Matter
S1	Sentinel-1
S2	Sentinel-2
SAR	Synthetic Aperture Radar
SLC	Single Look Complex
SM	Strip Map
SNAP	Sentinel Application Platform
SRGAN	Super-resolution GAN
SRTM	Shuttle Radar Topography Mission
TOPSAR	Terrain Observation Progressive Scans SAR
UTM	Universal Transverse Mercator
WGS	World Geodetic System
WKT	Well-known text
WP	Work Package

XML	eXtensible Markup Language
YAML	YAML Ain’t Markup Language

1 Quick Start

1.1 Overview

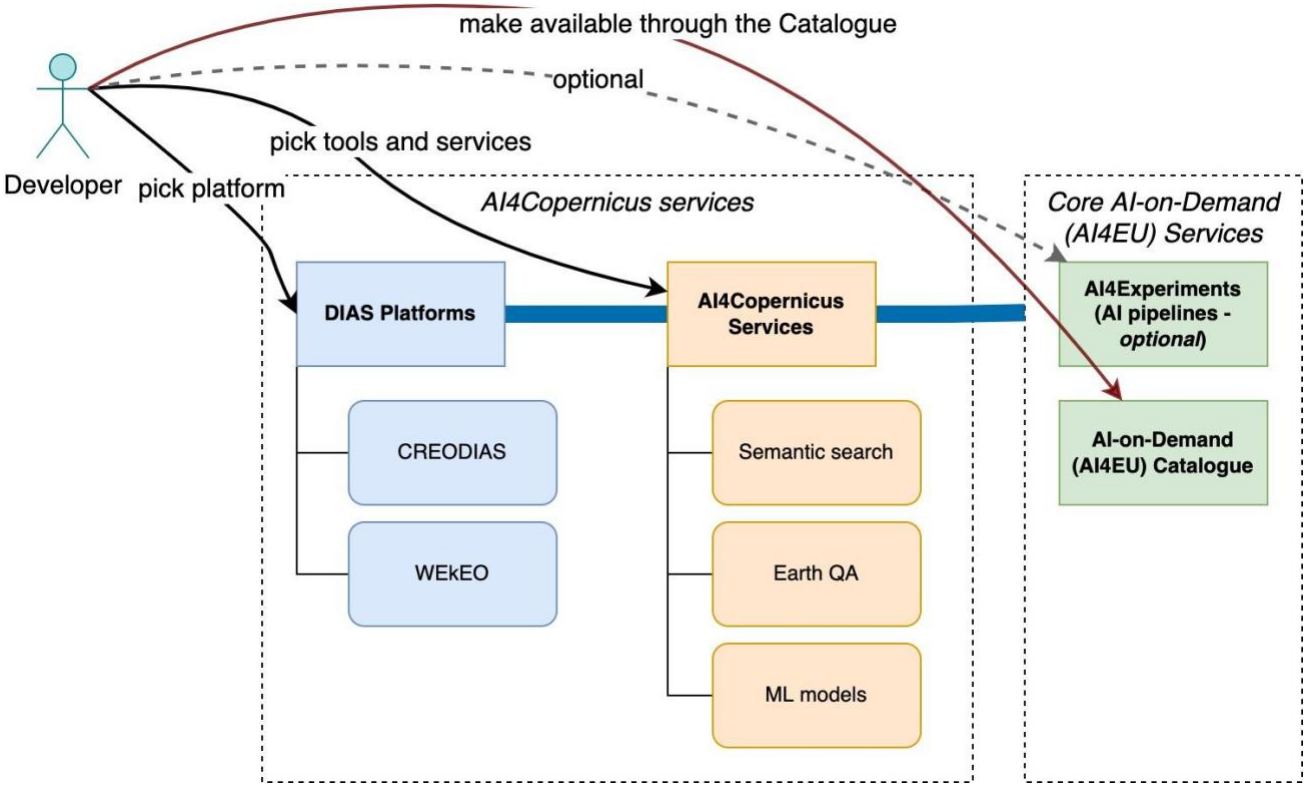

Figure 1. Ways of development in the AI4Copernicus environment

AI4Copernicus provides users with access to DIAS platforms and the AI-on-Demand Platform (AIoD, also known as AI4EU) resources in a streamlined way, offering support along the way.

The expected roadmap for interacting with AI4Copernicus resources is:

- **Develop** (CREODIAS / WEkEO or local/private resource)
- *Optional*: **Onboard** onto the AI4EU Experiments platform
  - **Refine** making use of other Experiments resources
  - Deploy the new workflow on CREODIAS or elsewhere
- **Publish** onto the public AI4EU catalogue

(1) We anticipate that, as a developer, you will develop your solution and models on CREODIAS - integration of more DIAS platforms is currently underway.

(Alternative or local resources may also be used for development, however in this case support and the integration of Copernicus data will not be provided by AI4Copernicus.)

CREODIAS/WEkEO will provide you with access to necessary cloud resources as well as to Copernicus datasets and other products.

(2) For exploring and experimenting AI4Copernicus provides a set of additional services and tools, outlined below, e.g. semantic searching.

(3) Once you have designed and built your model you can optionally onboard and publish it on the AI4EU public marketplace. This set of resources provide workflow and sharing functionality.

(4) As a final step, you are encouraged to publish your work on the AI4EU catalogue for other interested parties, researchers and practitioners to be able to discover it.

1.2 Getting access

1.2.1 AI4EU

The AI-on-Demand platform can be reached through https://www.ai4europe.eu. The AIoD catalogue is reachable at https://www.ai4europe.eu/research/ai-catalog. Successful bidders are expected to publish their finished products on this catalogue - more information will be provided upon success. Some useful links regarding publishing assets in the AIoD platform can be found at https://www.ai4europe.eu/education/education-catalog/publishing-contents-ai-demand-platform and https://aiondemand.readthedocs.io/.

The AI4Experiments platform is reachable via https://www.ai4europe.eu/development, where interested parties can register and documentation is provided. Further information and tutorials can be found at the relevant GitHub repository (https://github.com/ai4eu).

1.2.2 CREODIAS & WEkEO

Project users have access to resources present on the platforms like CREODIAS and WEkEO.

- *CREODIAS Access to platform and Resources*

To get access to CREODIAS resources in AI4Copernicus project, User should follow these steps:

- Please create an account on https://new.cloudferro.com/login You will then have access to CREODIAS WAW3-1 Public Cloud.
- If you register, please conntact the responsible CloudFfero contact point connected with AI4Copernicus Project. Your contact point will update information about billing your account and provisioning the project environment.
- Once CloudFerro confirms that the credits have been allocated and projects were, you can start using the resources available in the cloud environment

You can find flavours etc. which are available on WAW3-1 Public Cloud: https://creodias.eu/price-list (Figures 6-10)

You can find a lot of useful information (how to set up environment) in our FAQ: https://creodias.eu/faq

In some situation there may be a need to use technical support (doubts and questions related to the use of service): support@creodias.eu

- *WEkEO Access to platform and Resources*

To get access to WEkEO resources in AI4Copernicus project, User should follow these steps:

- Please create an account on https://wekeoelasticity.cloudferro.com/login You will then have access to WEkEO Elasticity Cloud.
- If you register, please conntact the responsible CloudFfero contact point connected with AI4Copernicus Project. Your contact point will update information about billing your account and provisioning the project environment.
- Once CloudFerro confirms that the credits have been allocated and projects were, you can start using the resources available in the cloud environment

You can find flavours etc. which are available on WAW3-1 Public Cloud: https://wekeo.docs.cloudferro.com/en/latest/static/pricelist.html

You can find a lot of useful information (how to set up environment) in our FAQ: https://wekeo.docs.cloudferro.com/en/latest/index.html

In some situation there may be a need to use technical support (doubts and questions related to the use of service): https://wekeo.docs.cloudferro.com/en/latest/Gstarted/helpdeskandsupport.html?highlight=support

1.2.3 AI4Copernicus

More information about the participating platforms can be found on the website of AI4Copernicus, under https://ai4copernicus-project.eu/platforms/.

1.3 Sample use-case

Let's consider a use case where we calculate the maximum value of a CO column in a given AOI (Area of Interest) and for a certain time period. To achieve this goal the user has access to AI tools to set up a pipeline which runs a workflow in an automated way. The result of this use-case is the value presented on the port of a container deployed in the Kubernetes cluster that is set up in the CREODIAS environment.

1.3.1 Prerequisites

1. Knowledge base for the example CO processing
    - Finder API, to look for relevant data. Documentation: https://creodias.eu/-/how-to-use-creodias-finder-
    - Help on using the S3 API can be found at the following resources:
        i. HOW TO ACCESS/LIST EO DATA USING BOTO3: https://creodias.eu/-/how-to-list-eo-data-using-boto3-?inheritRedirect=true&redirect=%2Ffaq-s3
        ii. HOW TO DOWNLOAD EO DATA FILE USING BOTO3: https://creodias.eu/-/how-to-download-a-eo-data-file-using-boto3-?inheritRedirect=true&redirect=%2Ffaq-s3

iii. EO DATA ACCESS - S3 OR NFS?: https://creodias.eu/-/eo-data-access-s3-or-nfs-?inheritRedirect=true&redirect=%2Ffaq-s3

iv. HOW TO ACCESS EO DATA AND OBJECT STORAGE USING S3CMD (LINUX): https://creodias.eu/-/how-to-access-eo-data-using-s3cmd-linux-v2?inheritRedirect=true&redirect=%2Ffaq-s3

- Interacting with the experimental AI4EU experiments requires knowledge on GRPC communication standards. This is provided at https://developers.google.com/protocol-buffers and https://grpc.io/
- For data acquisition in NetCDF format please refer to: https://gdal.org/index.html

2. CREODIAS Kubernetes cluster
   - kubectl setup for CREODIAS Kubernetes cluster
     Configuration is available after login to CREODIAS under:

     ~/.kube/config

Python 3: This use-case has been implemented in Python 3.

1.3.2 Steps

At the beginning the user needs to prepare a data broker to find relevant data products, and an analyzer to read data from product files. The user then needs to publish the relevant docker images in the image registry.

1. Information on model onboarding is presented at: https://www.youtube.com/watch?v=Ts4KqvvmkMg
2. Pipeline composition and local download: https://www.youtube.com/watch?v=gM-HRMNOi4w&t=6s
3. Configuration of kubectl to point execution environment in CREODIAS
4. Deployment and execution of the solution pipeline

For this particular use-case we prepare:

1. An EO Data broker that searches for Sentinel-5p's carbon monoxide data via the EO Finder API. This broker returns the file metadata to the pipeline designed on AI4EU experiments

2. A CO Max analyzer makes use of the EO data files using S3 API and makes available the calculated value via a REST API

The figure below outlines the procedure:

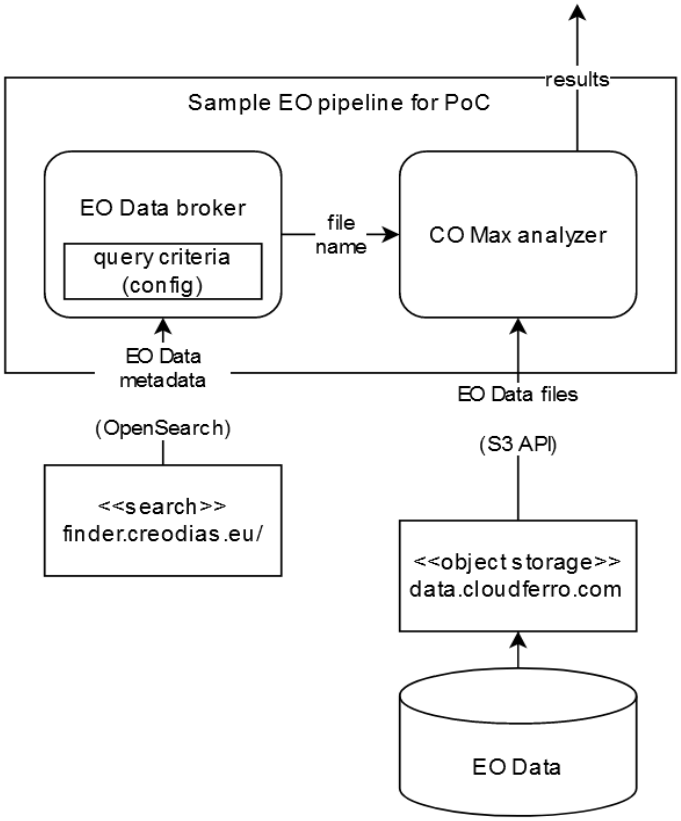

Figure 2. An overview of the pipeline of this example use-case

2 AI4Copernicus Services Overview

As part of our technical work in AI4Copernicus we will provide semantic searching over Copernicus data as well as pre-trained ML and related models to be used by our users and bidders. More information on how to access and use them will be provided here soon.

