AI4 copernicus

Sentinel-1 change detection

Technical Documentation

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

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 101016798.

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 change detection service, developed by SatCen in the frame of the Security domain.

2 Sentinel-1 change detection

2.1 Summary

The Sentinel-1 Change Detection resource is available as a dockerized application (see annex) 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.

2.2 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.

2.3 Exposed Parameters

Table 1. Exposed parameters in Sentinel-1 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)
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

2.4 Processing

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

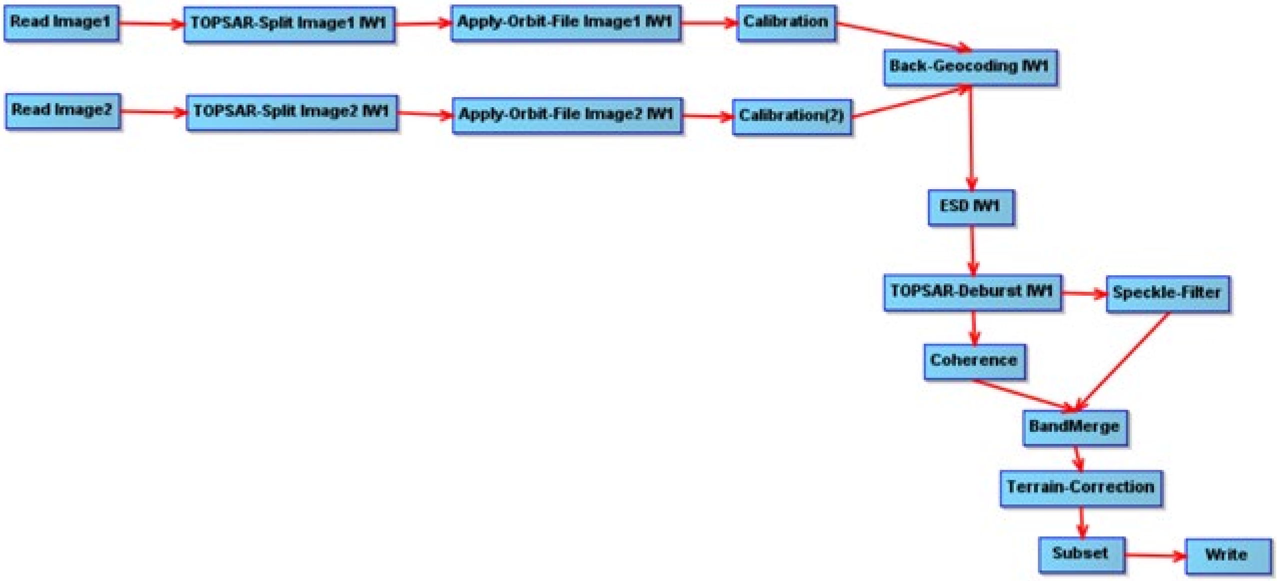


Figure 1. Sentinel-1 IW processing.

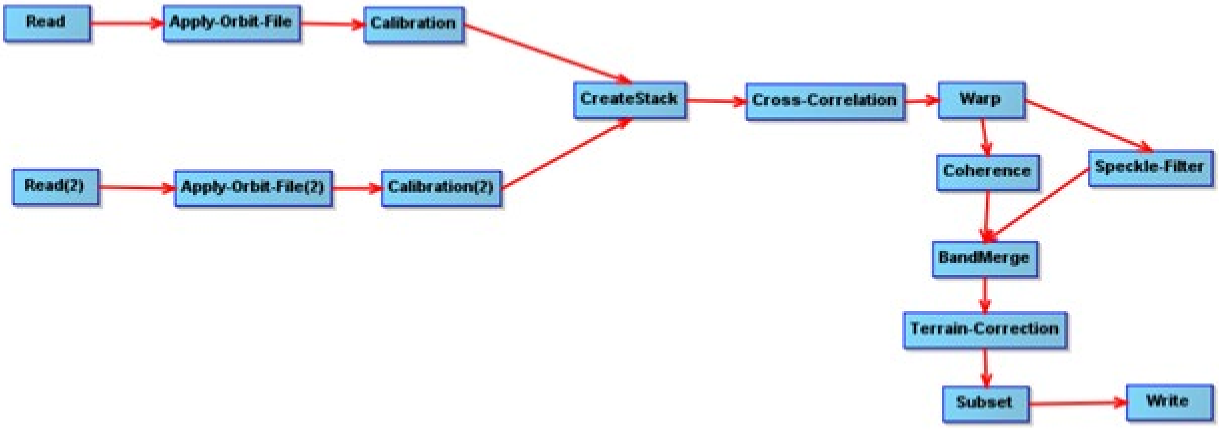


Figure 2. Sentinel-1 SM processing.

