AI4 copernicus

Sentinel-2 pre-processing

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-2 pre-processing service, developed by SatCen in the frame of the Security domain.

2 Sentinel-2 pre-processing

2.1 Summary

The Sentinel-2 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 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.

2.2 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.

2.3 Exposed Parameters

Table 1. 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

2.4 Processing

The main workflow has been designed using SNAP. The SNAP graph executed is represented in figure below.

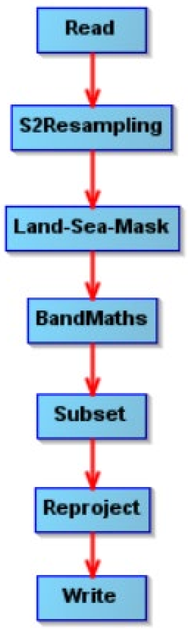


Figure 1. S2 pre-processing graph.

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.)

2.5 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.

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:

```
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.

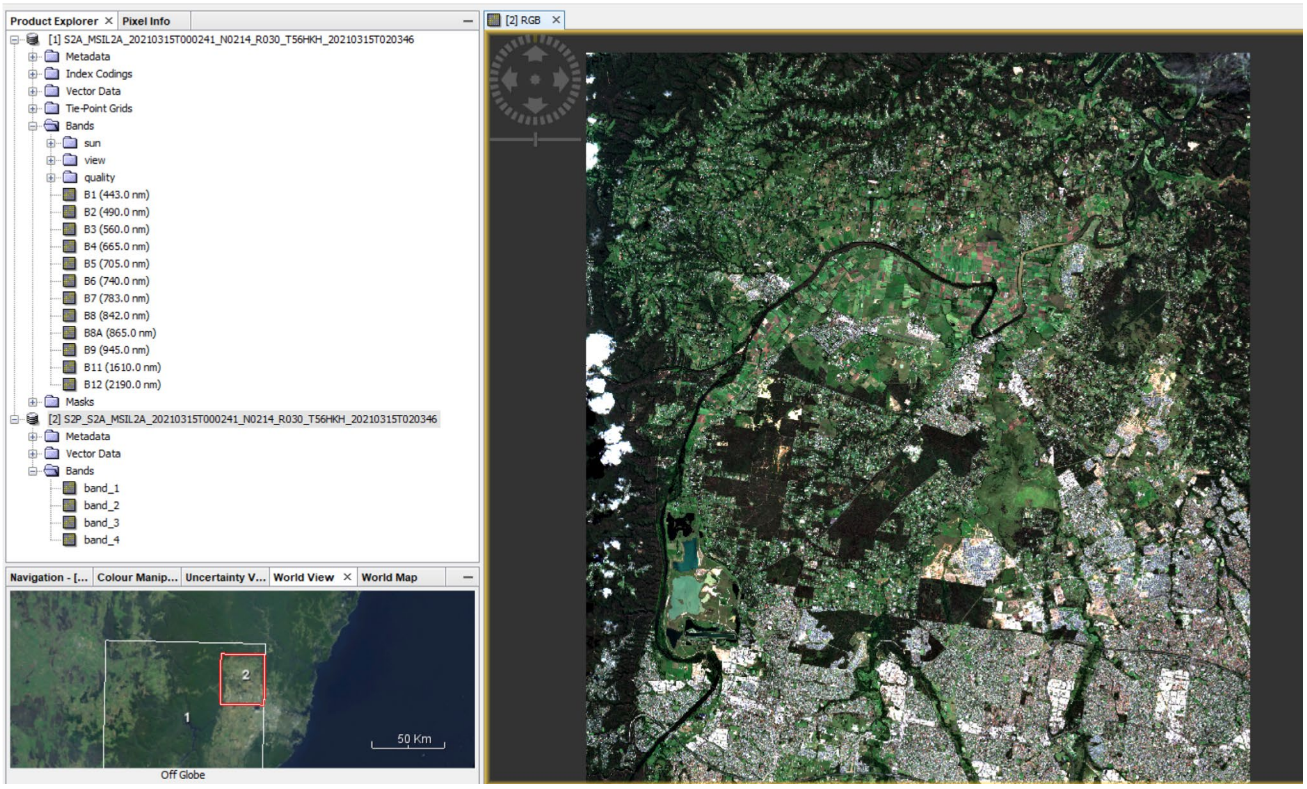
```
docker run -v [local_path]:[container_path] DOCKER_IMAGE  S2-preprocess --input "VALUE" [--bands "XX,XX,XX"] [--landseamask "VALUE"] [--cloudmask "VALUE"] [--AoI "WKT"] [--resolution "VALUE"] [--upsampling "VALUE"] [--downsampling "VALUE"] [--projection "VALUE"] [--output_format "VALUE"] --outdir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