Moreover, The AI4Copernicus consortium provides a set of services and resources made available from the Security, Agriculture, Energy and Health communities for the open calls winners to facilitate the development of AI applications.

For more information on these services, please consult the last part of this report "*AI4Copernicus bootstrapping services and resources*".

2.1 Contacts for technical information

CREODIAS: https://cloudferro.com/en/why-cloudferro/contact/

AI4EU Experiments: ai4eu-experiments-support@iais.fraunhofer.de

2.2 Tools for transformation, querying, interlinking and federating big linked geospatial data

Linked Data lies at the heart of the Semantic Web and allows large scale integration over RDF resources. However, to make the Web of Data a reality, it is important to have the huge amount of data on the Web available in a standard format, reachable and manageable by Semantic Web tools. Furthermore, not only does the Semantic Web need access to data, but relationships among data should be made available, too, to create a Web of Data (as opposed to a sheer collection of datasets).

AI4Copernicus offers a linked data suite that allows users to handle linked data at every step of the linked data pipeline. These tools are available in the AI4EU Research Catalogue and are distributed as open source through the AI4EU website.

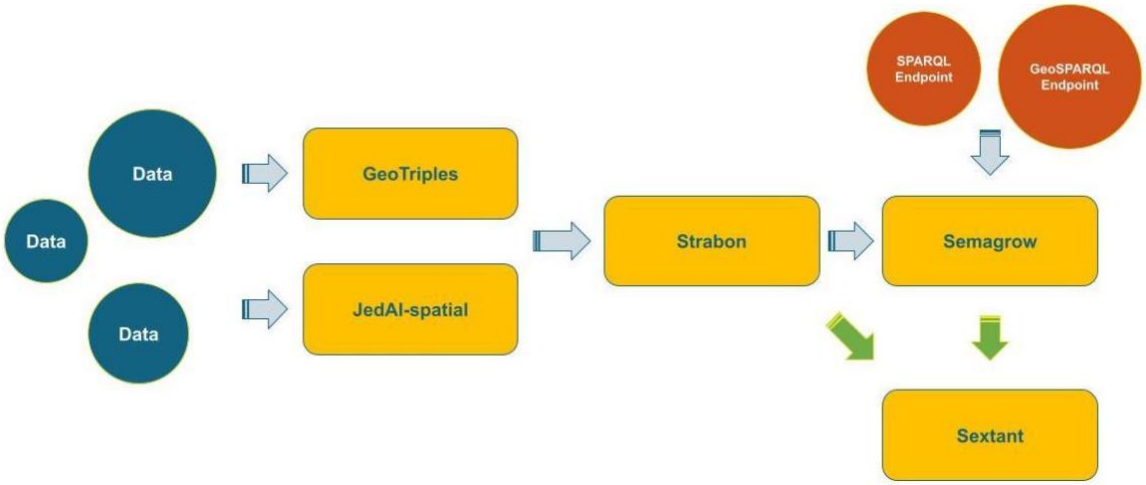

Figure 3. Linked Data pipeline

2.2.1 GeoTriples

GeoTriples[1] allows the user to transform geospatial data from their original formats into RDF. The software itself is GDPR compliant. Documents are processed locally and all data remains on the user's local computer. However, the users must ensure that they have the authority to store and process the documents, for example if they contain personal data or other sensitive GDPR-relevant information.

GeoTriples is a semi-automated tool that enables the automatic transformation of geospatial data into RDF graphs using state of the art vocabularies like GeoSPARQL, but at the same time, it is not tightly coupled to a specific vocabulary. The transformation process comprises three steps. First, GeoTriples generates automatically extended R2RML or RML mappings for transforming data that reside in spatially-enabled databases or raw files into RDF. As an optional second step, the user may revise these mappings according to her needs e.g., to utilize a different vocabulary. Finally, GeoTriples processes these mappings and produces an RDF graph. The input formats supported by GeoTriples are spatially-enabled relational databases (PostGIS and MonetDB), ESRI shapefiles, XML documents

[1] https://www.ai4europe.eu/research/ai-catalog/geotriples

following a given schema (hence GML documents as well), KML documents, JSON and GeoJSON documents and CSV documents.

GeoTriples is used in the first step of the Linked Data pipeline and assists users in the transformation of data in the RDF format.

2.2.2 Strabon

Strabon[2] is a spatiotemporal RDF store. You can use it to store linked geospatial data that changes over time and pose queries using two popular extensions of SPARQL. Strabon supports spatial datatypes enabling the serialization of geometric objects in OGC standards WKT and GML. It also offers spatial and temporal selections, spatial and temporal joins, a rich set of spatial functions similar to those offered by geospatial relational database systems and support for multiple Coordinate Reference Systems. Strabon can be used to model temporal domains and concepts such as events, facts that change over time etc. through its support for valid time of triples, and a rich set of temporal functions.

Strabon extends the well-known RDF store Sesame, allowing it to manage both thematic and spatial data expressed in stRDF and stored in the PostGIS spatially enabled DBMS. Strabon implements fully the Core, Geometry Extension and Geometry Topology Extension components of GeoSPARQL. It supports all three topological relation classes defined by GeoSPARQL (OGC-SFA, Egenhofer, RCC8), both geometry serializations (WKT, GML) and multiple CRS.

Strabon sits at the core of the Linked Data pipeline and is used to store our data once they are made available in the RDF format.

2.2.3 JedAI

JedAI[3] comprises a set of domain-independent, state-of-the-art techniques that apply to any domain. At their core lies an approximate, schema-agnostic functionality based on blocking for high scalability. JedAI constitutes an open source, high scalability toolkit that offers out-of-the-box solutions for any data integration task, e.g., Record Linkage, Entity Resolution and Link Discovery. At its core lies a set of domain-independent, state-of-the-art techniques that apply to both RDF and relational data. These techniques rely on an approximate, schema-agnostic functionality based on (meta-)blocking for high scalability.

JedAI-WebApp is a GUI developed with Spring (boot+ MVC) and ReactJS that facilitates the execution of JedAI. It enables the user to construct its desired workflow by sequentially selecting the algorithm(s) of each step. Furthermore, JedAI-WebApp provides the following capabilities:

- Multiple data input interfaces
- Data (entities) Exclusion
- Data Exploration
- Automatic configuration of the algorithms' parameters. User can specify the values of the parameters or he can leave them to the system to detect which parameters produce the best

[2] https://www.ai4europe.eu/research/ai-catalog/strabon

[3] https://www.ai4europe.eu/research/ai-catalog/jedai

results. The detection of the ideal parameters is performed by Grid Search or by Random Search.
- Detailed Results and display of the logs
- Exploration of the data and results.

Furthermore, it facilitates the benchmarking of different workflows or configurations over a particular dataset through the workbench window, which summarizes the outcome of all runs and maintains details about the performance and the configuration of every step. JedAI can be used to discover links between different sources.

JedAI has been extended with new algorithms for interlinking big geospatial data sources. We developed a new module, called JedAI-spatial, which serves both as an open-source library of the state-of-the-art works in the field and as an open-source system that implements new methods that go beyond existing works in terms of efficiency and scalability. JedAI-spatial has the following unique characteristics:

- It organizes the main algorithms for Geospatial Interlinking into a novel taxonomy that facilitates their use and adoption by practitioners and researchers.
- Its intuitive user interface supports both novice and expert users: they simply have to select one of the available methods per workflow step and optionally configure it. It also simplifies the benchmarking of the main algorithms in the field through the workbench window that summarizes the performance of the algorithms executed so far. This is a crucial task for identifying the best approach for a particular task at hand, given that the experimental analyses in the literature are usually limited with respect to the variety in datasets or the baseline methods.
- Its modular and extensible architecture allows for easily incorporating improvements to all algorithms.
- It optimizes the implementation of existing algorithms, some of which have not been applied to Geospatial Interlinking before.
- It conveys new techniques that achieve competitive performance.

2.2.4 Semagrow

Semagrow[4] is a dynamic data integration system that presents multiple (syntactically or semantically) heterogeneous datasets as a unified, homogeneous virtual dataset. Semagrow provides a federated query processor that allows combining, cross-indexing and, in general, making the best out of all public data, regardless of their size, update rate, and schema. In this manner, it offers a single SPARQL endpoint that serves data from remote data sources and that hides from client applications heterogeneity in both form (federating non-SPARQL endpoints) and meaning (transparently mapping queries and query results between vocabularies).

[4] https://www.ai4europe.eu/research/ai-catalog/semagrow

Semagrow has been used to integrate diverse datasets in multiple domains and applications. Among others, meteorological, land-usage, water availablity, and crops data for food security; meteorological, GIS, and dispersion modelling data for risk estimation, biology and pharmacology datasets for pharmacological research.

2.2.5 Sextant

Sextant[5] is a web based and mobile ready platform for visualizing, exploring and interacting with linked geospatial data. The core feature of Sextant is the ability to create thematic maps by combining geospatial and temporal information that exists in a number of heterogeneous data sources, ranging from standard SPARQL endpoints, to SPARQL endpoints following the standard GeoSPARQL defined by the Open Geospatial Consortium (OGC), or well-adopted geospatial file formats, like KML, GML and GeoTIFF.

2.2.6 EarthQA

EarthQA[6] is the user-friendly semantic search over EO data that provides users with the facility to find the satellite images by typing their requirements in natural language . EarthQA takes natural language questions as an input, converts it into SPARQL query and by executing the generated SPARQL query over EO metadata stored in rdf form provides users with the links to the satellite images to download. If user wants to search and download a satellite image based on specific properties, user can write the requirements in form of natural language query e.g. “Find Sentinel-1 products that show Etna in March 2018” and EarthQA will provide a link to download the images by clicking on the popup which can be seen in the following image.