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.)

2.5 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)

2.6 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”
```

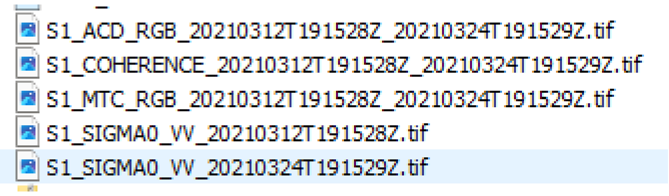
If any customization is needed in the processing graphs, they can be found in the docker and could be adapted by the users and executed directly using *gpt*.

2.6.1 Some examples

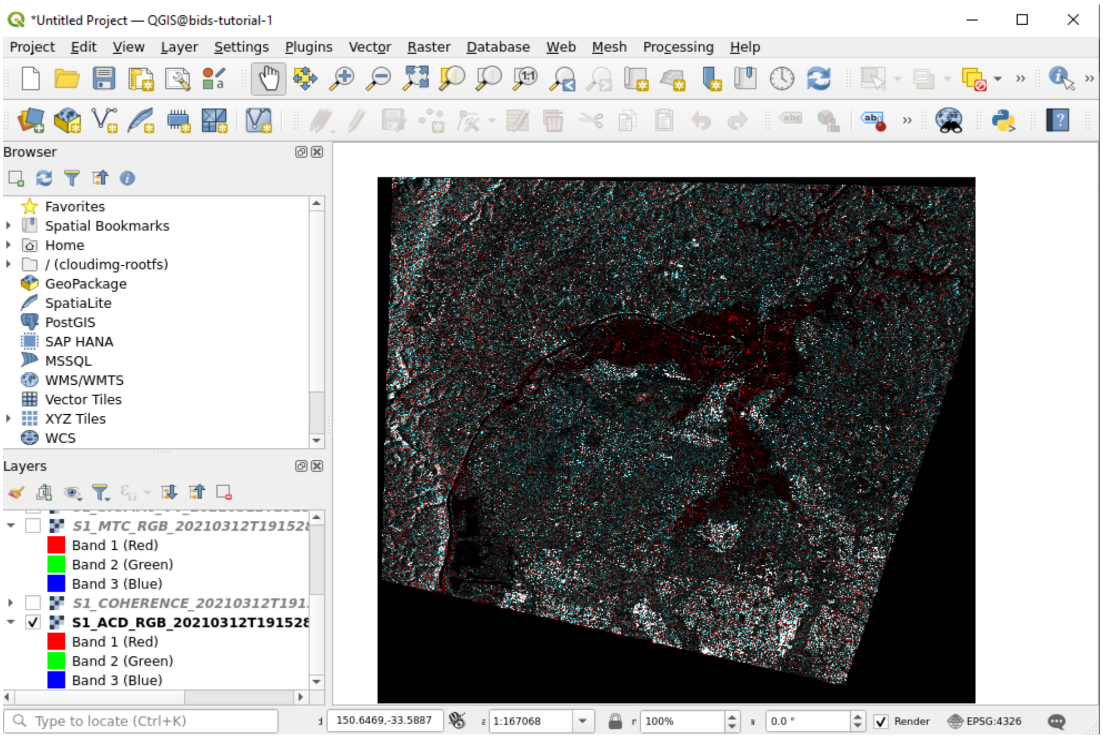
Computation of change detection in an area close to Sydney affected by a flooding in March 2021. They are used two inputs acquired respectively before and after the event:

```
S1-CD -i1
/output/S1A_IW_SLC__1SDV_20210312T191528_20210312T191555_036969_045983_7048.zip -
i2
/output/S1A_IW_SLC__1SDV_20210324T191529_20210324T191555_037144_045F93_9AB1.zip -
r 20 -p "POLYGON((150.6266 -33.49, 150.952 -33.49, 150.952 -33.795, 150.6266 -
33.795, 150.6266 -33.49))" -outdir /output/
```

The command generates the following products:



S1_ACD_RGB_20210312T191528Z_20210324T191529Z.tif
S1_COHERENCE_20210312T191528Z_20210324T191529Z.tif
S1_MTC_RGB_20210312T191528Z_20210324T191529Z.tif
S1_SIGMA0_VV_20210312T191528Z.tif
S1_SIGMA0_VV_20210324T191529Z.tif



*Untitled Project — QGIS@bids-tutorial-1

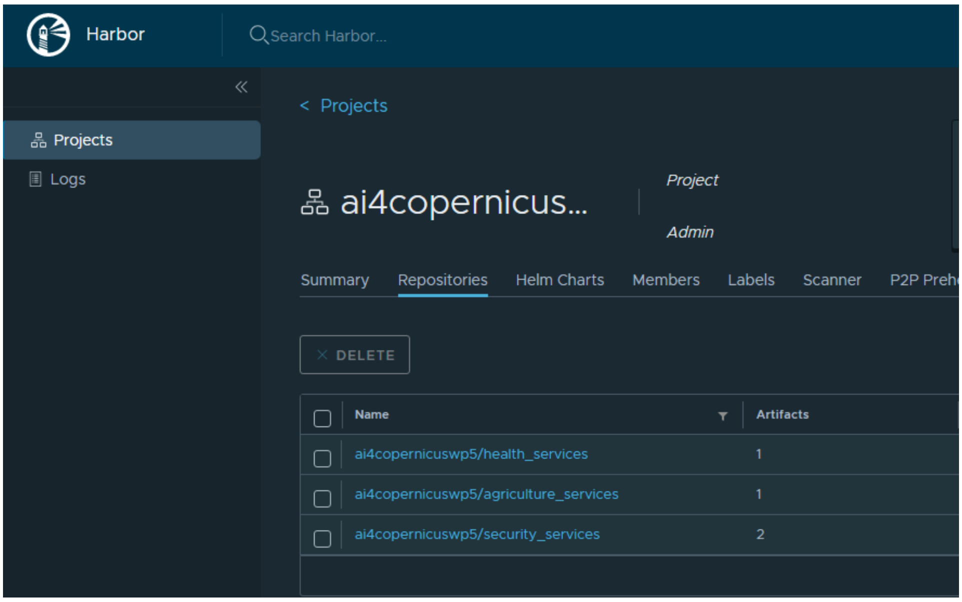
Layers:
- ☐ S1_MTC_RGB_20210312T19152…
  - 🟥 Band 1 (Red)
  - 🟩 Band 2 (Green)
  - 🟦 Band 3 (Blue)
- ☐ S1_COHERENCE_20210312T191…
- ☑ S1_ACD_RGB_20210312T19152…
  - 🟥 Band 1 (Red)
  - 🟩 Band 2 (Green)
  - 🟦 Band 3 (Blue)

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.



□	Name	Artifacts
□	ai4copernicuswp5/health_services	1
□	ai4copernicuswp5/agriculture_services	1
□	ai4copernicuswp5/security_services	2

Figure 3. 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

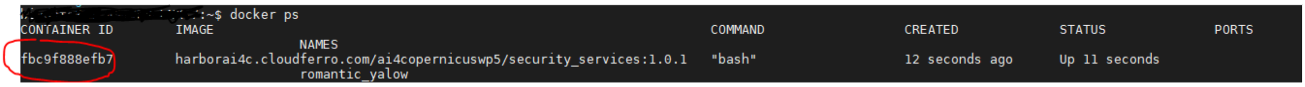
# from Container to Docker Host

```
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



```
[redacted]:~$ docker ps
CONTAINER ID        IMAGE                                                          COMMAND             CREATED             STATUS              PORTS
                NAMES
fbc9f888efb7        harborai4c.cloudferro.com/ai4copernicuswp5/security_services:1.0.1   "bash"              12 seconds ago      Up 11 seconds
                romantic_yalow
```