Sentinel-1 GRD pre-processing
Technical Documentation

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

Table of Contents

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.

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 following section describe the Sentinel-1 GRD pre-processing service, developed by SatCen in the frame of the Security domain.

2   Sentinel-1 GRD pre-processing

2.1   Summary

The Sentinel-1 GRD pre-processing pipeline is available as a dockerized application (see annex) 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.

2.2   Input

The input of this pipeline is a Level-1 Ground Range Detected (GRD) product in its native SENTINEL-SAFE [1] 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 approximately fixed off-nadir angle (this is subject to small variations because of roll steering).*

---

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

SM images have continuous along track image quality at an approximately constant incidence angle.[2]

- 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[3].*

2.3   Exposed Parameters

Table 1. 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

---

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

2.4   Processing

The pipeline has been designed using SNAP[4], 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…).

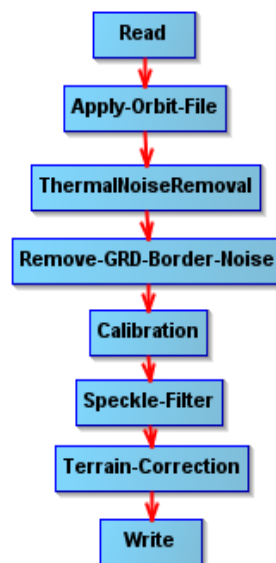In the case of this pipeline, the SNAP graph executed is represented in Figure 1.

Figure 1. 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.)

---

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

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).

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-GRD-preprocess --input "VALUE" [--calibration "VALUE"] [--polarization "VALUE"]
[--speckle "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--landseamask "VALUE"]
[--projection "VALUE"] [--output_format "VALUE"] --outdir "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).

2.6.1   Some examples

- S1 image over port of Valencia (Spain) at 10 meters resolution. It has been applied a sea mask.

```
S1-GRD-preprocess                                                      --input
S1A_IW_GRDH_1SDV_20231112T061020_20231112T061045_051180_062C59_75CD.SAFE     --
polarization VV --AoI "POLYGON((-0.37 39.50, -0.26 39.50, -0.26 39.38, -0.37 39.38,
-0.37 39.50))" --resolution 10 --landseamask Sea  --outdir /products/
```
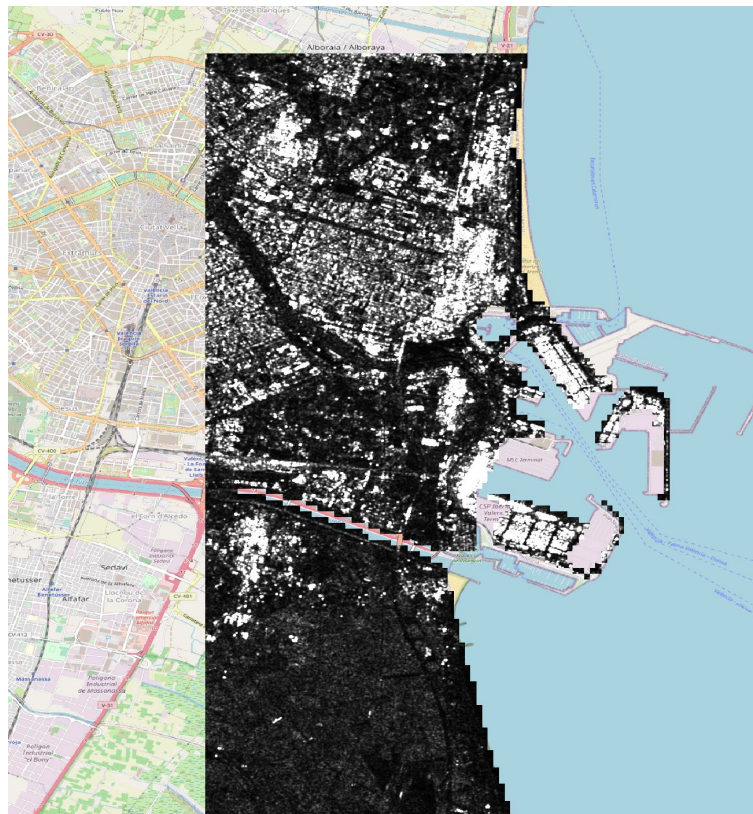
- S1 image over port of Valencia (Spain) at 10 meters resolution. It has been applied a land mask and its generated the polarization VH.

```
S1-GRD-preprocess                                                    --input
S1A_IW_GRDH_1SDV_20231112T061020_20231112T061045_051180_062C59_75CD.SAFE  --
polarization VH --AoI "POLYGON((-0.37 39.50, -0.26 39.50, -0.26 39.38, -0.37
39.38, -0.37 39.50))" --resolution 10 --landseamask Land  --outdir /products/
```
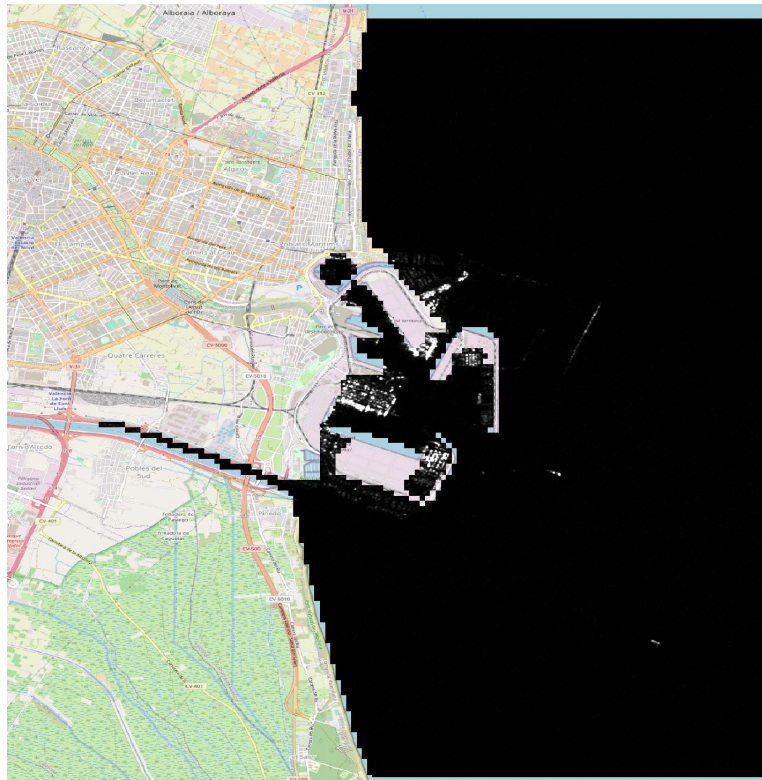
Appendix: docker registry access