[5] https://www.ai4europe.eu/research/ai-catalog/sextant

[6] http://teleios4.di.uoa.gr:15434/

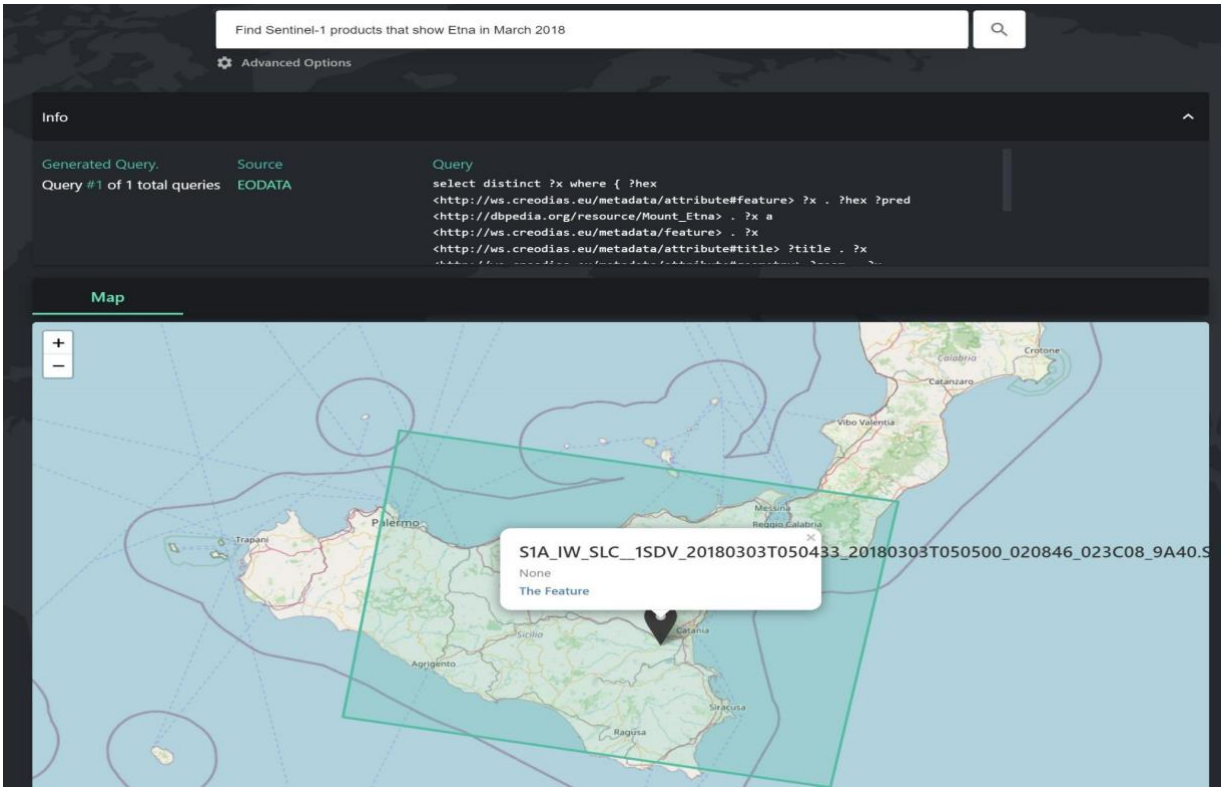

Figure 4. EarthQA Demo

3 AI4Copernicus Bootstrapping Services and Resources

3.1 Introduction

The AI4Copernicus consortium provides a set of services and resources made available from the Security, Agriculture, Energy and Health communities for the open calls winners.

The development of these bootstrapping services aimed to reduce the time and resources of the bidders in different tasks as data access (EO and ancillary data), pre-processing, labelling datasets, ML algorithm definition. The AI4Copernicus consortium support to the bidders allows to address open calls winner's effort on the development of innovative services based on AI.

Each service is documented in the section of the domain responsible of its deployment: following a cooperative approach, each service can be used by different domains, if relevant. The description of the services (e.g. the purpose, the input needed and the output produced) referred to the application deployed in the AI4Copernicus infrastructure.

3.2 Methodology and Structure of the services description

While the domain responsible was in charge to deploy the specific services, other domains can use the different services if considered useful for their own domain (e.g. pre-processing services from Security on Sentinel-1 and Sentinel-2 are useful also for other domains). This approach avoided the duplication of efforts to develop the same service and the simplification for the users.

Each section briefly introduces the approach followed to deploy the services and listed all the services available for the first batch of open calls. The description of the services describes the purpose, the structure, the input needed, the output produced and explains how to run them.

3.3 Summary table of services and resources

The tables below summarize the services and resources described in the next sections of this document.

The use of the services has been formalized in the Subgrant Agreement between each project and AI4Copernicus. In Article 2 of the SubGrant Agreement, *all bootstrapping services* (see section 1.3.1 < PartB< Annex1 [SubProject] < SubGrant Agreement) *are offered solely during the 16 month period commencing from the Effective Date. Bootstrapping services beyond this term are subject to a separate agreement with the relevant AI4Copernicus partner and lie outside the scope of the SubGrant Agreement and the obligations of the AI4Copernicus Consortium there of.*

In the case of the service "Deep network for pixel-level classification of S2 patches", provided by Thales Six, an additional licence agreement to be signed was provided to the open call winners in order to access and use the related bootstrapping service.

Table 1. Summary table of services

Resource	Summary	Domain	Origin
Sentinel-1 GRD pre-processing	This pipeline processes a S1 GRD product in native format to generate a terrain corrected image representing the calibrated backscatter in GeoTiff format.	Security/ General	AI4Copernicus
Sentinel-1 SLC pre-processing	This pipeline processes a S1 SLC product in native format to generate a terrain corrected image representing the calibrated backscatter in GeoTiff format.	Security/ General	AI4Copernicus
Sentinel-2 pre-processing	This pipeline processes a S2 product in native format to generate a product with a common resolution for all the bands in GeoTiff format. The process allows to apply a land/sea mask and a cloud mask in order to have an output product ready for analysis.	Security/ General	AI4Copernicus
Sentinel-1 Change detection– Amplitude Change Detection and Multi-temporal Coherence	This pipeline processes pairs of S1 SLC products in native format to generate a series of products to assess the changes between both images. These products include: coherence, ACD (Amplitude Change Detection), MTC (Multi-Temporal Coherence) and binary mask of changes.	Security/ General	AI4Copernicus

Sentinel-2 Change Detection	This pipeline computes (and classifies) the changes using as input a pair of S2-L2A products by using the Change Vector Analysis approach.	Security/ General	AI4Copernicus
Vector data of human features	SatCen has pre-processed and ingested in a data model the OSM data and can provide the data as a service in the scope of the project	Security	AI4Copernicus
Deep network for pixel-level classification of S2 patches	This service provides functionality for users to train a custom pixel-level classifier of Sentinel 2 patches. For example users can train a classifier for crop types (corn, sunflower, wheat, etc), land cover (urban vs. natural, water vs land), road extraction (road vs other).	Agriculture/ General	AI4Copernicus
Harmonization of pre-processed Time Series of Sentinel-2 data	The harmonization of pre-processed time series of Sentinel-2 data considers a statistic-based approach that computes the median for each pixel in the different images acquired in a particular month. The pixel composite approach to mosaic generation provides consistent results at large scale, allowing the processing of harmonized acquisitions.	Agriculture	AI4Copernicus
Long Short-Term Memory Neural Network for Sentinel-2	The Long Short Term-Memory architecture can be trained using samples selected by the user. The service exploits the data given by the user to train from scratch an LSTM and stores the resulting weights. Several parameters are exposed to allow the user to custom the model	Agriculture	AI4Copernicus
Pre-Trained Long Short-Term Memory	The pre-trained Long Short Term-Memory architecture is already trained using the TimeSen2Crop database and is available in .h5 format. The service exploits a pre-trained architecture to classify the specified tile harmonized using the monthly composite approach	Agriculture	AI4Copernicus
Probabilistic downscaling of CAMS air quality model data	This service generates high-resolution (currently ~ 10km) air quality maps from low-resolution (~40 - 80km) CAMS model (re)analysis and/or forecast output	Health	AI4Copernicus

Table 2. Summary table of datasets

Resource	Summary	Domain	Origin
TimeSen2Crop	TimeSen2Crop is a pixel-based dataset made up of more than 1 million crop type samples of Sentinel-2 time series. The dataset includes atmospherically corrected images and reports the snow, shadows, and clouds information per labelled unit, as well as the spectral signature of the samples of nine Sentinel-2 spectral bands at 10m of spatial resolution.	Agriculture	AI4Copernicus
Energy datasets	Meteorological data: ERA5 Example: offshore wind farms are located (training data) JRC Open Power Plants Database (JRC-PPDB-OPEN) Open data from the floating offshore wind farm, Hywind Scotland	Energy	External references

4 Security bootstrapping services and resources

4.1 Introduction

The Security Bootstrapping services and resources have been developed considering the objective of the open calls, which was *"the development of EO applications leveraging on AI algorithms to detect, identify and/or predict features and events in response to current Security challenges. The applications are expected to exploit EO data, in conjunction with relevant collateral data sources as suitable (e.g. geolocalization, AIS, statistical data, climate/weather, in-situ sensors…) with the use of the latest technologies, also contributing to shape the development of a Digital Twin Earth (DTE) for Security".*

The target of the open calls highlights the need of AI algorithms to detect changes using EO and collateral data. To build the above-mentioned services, data and resources coming from EO repositories (e.g. DIASes) needed to be pre-processed for further advanced processing. The AI4Copernicus consortium therefore decided to contribute with the provision of *Analysis Ready Data (ARD)* and possible training datasets for AI algorithms. The development of these bootstrapping services aimed to reduce the time and resources of the bidders in the data preparation and allow them to focus on the development of innovative services based on AI. Moreover, in-house change detection algorithms were also provided as benchmarks to compare the accuracy of possible change detection algorithms provided by bidders.

The following sections describe the services developed by SatCen in the frame of the Security domain.

4.2 Services

4.2.1 Sentinel-1 GRD pre-processing

4.2.1.1 Summary

The Sentinel-1 GRD pre-processing pipeline is available as a dockerized application that can be executed in any environment with a properly configured Docker client.

This pipeline processes a S1 GRD product in native format to generate a terrain corrected image representing the calibrated backscatter in GeoTiff format. Several parameters will be exposed to the users (e.g. final resolution, polarization and projection). Where possible, a standard value will be set, to facilitate the use by the less-expert users.

4.2.1.2 Input

The input of this pipeline is a Level-1 Ground Range Detected (GRD) product in its native SENTINEL-SAFE [6] format (zipped or unzipped product).

Regarding the acquisition mode, they are supported:

- Stripmap (SM): *Stripmap (SM) mode acquires data with an 80 km swath at slightly better than 5 m by 5 m spatial resolution (single look). The ground swath is illuminated by a continuous sequence of pulses while the antenna beam is pointing to a fixed azimuth angle and an*

[6] https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/data-formats/sar-formats

approximately fixed off-nadir angle (this is subject to small variations because of roll steering). SM images have continuous along track image quality at an approximately constant incidence angle.[7]

- Interferometric Wide Swath (IW): *the main acquisition mode over land and satisfies the majority of service requirements. It acquires data with a 250 km swath at 5 m by 20 m spatial resolution (single look). IW mode captures three sub-swaths using Terrain Observation with Progressive Scans SAR (TOPSAR). With the TOPSAR technique, in addition to steering the beam in range as in ScanSAR, the beam is also electronically steered from backward to forward in the azimuth direction for each burst, avoiding scalloping and resulting in homogeneous image quality throughout the swath*[8].

4.2.1.3 Exposed Parameters

Table 3. Exposed parameters in Sentinel-1 GRD pre-processing

Parameter	Valid values	Default Value
CalibrationOutput: backscatter convention selected for the output. Each value uses a different reference area. Values ending with DB will be converted to decibel.	Sigma, Gamma, Beta, SigmaDB, GammaDB, BetaDB	Sigma
Polarization	VV,VH… (valid values depend on the specific product)	VV
Speckle-filter (+ specific parameters if needed)	Supported by SNAP	Lee Sigma
AoI: Area of Interest	WKT polygon or path to vector file	None
Resolution: output resolution in meters. The minimum value recommended is the default value (10m for IW products and 5 m for SM products)	Any in meters	10m (IW), 5m (SM)
Land/Sea mask: type of pixels to be removed considering SRTM3Sec. If “Sea” is selected, all values with elevation=0 in SRTM 3Sec will be set to NoData.	Land/Sea/None	None
Projection: output projection	Any supported by SNAP	WGS84
Output format: output format	Supported by GDAL and SNAP	GeoTiff

[7] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/stripmap

[8] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/interferometric-wide-swath

4.2.1.4 Processing

The pipeline has been designed using SNAP[9], the common software platform and host for the Sentinel Toolboxes.

One of the components of SNAP, the Graph Processing Tool (GPT), allows to execute SNAP in batch-mode from command-line, with the possibility to create complex workflows using most of the SNAP operators (e.g. Readers, Subset, Reprojection, Band Math, Filters, Calibration…).