2.6.1 Some examples

Computation of pre-processed S2 in an area close to Sydney using a S2 image from March 2021. The bands selected are B2, B3, B4 and B8 at 20 meters resolution.

```
S2-preprocess --input
/output/S2A_MSIL2A_20210315T000241_N0214_R030_T56HKH_20210315T020346.SAFE --bands
B2,B3,B4,B8 --resolution 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/
```



Computation of pre-processed S2 using a S2 image from March 2021. The bands selected are B2, B3, B4 at 60 meters resolution. Cloud mask is applied.

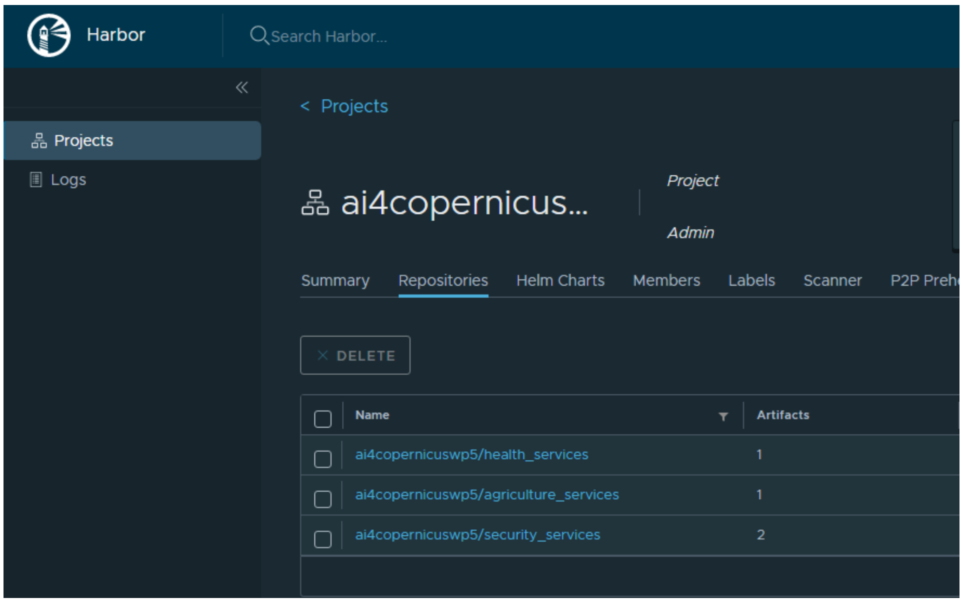
```
S2-preprocess --input
./S2A_MSIL2A_20210315T000241_N0214_R030_T56HKH_20210315T020346.SAFE --bands
B2,B3,B4 --resolution 60 --cloudmask L2A --outdir /output/
```

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)](https://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 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

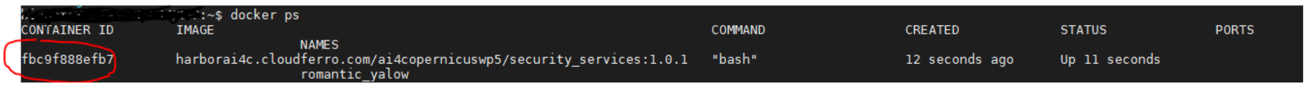
# from Container to Docker Host

```
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`



```
[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
```