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.
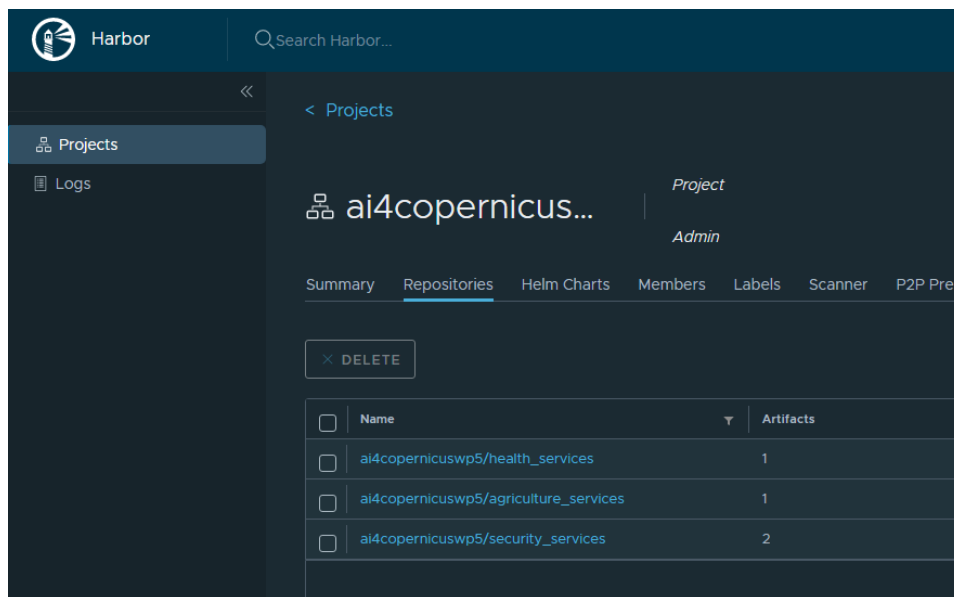

Figure 2. Docker registry screenshot.

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