In the case of this pipeline, the SNAP graph executed is represented in Figure 1.

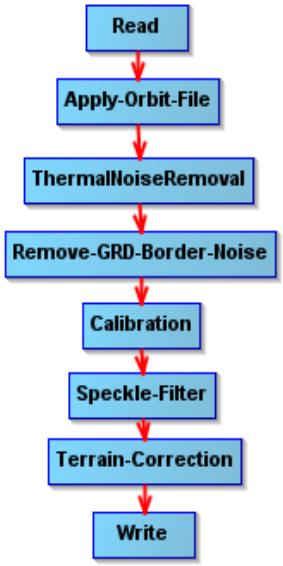

Figure 5. S1 GRD Processing graph

Where:

- Read: the operator in charge of reading a product to the SNAP internal data model.
- Apply-Orbit-File: search, download and apply the corrected information about the orbit that is provided some days after the S1 acquisition to improve the geolocation.
- ThermalNoiseRemoval: removes thermal noise
- Remove-GRD-Border-Noise: remove border noise
- Calibration: convert pixel values to calibrated sigma0, gamma0 or beta0
- Speckle-Filter: applies filter to reduce speckle
- Terrain-Correction: orthorectify the product
- Write: write the output product to the desired format.

(More information about the specific operators can be found in the SNAP help and documentation.)

[9] https://step.esa.int/main/toolboxes/snap/

4.2.1.5 Output

The output is a GeoTiff (by default) terrain-corrected image with one float32 band representing the calibrated backscatter (sigma0, gamma0 or beta0 depending on the selected parameters).

4.2.1.6 How to use

Minimum requirements: 16GB of RAM.

Inside the docker, the pipeline can be found in `/app/pipelines` and can be executed with the following command:

```
S1-GRD-preprocess --input "VALUE" [--calibration "VALUE"] [--polarization "VALUE"]
[--speckle "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--landseamask "VALUE"]
[--projection "VALUE"] [--output_format "VALUE"] --output_path "VALUE"
```

It can also be launched with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -v [local_path]:[container_path] DOCKER_IMAGE S1-GRD-preprocess --
input "VALUE" [--calibration "VALUE"] [--polarization "VALUE"] [--speckle "VALUE"]
[--AoI "WKT"] [--resolution "VALUE"] [--landseamask "VALUE"] [--projection
"VALUE"] [--output_format "VALUE"] -outdir "VALUE"
```

In the case of any customization is needed in graph, it can be found in the source code in the docker and could be adapted by the users and executed directly using the SNAP's Graph Processing Tool (gpt).

4.2.2 Sentinel-1 SLC pre-processing

4.2.2.1 Summary

The Sentinel-1 SLC pre-processing pipeline is available as a dockerized application that can be executed in any environment with a properly configured Docker client.

This pipeline processes a S1 SLC product in native format to generate a terrain corrected image representing the calibrated backscatter in GeoTiff format. Several parameters are exposed (e.g. final resolution, polarization and projection), including when possible, a default value to facilitate the use by non-expert users.

4.2.2.2 Input

The input of this pipeline is a Level-1 Single Look Complex (SLC) product in its native SENTINEL-SAFE [10] format (zipped or unzipped product are both supported).

Regarding the acquisition mode, they are supported SM and IW (see 2.2.1.2 for a description of these modes).

[10] https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/data-formats/sar-formats

4.2.2.3 Exposed Parameters

Table 4. Exposed parameters in Sentinel-1 SLC pre-processing

Parameter	Valid values	Default Value
CalibrationOutput: backscatter convention selected for the output. Each value uses a different reference area. Values ending with DB will be converted to decibel.	Sigma, Gamma, Beta, SigmaDB, GammaDB, BetaDB	Sigma
Polarization	VV,VH… (valid values depend on the specific product)	VV
Speckle-filter (+ specific parameters if needed)	Supported by SNAP	Lee Sigma
AoI: Area of Interest	WKT polygon or path to vector file	None
Resolution: output resolution in meters. The minimum value recommended is the default value (10m for IW products and 5 m for SM products)	Any in meters	10m (IW), 5m (SM)
Land/Sea mask: type of pixels to be removed considering SRTM3Sec. If “Sea” is selected, al values with elevation=0 in SRTM 3Sec will be set to NoData.	Land/Sea/None	None
Projection: output projection	Any supported by SNAP	WGS84
Output format: output format	Supported by GDAL and SNAP	GeoTiff

4.2.2.4 Processing

The main pipeline has been designed using SNAP. The SNAP graph executed for IW products is represented in Figure 2. For SM products, the graph varies a little since operators like TOPSAR-Deburst are not needed.

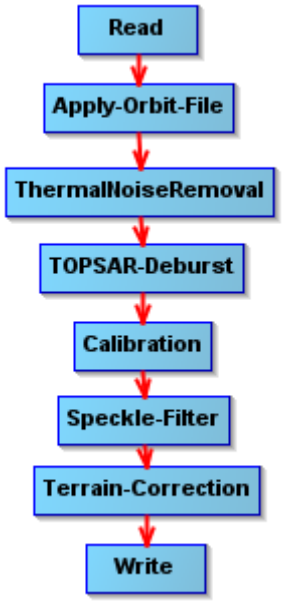

Figure 6. S1 SLC pre-processing graph for IW products.

Where:

- Read: the operator in charge of reading a product to the SNAP internal data model.
- Apply-Orbit-File: search, download and apply the corrected information about the orbit that is provided some days after the S1 acquisition to improve the geolocation.
- ThermalNoiseRemoval: removes thermal noise
- TOPSAR-Deburst: merge the bursts
- Calibration: convert pixel values to calibrated sigma0, gamma0 or beta0
- Speckle-Filter: applies filter to reduce speckle
- Terrain-Correction: orthorectify the product
- Write: write the output product to the desired format.

(More information about the specific operators can be found in the SNAP help and documentation.)

4.2.2.5 Output

The output is a GeoTiff (by default) terrain-corrected image with one float32 band representing the calibrated backscatter (sigma0, gamma0 or beta0 depending on the selected parameters).

4.2.2.6 How to use

Minimum requirements: 16GB of RAM.

Inside the docker, the pipeline can be found in `/app/pipelines` and can be executed with the following command:

```
S1-SLC-preprocess --input "VALUE" [--calibration "VALUE"] [--polarization "VALUE"]
[--speckle "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--landseamask "VALUE"]
[--projection "VALUE"] [--output_format "VALUE"] --outdir "VALUE"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -v [local_path]:[container_path] DOCKER_IMAGE S1-SLC-preprocess --input "VALUE" [--calibration "VALUE"] [--polarization "VALUE"] [--speckle "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--landseamask "VALUE"] [--projection "VALUE"] [--output_format "VALUE"] --outdir "VALUE"
```

In the case of any customization is needed in graph, it can be found in the docker and could be adapted by the users and executed directly using *gpt*.

4.2.3 Sentinel-2 pre-processing

4.2.3.1 Summary

The Sentinel-2 pre-processing pipeline is available as a dockerized application that can be executed in any environment with a properly configured Docker client.

This pipeline processes a S2 product in native format to generate a product with a common resolution for all the bands in GeoTiff format. The process allows to apply a land/sea mask and a cloud mask in order to have an output product ready for analysis.

Several parameters are exposed (e.g. DEM, cloud mask type), including when possible, a default value to facilitate the use by non-expert users.

4.2.3.2 Input

The input of this pipeline is a Sentinel-2 L2A product in its native SENTINEL-SAFE format (zipped or unzipped products are both supported).

They are also supported Sentinel-2 L1c products in its native SENTINEL-SAFE format. In this case, *sen2cor* tool is used internally to process the L2A product before preprocessing.

4.2.3.3 Exposed Parameters

Table 5. Exposed parameters in Sentinel-2 pre-processing.

Parameter	Valid values	Default Value
Resolution: output resolution in meters. The minimum value recommended is the default value (10m)	Any in meters	10
Bands	Any combination of S2 L2A bands: B1,B2, B3, B4, B5, B6, B7, B8, B8A, B9, B10, B11, B12	B2, B3, B4, B8
Land/Sea mask: type of pixels to be removed considering SRTM3Sec. If "Sea" is selected, al values with elevation=0 in SRTM 3Sec will be set to NoData.	Land/Sea/None	None
AoI: Area of Interest	WKT polygon or path to vector file	None

CloudMask: assign NoData to cloudy pixels according to the cloud mask type selected	L1C/L2A/other?/None	None
Upsampling method	Nearest/Bilinear/Bicubic	Nearest
Downsampling method	First,Min,Max,Mean,Median	First
Flag Downsampling method	First,FlagAnd,FlagOr,FlagMedianAnd,FlagMedianOr	First
Projection: output projection	Any supported by SNAP	UTM(Auto)
Output format: output format	Supported by GDAL and SNAP	GeoTiff

4.2.3.4 Processing

The main workflow has been designed using SNAP. The SNAP graph executed is represented in Figure 3.

Figure 7. S2 pre-processing graph.

Where:

- Read: the operator in charge of reading a product to the SNAP internal data model.
- S2Resampling: this operator resamples the product to a common resolution.
- Land/Sea Mask: It applies land/sea mask based on srtm 3sec.
- BandMaths: it is used to compute the cloud mask when needed.
- Subset: it filters out the non-desired bands and crop to the AoI.
- Reproject: reprojects to the selected projection.
- Write: write the output product to the desired format.

(More information about the specific operators can be found in the SNAP help and documentation.)

4.2.3.5 Output

The output is a GeoTiff (by default) terrain-corrected image containing the selected bands. Depending on the selected parameters, pixels in sea/land or/and cloudy have been set to NoData.

4.2.3.6 How to use

Minimum requirements: 16GB of RAM.

Inside the docker, the pipeline can be found in `/app/pipelines` and can be executed with the following command:

```
S2-preprocess --input "VALUE" [--bands "XX,XX,XX"] [--landseamask "VALUE"] [--cloudmask "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--upsampling "VALUE"] [--downsampling "VALUE"] [--projection "VALUE"] [--output_format "VALUE"] --outdir "VALUE"
```

A concrete example could be:

```
S2-preprocess --input
"/2/S2B_MSIL2A_20230106T111349_N0509_R137_T30TUL_20230106T125051.SAFE" --AoI
"POLYGON((-5.25080726428997 40.89918079640174,-5.222971610006687
40.89929182499685, -5.225321849946173 40.87724899587295,-5.256095304153813
40.87763772395914,-5.25080726428997 40.89918079640174))" --bands "B4,B8" --outdir
"/1"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -v [local_path]:[container_path] DOCKER_IMAGE  S2-preprocess --input "VALUE" [--bands "XX,XX,XX"] [--landseamask "VALUE"] [--cloudmask "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--upsampling "VALUE"] [--downsampling "VALUE"] [--projection "VALUE"] [--output_format "VALUE"] --outdir "VALUE"
```

In the case of any customization is needed in graph, it can be found in the docker and could be adapted by the users and executed directly using *gpt*.

4.2.4 Sentinel-1 Change detection– Amplitude Change Detection and Multi-temporal Coherence

4.2.4.1 Summary

The Sentinel-1 Change Detection resource is available as a dockerized application that can be executed in any environment with a properly configured Docker client.

This pipeline processes pairs of S1 SLC products in native format to generate a series of products to assess the changes between both images. These products include:

- the coherence (the amplitude of correlation between the images),
- the ACD (Amplitude Change Detection), which is a RGB composite of the backscatter of the input images

- the MTC (Multi-Temporal Coherence), which is a RGB composite of the backscatters and the coherence
- binary mask of changes

Several parameters are exposed (e.g. resolution, speckle), including when possible, a default value to facilitate the use by non-expert users.

4.2.4.2 Input

The input of this pipeline is a pair of Sentinel-1 SLC product in their native SENTINEL-SAFE format (zipped or unzipped product are both supported). The inputs shall be acquired with the same acquisition geometry (i.e. same relative orbit), which means in practice that they are acquired in a multiple of six days (6, 12, 18 days) of difference (considering both satellites, S1A and S1B). The validity of the inputs will be checked by the pipeline and when the requirements are not fulfilled, an error will be raised.

4.2.4.3 Exposed Parameters

Table 6. Exposed parameters in S1 Change detection– Amplitude Change Detection and Multi-temporal Coherence

Parameter	Valid values	Default Value
Resolution: output resolution in meters. The minimum value recommended is the default value (10m for IW products and 5 m for SM products)	Any in meters	10m (IW), 5m (SM)
CalibrationOutput: backscatter convention selected for the output. Each value uses a different reference area.	Sigma, Gamma, Beta	Sigma
Polarization	VV,VH… (valid values depend on the specific product)	VV
Land/Sea mask: type of pixels to be removed considering SRTM3Sec. If “Sea” is selected, all values with elevation=0 in SRTM 3Sec will be set to NoData.	Land/Sea/None	None
AoI: Area of Interest	WKT polygon or path to vector file	None
Speckle-filter (+ specific parameters if needed)	Supported by SNAP	Lee Sigma
Projection: output projection	Any supported by SNAP	WGS84
Output format: output format	Supported by GDAL and SNAP	GeoTiff

4.2.4.4 Processing

The pipeline has been designed mostly using SNAP and GDAL. The SNAP graphs executed to generate the coherence are represented in Figure 4 and Figure 5.

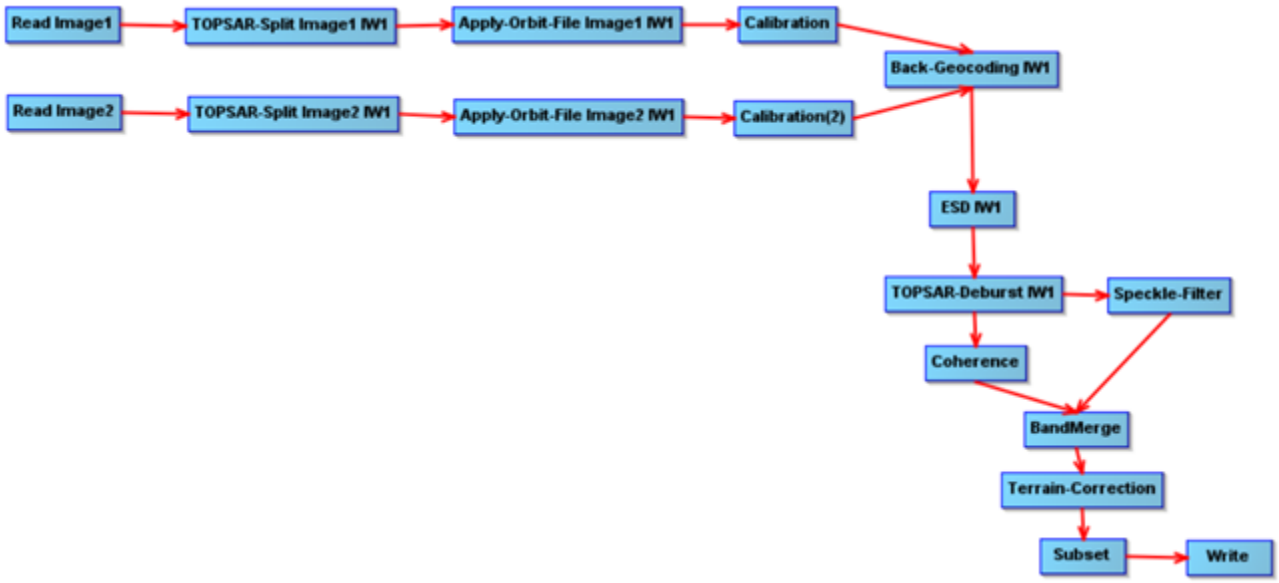

Figure 8. Sentinel-1 IW processing.

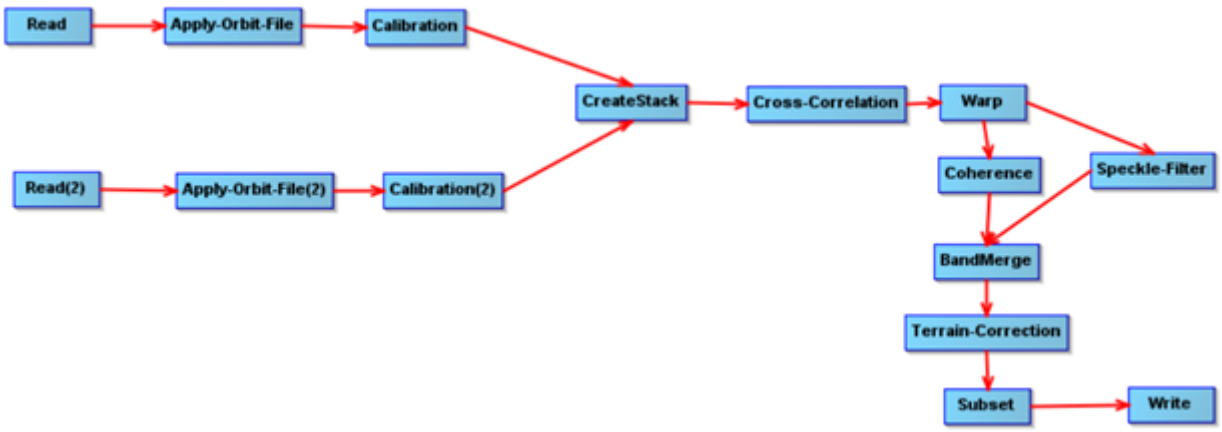

Figure 9. Sentinel-1 SM processing.

Where:

- Read: the operator in charge of reading a product to the SNAP internal data model.
- Apply-Orbit-File: search, download and apply the corrected information about the orbit that is provided some days after the S1 acquisition to improve the geolocation.
- TOPSAR-Split: split the product using the indicated subswath.
- Back-Geocoding: co-register the input products.
- ESD: Enhance the coregistration when more than one burst have been selected.

- Calibration: convert pixel values to calibrated sigma0, gamma0 or beta0
- Speckle-Filter: applies filter to reduce speckle
- Coherence: estimates the coherence
- Terrain-Correction: orthorectify the product
- CreateStack + Cross-Correlation + Warp: coregister the input products.
- Write: write the output product to the desired format.

(More information about the specific operators can be found in the SNAP help and documentation.)

4.2.4.5 Output

The outputs are:

- Coherence: GeoTiff image that represents the amplitude of correlation between the images. The pixel type is float32.
- Backscatters: 2 GeoTiff products (one for each of the input images) with one float32 band representing the calibrated backscatter.
- ACD: a RGB composite of the backscatter of the input images.
    - GeoTiff with three bands. R: backscatter of image 1; G : backscatter of image 2; B: backscatter of image 2
    - Pixel type is Byte, where byte value is computed by data conversion of float values using a linear interpolation taking as min and max values the percentiles 2.5 and 7.5.
- MTC (Multi-Temporal Coherence): a RGB composite of the backscatters and the coherence
    - GeoTiff with three bands. R: backscatter of image 1; G : backscatter of image 2; B: coherence
    - Pixel type is Byte, where byte value is computed by data conversion of float values using a linear interpolation taking as min and max values the percentiles 2.5 and 7.5 for backscatter band. For the coherence band the linear conversion is (0,1)->(0,255)
- Binary mask of changes. The pixel type is byte.

4.2.4.6 How to use

Minimum requirements: 32GB of RAM.

Inside the docker, the pipeline can be found in `/app/pipelines` and can be executed with the following command:

```
S1-CD --input1 "VALUE" --input2 "VALUE" [--calibration "VALUE"] [--polarization
"VALUE"] [--landseamask "VALUE"] [--speckle "VALUE"] [--AoI "WKT"] [--resolution
"VALUE"] [--projection "VALUE"] [--output_format "VALUE"] --outdir "VALUE"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -v [local_path]:[container_path] DOCKER_IMAGE  S1-CD --input1 "VALUE"
--input2 "VALUE" [--calibration "VALUE"] [--polarization "VALUE"] [--landseamask
"VALUE"] [--speckle "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--projection
"VALUE"] [--output_format "VALUE"] --outdir "VALUE"
```

If any customization is needed in the processing graphs, they can be found in the docker and could be adapted by the users and executed directly using *gpt*.

4.2.5 Sentinel-2 Change detection

4.2.5.1 Summary

The Sentinel-2 Change Detection pipeline is available as a dockerized application that can be executed in any environment with a properly configured Docker client.

This pipeline computes (and classifies) the changes using as input a pair of S2-L2A products by using the Change Vector Analysis approach.

Several parameters are exposed (e.g. resolution, bands, number of change classes), including when possible, a default value to facilitate the use by non-expert users.

4.2.5.2 Input

The input of this pipeline is a pair of Sentinel-2 L2A products in their native SENTINEL-SAFE format (zipped or unzipped products are both supported). The inputs shall correspond to the same tile (e. g. same relative orbit).

4.2.5.3 Exposed Parameters

Table 7. Exposed parameters in Sentinel-2 Change detection.

Parameter	Valid values	Default Value
Resolution: output resolution in meters. The minimum value recommended is the default value (10m for IW products and 5 m for SM products)	Any in meters	10m (IW), 5m (SM)
Bands: list of S2 bands that are going to be used for computing the changes.	Any combination of S2 L2A bands	B2,B3,B4,B8
AoI: Area of Interest	WKT polygon or path to vector file	None
Projection: output projection	Any supported by SNAP	WGS84
NumberOfClasses: number of classes in which changes will be automatically classified.	Any integer	4
ReferenceVector: Reference vector to be used for computing the angle in CVA methodology.		1,0,0,…
Level of confidence:	Any float value lower than 100.	99,99
Output format: output format	Supported by GDAL and SNAP	GeoTiff

4.2.5.4 Processing

The standard approach when computing changes is simplified in Figure 6.

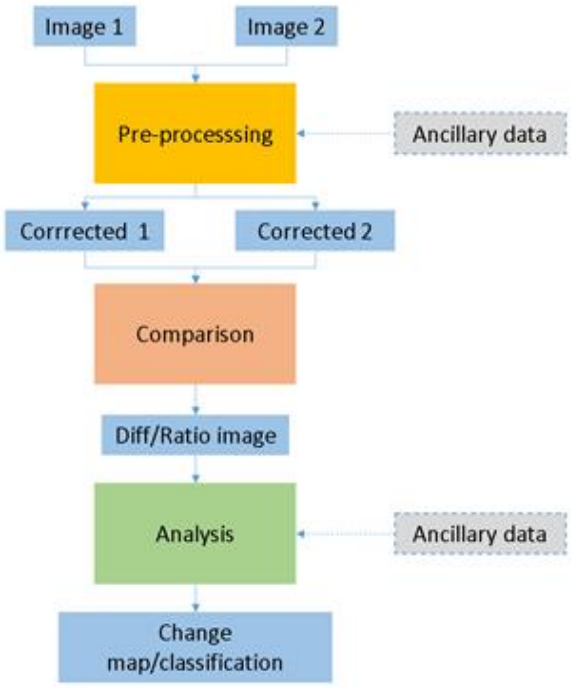

Figure 10. S2 Change detection processing flow.

The S2 change detection pipeline developed implements the following steps:

1. Pre-processing:
   . Resampling: Bands that are needed for the processing are resampled to the common selected resolution. These bands include the bands selected by the user and the scene classification bands. This step is performed using SNAP.
   a. Radiometric correction/histogram matching: in order to minimize errors caused by not accurate radiometric corrections (including atmospheric), a relative radiometric correction is applied to one of the inputs. For this, it is computed a linear regression using as references the pixels with less changes after removing the ones affected by clouds or where the land cover is different.
   b. Crop the image to the AoI.
   c. Generate cloud masks.
2. Computation of normalized vector of differences in pre-processed images
3. Compute the module of the vector and angle with respect to the reference vector.
4. Compute binary mask of changes by assuming that difference between bands follow gaussian distributions and the module of the change follows a chi-squared distribution:
   - After having applied the radiometric correction/histogram matching in 1.b, it is assumed that the difference of the same band in two images is following a Gaussian distribution. The values that cannot be explained with this distribution are classified as changes.

- When taking into account the full set of bands selected for the processing, the amplitude of the change is computed with the normalized differences and it is assumed that it follows a chi-squared distribution.

5. Classify changes using K-means algorithm in the polar representation of the vector of differences (amplitude and angle).

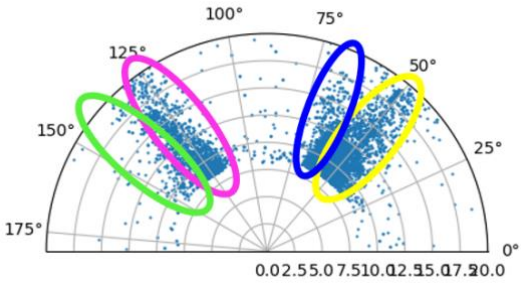

Figure 11. Example of classified changes in the polar representation using K-means.

4.2.5.5 Output

The outputs are:

- CVA: GeoTiff image with two bands. The first band is the amplitude of the change and the second band is the angle with respect to the reference vector.

- S2-CD: GeoTiff image with one band with pixel type Byte. It represents the classes of the detected changes.

4.2.5.6 How to use

Minimum requirements: 16GB of RAM.

Inside the docker, the pipeline can be found in `/app/pipelines` and can be executed with the following command:

```
S2-CD --input1 "VALUE" --input2 "VALUE" [--bands "XX,XX,XX"] [--AoI "WKT"] [--resolution "VALUE"] [--projection "VALUE"] [--numberClasses "VALUE"] [--referenceVector "VALUE"] [--levelConfidence "VALUE] [--output_format "VALUE"] --outdir "VALUE"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -v [local_path]:[container_path] DOCKER_IMAGE  S2-CD --input1 "VALUE" --input2 "VALUE" [--bands "XX,XX,XX"] [--AoI "WKT"] [--resolution "VALUE"] [--projection "VALUE"] [--numberClasses "VALUE"] [--referenceVector "VALUE"] [--levelConfidence "VALUE] [--output_format "VALUE"] --outdir "VALUE"
```

If any customization is needed in the processing graph, it can be found in the docker and could be adapted by the users and executed directly using *gpt*.

4.2.6 Vector data of human features

4.2.6.1 Summary

The vector data of human features service is based in OpenStreetMap (OSM). OSM is a collaborative project to create a free editable map of the world with an open-content license. The OSM License allows free (or almost free) access to the map images and all of the underlying map data. But this access is not always easy since there are limitations in the servers and APIs. Moreover, the data structure is not the preferred by Security domain.

SatCen has pre-processed and ingested in their own data model (SatCen Data Dictionary) the OSM data and can provide the data as a service in the scope of the project.

4.2.6.2 Input and parameters

The input data in which the service is based is the OSM data, but it is offered to the users in SatCen's data model.

For obtaining the data, the user has to indicate the features of interest (e.g. airport) and the area of interest.

4.2.6.3 Processing

The process can be considered a ECTL process (Extract, Cleanse, Transform and Load workflow). It was developed in FME and based in schema mapping documents.

4.2.6.4 Output

The outputs are the desired features in the AoI in geojson format.

4.2.6.5 How to use

This service is available as a web service where the users can select the desired feature and define the AoI.

5 Agriculture bootstrapping services and resources

5.1 Introduction

The Agriculture Bootstrapping service and resources have been developed to support the development of EO applications leveraging on AI algorithms for the food security and agriculture fields. The services aim to facilitate the integration of the proposals of the bidders in the AI4Copernicus environment.

To be effective at large scale, data and resources need to be properly pre-processed to accurately perform crop dynamic monitoring. The bootstrapping services described in this section allow the definition of a generic processing pipeline to perform crop type mapping, and to support the testing and implementation of innovative AI algorithms.

5.2 Services

The services described are available as a dockerized application that can be executed in any environment with a properly configured Docker client.

The Deep network for pixel-level classification of S2 patches service can be trained using the SEN12MS dataset, or with any user data that respects the Sentinel-2 data format. User can also adjust a various set of hyperparameters listed in the table below in order to optimize the results.

Concerning the TimeSen2Crop service, the architectures described in this section can be trained on training sets given by the users or using the TimeSen2Crop pixel-based dataset, described later in the Deliverable. The user has access to the model parameters to define the best combination of hyperparameters required for each particular application. The architectures can also be fine-tuned or be slightly modified to adapt to the use-case. However, to modify the scripts requires user technical knowledge on scripting languages (python).

The services have not been modified in the second update of the resources due to the absence of feedback and requests to change. Minor updates have been implemented to improve the QoL of the services.

5.2.1 Deep network for pixel-level classification of S2 patches

5.2.1.1 Summary

This service provides functionality for users to train a custom pixel-level classifier of Sentinel 2 patches. For example users can train a classifier for crop types (corn, sunflower, wheat, etc), land cover (urban vs. natural, water vs land), road extraction (road vs other).

The service will be available as a dockerized application that can be executed in any environment with a properly configured Docker client.

The service will implement code to train a U-NET model on Sentinel2 hyperspectral images (that can be first converted to smaller patches). The weights of a part of the U-NET model will be pre-trained with a self- supervised approach, increasing final classification performance.

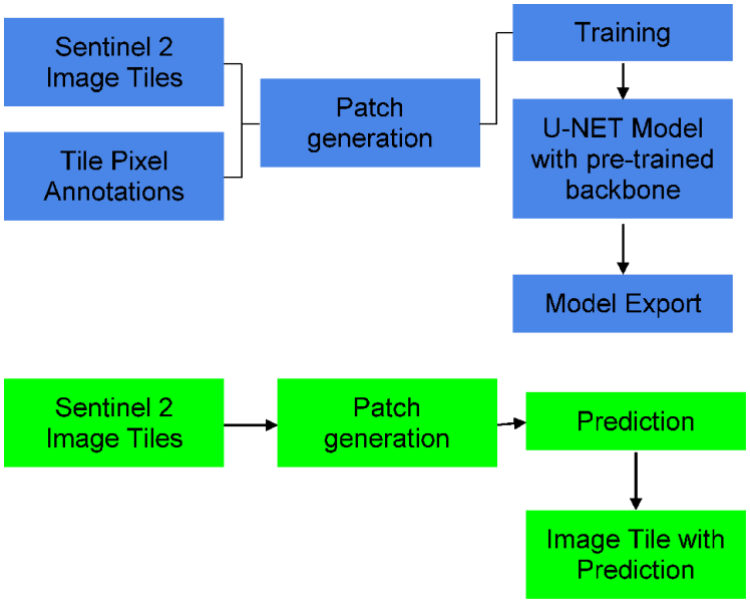

Figure 12. Deep network for pixel-level classification of S2 patches.

5.2.1.2 Input/output

The docker container will take as input a directory containing patches and their labels (patches with pixels = 1..N labels). The output will be a U-NET model which makes the classification by pixel with these N classes.

5.2.1.3 Pipeline

The service design is based on a network architecture with a backbone, a ResNet-50, which is pre-trained separately on SEN12MS. By using self supervised learning, an emerging unsupervised training procedure, we will learn good features on Sentinel-2 images, without requiring labelling, to initialize our network's backbone (*Ref. Citation: Ciocarlan, A.; Stoian, A. Ship Detection in Sentinel 2 Multi-Spectral Images with Self-Supervised Learning. Remote Sens. 2021, 13, 4255. https://doi.org/10.3390/rs13214255)*

Users provide their own data under the Sentinel format and can train a segmentation model. A various number of parameters are available to customise the training phase (see table below). The service also provides the code to apply the trained model to a new set of data.

5.2.1.4 Exposed parameters

Table 8. Exposed parameters in Deep network for pixel-level classification of S2 patches.

Parameter	Valid values	Default Value
-action: action to perform. One among "full", "extract_data", train","predict"'	full, extract_data, train, predict	_

-criterion: loss criterion used during training	cross_entropy, focal_loss	cross_entropy
-pretrained: enable pretrained backbone	True/False	False
-bands: comma separated list of bands:	Any combination of S2 L2A bands: B1,B2, B3, B4, B5, B6, B7, B8, B8A, B9, B10, B11, B12	_
-selected-classes: comma separated id of classes.	1-255	1
-data_aug: enable data augmentation	True/False	False
-simplified: enable simplified label representation for the Sen12MS dataset	True/False	False
-b, --batch-size: batch size for training	2-64	2
-epochs: number of epochs for training	1-99	20
-patch_size: size of input data for training	256/64	256
-lr: initial learning rate	Any non-zero positive real value	0,0001
-model_format: model save format	Pytorch, onnx	Pytorch
-model_file: model file name	String value	_

5.2.1.5 How to use

Recommended requirements : 16GB of RAM, RTX 2080ti.

First of all the user should obtain a data set of S2 image tiles with any combination of bands. Furthermore, the user should then provide pixel level annotation of these image tiles. These tiles can be of any size, with the lowest size being 64x64 pixels. Larger image tiles will be cut into 64x64 patches by the service, automatically.

The annotations do not need to cover all image pixels, some pixels that have unknown/irrelevant types will have class 0. This can be considered a background class. For example if the user wants to segment roads, she/he will annotate all road pixels with class 1 and leave the rest of the pixels as class 0.

To train a model the user will invoke the dockerized applications through the command line or eventually through the cloud interface. He/she will point the application to a directory containing Sentinel 2 images and their annotations. After training, the application will produce a "trained model" that can produce predictions on new image data. Calling the application in "Predict" mode while supplying such a saved trained model will allow the user to predict pixel level classification on new data.

5.2.2 TimeSen2Crop

5.2.2.1 Summary

TimeSen2Crop is a pixel-based dataset made up of more than 1 million crop type samples of Sentinel-2 time series. The dataset includes atmospherically corrected images and reports the snow, shadows, and clouds information per labelled unit, as well as the spectral signature of the samples of nine Sentinel-2 spectral bands at 10m of spatial resolution.

5.2.2.2 Dataset Description

The dataset is organized hierarchically, as shown in Figure 8. The data are organized per Sentinel-2 tiles, i.e. 16 folders. Each folder contains 16 sub-folders, each associated to a particular crop type. The multispectral temporal signature is stored in a .csv file, which provides a matrix where each row indicates the acquisition date, and each column indicates the spectral band associated. The last column in the file shows the condition of the pixel. A csv containing the list of the dates in which the samples have been acquired is stored inside the tile folder.

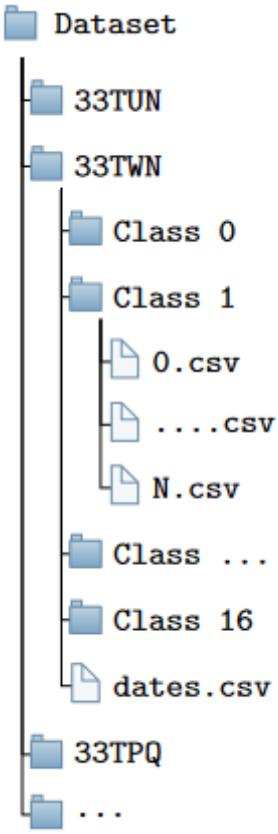

Figure 13. Hierarchical folder structure of TimeSen2Crop.

Table 9. Dataset specifics for TimeSen2Crop.

Dataset Specifics	Values
Spatial Resolution	10 [m]
Spectral Resolution	B2-490 [nm], B3-560 [nm], B4-665 [nm], B5-705 [nm], B6-740 [nm], B7-783 [nm], B8A-865 [nm], B11-1610 [nm], B12-2190 [nm]
Pixel condition	clear [0], cloud [1], shadow [2], snow [3]
Agronomic Year	September 2017 - August 2018
Crop Types – Classification Scheme	Legumes [0], Grassland [1], Maize [2], Potato [3], Sunflower [4], Soy [5], Winter Barley [6], Winter Caraway [7], Rye [8], Rapeseed [9], Beet [10], Spring Cereals [11], Winter Wheat [12], Winter Triticale [13], Permanent Plantations [14], Other Crops [15]

Sentinel-2 Tiles	32TNT, 32TPT, 32TQT, 33TUM, 33TUN, 33TVM, 33TVN, 33TWM, 33TWN, 33TXN, 33UUP, 33UVP, 33UWP, 33UWQ, 33UXP

5.2.3 Harmonization of pre-processed Time Series of Sentinel-2 data

5.2.3.1 Summary

One of the challenges of crop type mapping at large scale is the definition of a regular temporal sampling grid to perform analysis in a standardized manner. This optical pre-processing harmonizes the irregular time series of images and mitigates the cloud coverage problem. The harmonization of pre-processed time series of Sentinel-2 data considers a statistic-based approach that computes the median for each pixel in the different images acquired in a particular month. The pixel composite approach to mosaic generation provides consistent results at large scale, allowing the processing of harmonized acquisitions.

5.2.3.2 Input

The inputs of this service are a Sentinel-2 Bottom of Atmosphere time series and the LAI images associated. The input to this service can be obtained using the pre-processing pipeline defined in the previous section. The `input_dir` specified to execute the service must contain two folders, "REFBOA" and "LAI", each containing the .tif images.

5.2.3.3 Exposed Parameters

Table 10. Exposed parameters in Harmonization of pre-processed Time Series of Sentinel-2 data.

Parameter	Valid Values	Default Value
tile: UTM code associated with the selected tile (i.e. "33UVP")	string	None
season_start: Year at which the agronomic year starts defined from September (A) to August (B).	number	2017

5.2.3.4 Processing

The integrated algorithm has been designed using python. The script reads the LAIs of the images in a particular month, sets to 0 the corresponding pixels in the BOAs, and performs the median operation masking the cloudy samples.

5.2.3.5 Output

The outputs of the Harmonization step are 12 monthly composite GeoTiff images in uint16 representing the agronomic year specified by the user.

5.2.3.6 How to use

Minimum Requirements: 16GB RAM, GPU: None

Inside the docker, the pipeline can be executed with the following command:

```
python3 AI4C_MComp.py --input_dir="VALUE" --output_dir="VALUE" --season_start="VALUE" --tile="VALUE"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -u $(id -u):$(id -g) -v $(pwd):$(pwd) -w $(pwd) docker_name python3 AI4C_MComp.py --input_dir="VALUE" --output_dir="VALUE --season_start="VALUE" --tile="VALUE"
```

The user can check the help page to have an overview of the variables that can be defined and a short description of each parameter.

```
monthlycomposite --help
```

5.2.4 Long Short-Term Memory Neural Network for NDVI prediction

5.2.4.1 Summary

The Long Short-Term Memory neural network for NDVI prediction performs the training of the architecture using the data provided by the user. The service exploits the data to train from scratch an LSTM for the prediction of NDVI values (or other indices) in the time series and stores the resulting weights and the trained model. Several parameters are exposed to allow the customization of the model.

5.2.4.2 Input

The input data must be stored inside a directory that contains three subdirectories: train, test, and val. You can see an example of input data inside the folder "example" within the docker. The input data must be in .npy format or .csv format. The input data consists in an array of dimension [TimeSteps, nSamples], where TimeSteps is the number of time steps on which the train model is trained, and nSamples the number of training samples.

The data in the input folder must be called "X.csv" or "X.npy" for the reader to automatically load the input samples. The targets of the LSTM must be stored in the same folder of the input samples and be called "y.csv" or "y.csv". In the case the secondary file called "y.csv" or "y.csv" is missing, the architecture will consider the last row of "X.csv" or "X.npy" as the targets of the prediction of the LSTM. If "y.csv" or "y.npy" are stored together with "X.csv" or "X.npy", the script will use the entire time series to predict the targets in "y.csv" or "y.npy".

5.2.4.3 Exposed Parameters

Table 11. Exposed parameters in Long Short-Term Memory Neural Network for Sentinel-2 for prediction.

Parameter	Valid Values	Default Value
input_dir: Path where the .npy or the .csv are stored. The input dir must contain three subfolders: train, val, test. Subfolders can be empty, with the exception of the train subfolder.	string	/input/

output_dir: Path where the model will be stored after training. If a test is specified, the predictions will be stored in this folder.	string	/output/
epochs: Number of epochs to train the model.	integer	50
batch_size: Batch size for the training.	integer	64
lr: number describing the step size.	float	1e-3

5.2.4.4 Processing

The service has been designed using python. The script trains the LSTM using the samples provided by the user and performs the prediction if a test time series is provided.

5.2.4.5 Output

The output of the Long Short-Term Memory network is a .h5 file containing the weights and the model configuration. If the user provided a test .npy or .csv file, the service stores a prediction.csv file containing the predicted value after the LSTM has been trained.

5.2.4.6 How to use

Minimum Requirements: 1 GPU x worker, RAM depending on the size of the user dataset.

Inside the docker, the pipeline can be executed with the following command:

```
python3 Distributed_main.py --input_dir="VALUE" --output_dir="VALUE" --epochs="VALUE" --batch_size="VALUE" --lr="VALUE"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -u $(id -u):$(id -g) -v $(pwd):$(pwd) -w $(pwd) docker_name python3 Distributed_main.py --input_dir="VALUE" --output_dir="VALUE" --epochs="VALUE" --batch_size="VALUE" --lr="VALUE"
```

The user can check the help page to have an overview of the variables that can be defined and a short description of each parameter.

```
distributed_main --help
```

The docker also contains an example of training using a few test samples. The example can be run using the following:

```
python3 Distributed_main.py --input_dir=example --output_dir=output --epochs=50 --batch_size=64 --lr=0.0001
```

5.2.5 Long Short-Term Memory Neural Network for Sentinel-2 for crop type classification

5.2.5.1 Summary

The Long Short Term-Memory neural network for crop type classification can be trained using samples selected by the user. The service exploits the data given by the user to train from scratch an LSTM and stores the resulting weights and model. Several parameters are exposed to allow the customization of the model.

5.2.5.2 Input

The service requires two different .npy files stored inside the `input_dir` specified, one containing the multitemporal spectral signature of the samples used to train the LSTM ("train.npy"), and a .npy file containing the labels associated with the samples ("y_train.npy"). The dimension of the array containing the samples used to train the model must be [N,ts,b], where N is the number of samples, ts the number of time steps in the Time Series, and b the number of bands.

5.2.5.3 Exposed Parameters

Table 12. Exposed parameters in Long Short-Term Memory Neural Network for Sentinel-2.

Parameter	Valid Values	Default Value
n_epochs: number of epochs to update the internal model parameters	Any number >1	50
batch_size: number of samples used to update the internal models	Any number >2 and power of 2	32
class_weights: flag used in the training to give different weights based on the a-priori probability of each class.	0 (not used) - 1 (used)	0
learning_rate: number describing the step size.	Any number > 0 and < 1	1e-3
val: string pointing to the path where a validation set is stored.	path to folder containing the .npy data	None

5.2.5.4 Processing

The service has been designed using python. The script performs a normalization on the input data and trains the neural network.

5.2.5.5 Output

The output of the Long Short-Term Memory network is a .h5 file containing the weights and the model configuration.

5.2.5.6 How to use

Minimum Requirements: 16GB RAM, 1 GPU x worker

Inside the docker, the pipeline can be executed with the following command:

```
python3 AI4C_LSTMTrain.py --input_dir="VALUE" --output_dir="VALUE" --
n_epochs="VALUE" --batch_size="VALUE" --class_weights="VALUE" --
learning_rate="VALUE" --val="VALUE"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -u $(id -u):$(id -g) -v $(pwd):$(pwd) -w $(pwd) docker_name python3
AI4C_LSTMTrain.py --input_dir="VALUE" --output_dir="VALUE" --n_epochs="VALUE" --
batch_size="VALUE" --class_weights="VALUE" --learning_rate="VALUE" --val="VALUE"
```

The user can check the help page to have an overview of the variables that can be defined and a short description of each parameter.

```
train_lstm --help
```

5.2.6 Pre-Trained Long Short-Term Memory for crop type classification

5.2.6.1 Summary

The pre-trained Long Short Term-Memory architecture is already trained using the TimeSen2Crop database and is available in .h5 format. The service exploits a pre-trained architecture to classify the specified tile harmonized using the monthly composite approach.

5.2.6.2 Input

The input of this service is a GeoTiff image containing the information regarding the agricultural samples that the LSTM must classify (if no mask is provided, the service will classify the entire image), and a monthly composite time series. The .tif image and the .h5 pre-trained network must be in the --input_dir folder.

5.2.6.3 Processing

The service has been designed using python. The monthly composite time series is classified according to the TimeSen2Crop classification scheme. The GeoTiff image defined by the user is used to mask the samples not related to agriculture and the samples where classification is not needed. If the binary crop mask is specified, pixels with 0 value will be classified, while pixels with other values will be ignored. The script performs a block-wise classification of the monthly composites and applies morphological operators to the result.

5.2.6.4 Output

The output is the crop type map associated to the monthly composite time series as a GeoTiff image in uint8.

5.2.6.5 How to use

Inside the docker, the pipeline can be executed with the following command:

```
python3 AI4C_LSTMInference.py --input_dir="VALUE" --output_dir="VALUE" --
mc_dir="VALUE"
```

It can be also executed with "*docker run*" taking into account that a volume has to be mounted in order to write on it the output file so it is accessible at the end of the processing.

```
docker run -u $(id -u):$(id -g) -v $(pwd):$(pwd) -w $(pwd) docker_name python3
AI4C_LSTMInference.py --input_dir="VALUE" --output_dir="VALUE" --mc_dir="VALUE"
```

The user can check the help page to have an overview of the variables that can be defined and a short description of each parameter.

```
tile_classifier --help
```

6 Energy bootstrapping services and resources

6.1 Introduction

The AI4Copernicus Energy Bootstrapping services have been identified to support the application of AI technology towards satellite data to produce models or forecasts that address one of the three broad energy questions.

The three broad questions are:

- As a user I want to know where I can and cannot build low carbon and renewable energy infrastructure
- As a user I want a better understanding of energy consumption and energy needs of a society
- As a user I want to know where to carry out precision, pre-emptive, maintenance in my energy infrastructure

The main function of the services are labelled data sets used to train machine learning and artificial intelligence services. In addition to these domain specific resources, the energy domain will also exploit the bootstrapping services from other domains (e.g. pre-processing services of Sentinel-1 and -2 from the security) if requested by open calls winners. In order not to duplicate the information, the details of other generic services can be found in the other domains' sections.

6.2 Resources

Table 13. Labelled datasets for Energy domain.

Meteorological data: ERA5	CREODIAS - ERA5 product home page How to order CDS ERA5 products - FAQ Answer - CREODIAS
Example: offshore wind farms are located (training data)	EMODNET Data Catalogue EMODnet Product Catalogue - EMODnet (europa.eu)

JRC Open Power Plants Database (JRC-PPDB-OPEN)	WMS service identified (https://data.europa.eu/data/datasets/9810feeb-f062-49cd-8e76-8d8cfd488a05?locale=en)
Open data from the floating offshore wind farm, Hywind Scotland	Platform for Operational Data (POD) | POD (catapult.org.uk)

7 Health bootstrapping services and resources

7.1 Introduction

The AI4Copernicus Health Bootstrapping services and resources are being developed to address current public health and air pollution (or air quality) challenges using Earth observation and in-situ measurement data. The services are focused on probabilistic downscaling (super-resolution) of air quality (AQ) and atmospheric composition (AC) model output. Current AC and AQ models output forecasts at relatively low-resolution (e.g., ca. 10 to 40 km for the CAMS EU and global domains). Previous research has demonstrated that it is possible to make use of Earth observation (EO) and in-situ measurement data to downscale (i.e., increase the spatio-temporal resolution) of dynamical model output, allowing the identification of pollution or greenhouse gas (GHG) emission hotspots. This service is well aligned with all proposals submitted to the AI4Copernicus first open call under the health domain. These proposals have identified the need and market potential for hyper-local short to medium-term air pollution forecasts, with up to street-scale resolution in densely populated areas that are subject to heavy pollution episodes. Real-time knowledge of AQ plus high-resolution forecasts up to a few days ahead allow policymakers to make science-informed decisions (e.g. limit traffic, issue health advisories, deploy additional health personnel, etc.). Citizens can also benefit by reducing their exposure to air pollutants - staying indoors during aerosol peaks, for example, or using AQ-aware routing when navigating through traffic-congested cities.

The health services also provide an estimate of the systematic uncertainties present in the downscaled product.

It is important to note that the technical specification of the service described below is subject to changes. The main objectives are to (1) provide a health bootstrap service that is useful for the winning bidders of the first open call and (2) ensure interoperability with other services described in this document if advantageous for end users.

7.2 Services

7.2.1 Probabilistic downscaling of CAMS air quality model data

This service generates high-resolution (currently ~ 10km) air quality maps from low-resolution (~40 - 80km) CAMS model (re)analysis and/or forecast output.

7.2.1.1 Input data

The input data sources for the probabilistic downscaling service are summarized in the table below.

Table 14. Input data for Probabilistic downscaling of CAMS air quality model data.

Dataset	Summary	Format(s)
CAMS European air quality analysis and forecasts	This dataset provides daily air quality analyses and forecasts for Europe (east boundary=25.0° W, west=45.0° E, south=30.0° N, north=72.0° N) at a spatial resolution of 0.1 degrees (ca. 10km). The set of atmospheric variables includes particulate matter, pollen, ozone, dust, CO, NO, SO2, NO2 and ammonia. The downscaling service currently uses only surface data. Other vertical levels may be incorporated depending on user need.	netCDF
CAMS global (re-)analysis	The EAC4 reanalysis. The downscaling service currently uses only surface data. Other vertical levels may be incorporated depending on user need and/or impact on downscaled field quality.	netCDF
Sentinel-2	The use of Level-2A Sentinel-2 image data for identification of pollution hotspots is currently being investigated. Satellite-retrieved aerosol optical depth (AOD)[11] can be correlated to aerosol conditions over urban environments at a fine scale. Furthermore, it has been demonstrated that data from the Sentinel-2 MultiSpectral Instrument can be used to detect and quantify anomalously large methane point sources.[12]	GeoTIFF (can be the output of the Sentinel-2 pipeline described for the "Security" domain above)
Sentinel-5P[13]	The Copernicus Sentinel-5 Precursor mission provides atmospheric measurements of air quality and atmospheric composition (aerosol, ozone, NO2, CH4, CO, HCHO and SO2) with high temporal and spatial resolution (7 x 7 km at nadir	xml (header) and netCDF (data)

[11] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-3-slstr/product-types/level-2-aod (product under development)

[12] https://amt.copernicus.org/articles/14/2771/2021/ and https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-5P/Mapping_high-resolution_methane_emissions_from_space

[13] https://sentinel.esa.int/web/sentinel/sentinel-data-access

	resolution). Sentinel-5P NO2 total columns are assimilated operationally into the CAMS-IFS 4D-Var model. The data are available in near-real-time (within three hours of sensing, except for total ozone which has a 24hrs delay).	
CDS weather data	The Climate Data Store (CDS, http://cds.climate.copernicus.eu) is the operational data access portal of the Copernicus Climate Change Service (C3S), which is implemented by ECMWF on behalf of the European Commission. It provides access to the C3S portfolio of products through a web interface and API. The CDS includes high-resolution weather and climate reanalysis datasets such as the ERA5 and UERRA (EU-regional) multi-decadal re-analyses. Air quality and atmospheric composition are affected by weather, hence the addition of weather information often leads to an improvement in the accuracy of the downscaled data product.	netCDF, GRIB
In-situ data: e.g. low-cost sensors, IoT devices, weather stations, ...	These data will be made available and incorporated into the service at a later date, subject to licensing agreements.	XML, JSON, CSV, ...
Population density and built area maps	Regridded to 0.1 degrees. Original data available at 1km global resolution.	netCDF
Orography (surface elevation)	Regridded to 0.1 degrees. High-resolution data available e.g. from EU-DEM[14] for the European domain (25m pixels)	netCDF, Geotiff

7.2.1.2 Algorithm

The probabilistic downscaling engine is based on a generative adversarial network (GAN). Recent GAN-based models have achieved impressive performance in artificial high-resolution "image" generation for the Earth sciences[15]. Crucially, GANs allow the generation of "image" (i.e., 2D dataset) ensembles that quantify the uncertainty in the high-resolution output fields.

7.2.1.3 Output Data

Table 15. Output data for Probabilistic downscaling of CAMS air quality model data.

High-resolution air quality data	10km (or finer) spatial resolution for the initial deployment of the service on WEkEO and CREODIAS. This is envisaged to go up to	netCDF, CSV (for point locations of interest)

[14] https://land.copernicus.eu/user-corner/publications/eu-dem-flyer/view

[15] e.g., Ledig et al., 2016: https://arxiv.org/abs/1609.04802 ; Stengel et al., 2020: https://www.pnas.org/content/117/29/16805 ; Leinonen et al., 2020: https://arxiv.org/abs/2005.10374 ;

	street-level after hyper-local data sources are incorporated into the machine learning algorithms. PM2.5 is currently supported. Other variables to be added in subsequent iterations.	
High-resolution atmospheric composition data	10km (or finer resolution). NO2	netCDF, CSV

7.2.1.4 How to use

The downscaling pipeline is available as a dockerized application that can be executed in any environment with a properly configured Docker client. Inside the docker, the pipeline can be executed with the following commands:

Table 16. Commands to execute Probabilistic downscaling of CAMS air quality model data.

Pre-process input data	`aqgan-preproc --start-date <YYYYMMDDHH> --end-date <YYYYMMDDHH> --dataset <DATASET> --config config.yaml`	The YAML configuration file contains important model settings, e.g. the size of a data "tile" (hi-res: 64 x 64, low-res: 8 x 8). An example config file is provided in the source tree under aqgan/src/config
Train the GAN model on a predefined data interval	`aqgan-train --model srgan --start-date <YYYYMMDDHH> --end-date <YYYYMMDDHH> --config config.yaml`	Start and end dates for training must be provided. Currently only the srgan (super-resolution GAN) model is supported.
Generate new high-resolution frames with a pre-trained SR-GAN model	`aqgan-generate --model srgan --start-date <YYYYMMDDHH> --end-date <YYYYMMDDHH> --config config.yaml`	

7.2.1.5 Compute requirements

It is recommended to run the downscaling pipeline on an NVIDIA GPU (CUDA 10.2 or newer) with a minimum of 16 GB RAM. The minimum CPU memory required is 32GB, but this number can vary according to the size of the input dataset (we recommend 128GB or more for maximum performance). Should the total size of the input dataset exceed the amount of available CPU RAM, input data can be loaded lazily using a library such as Dask[16].

7.2.1.6 Exposed parameters

The user can change the model settings through the YAML configuration file. This way the model architecture can be adapted to the task at hand, new input data sources can be ingested, etc.

Table 17. Exposed parameters in Probabilistic downscaling of CAMS air quality model data.

Parameter	Default Value
Input and output frame size	8 x 8 and 64 x 64
Type of super-resolution model	srgan[17]
Area of Interest (AoI)	lat-lon bounding box
Output path	netCDF
Number of filters in the generator and discriminator	64 and 32
Learning rate	10^-5
Batch size	128
Type of adversarial loss	Wasserstein[18]
Low-resolution inputs	"pm2p5", "u10", "v10", "t2m", "lsm", "sp", "z", "pm10"
High-resolution (static) inputs	"urban_frac" (urban fraction), "orog_scal" (min-max scaled orography)
High-resolution output(s)	“pm2p5” (PM2.5)

[16] https://www.dask.org/

[17] inspired by https://arxiv.org/abs/1609.04802

[18] Arjovsky et al., 2017: https://arxiv.org/abs/1701.07875

Input, output data paths	See the config file.
Output “ensemble” size	10

- Appendix: docker registry access

A Docker registry is a storage and distribution system for Docker images. It is organised in Docker repositories that contain all the versions published of a specific image. It allows the developers/providers to tag and push their images that can be pulled by the users to run them.

CloudFerro has deployed an instance of Harbor (goharbor.io), which is an open source registry that can be accessed in https://harborai4c.cloudferro.com/ .

Different users have been created for the service providers (with 'Developer' role) and another user for the funded projects with 'Guest' role that allows them to pull the images.

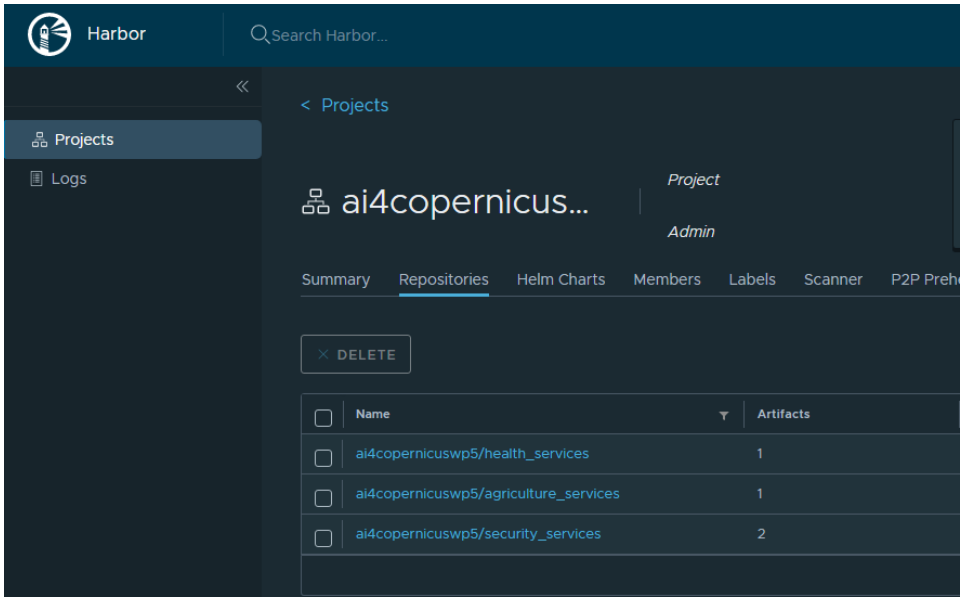

Figure 14. Docker registry screenshot.

The typical steps for pulling and running the services are:

- Login to registry

```
docker login -u=[YOUR_USER] -p=[PASSWORD] harborai4c.cloudferro.com
```

- Pull images (example with security services image)

```
docker pull harborai4c.cloudferro.com/ai4copernicuswp5/security_services:1.0.1
```

- Run a container

```
docker run -it harborai4c.cloudferro.com/ai4copernicuswp5/security_services:1.0.1 bash
```

- Run a container with a volume (local folder mounted in container)

```
docker run -it -v /tmp/example_products/:/output harborai4c.cloudferro.com/ai4copernicuswp5/security_services:1.0.1 bash
```

where /tmp/example_products is a local (Docker host) folder and /output is the folder in the container

- Copy files from/to the container

from Container to Docker Host

```
docker cp {options} CONTAINER:SRC_PATH DEST_PATH
```

from Docker Host to Container

```
docker cp {options} SRC_PATH CONTAINER:DEST_PATH
```

where the container can be obtained from `docker ps`