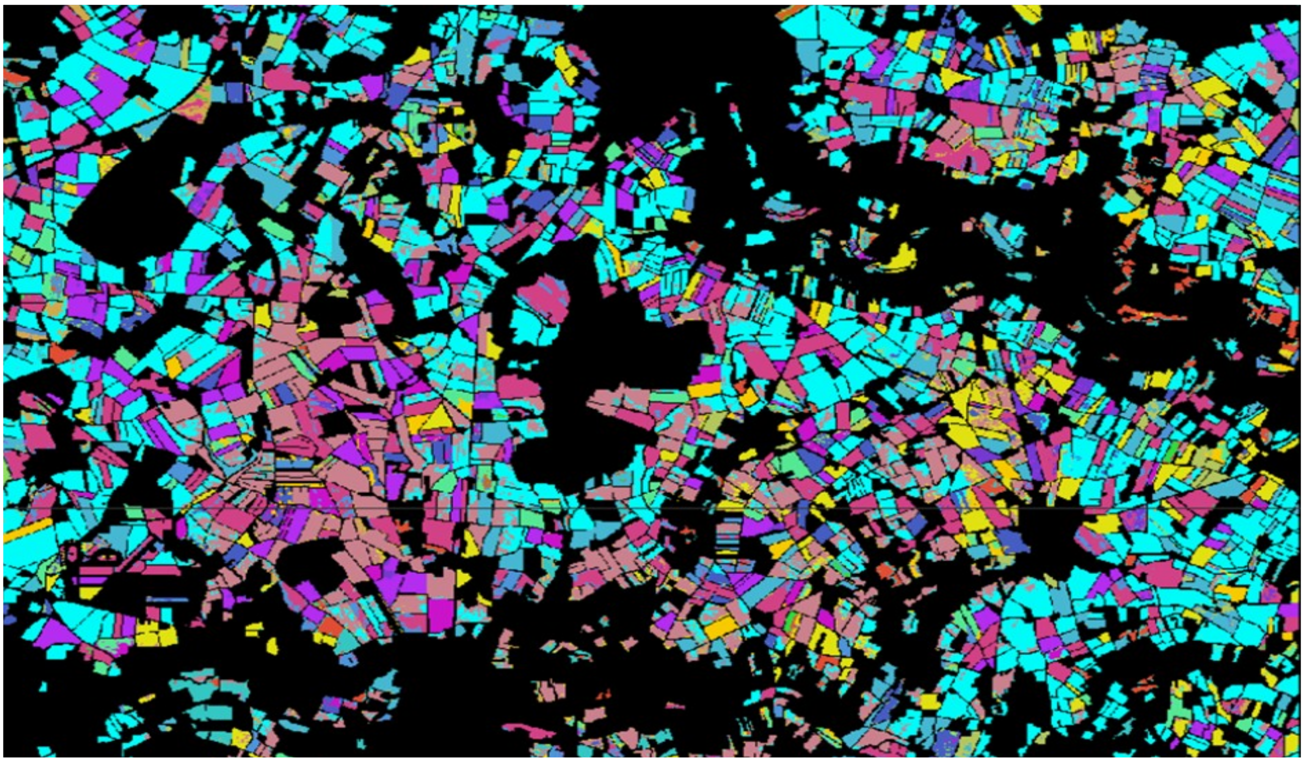


FOSS4G 2019 - Workshop

Orfeo ToolBox and SNAP open source tools for Copernicus Sentinel 1&2 data fusion

Table of Contents

This training kit has been originally created in the frame of the [Copernicus RUS project] for a training session during the IGARSS 2018 Conference.

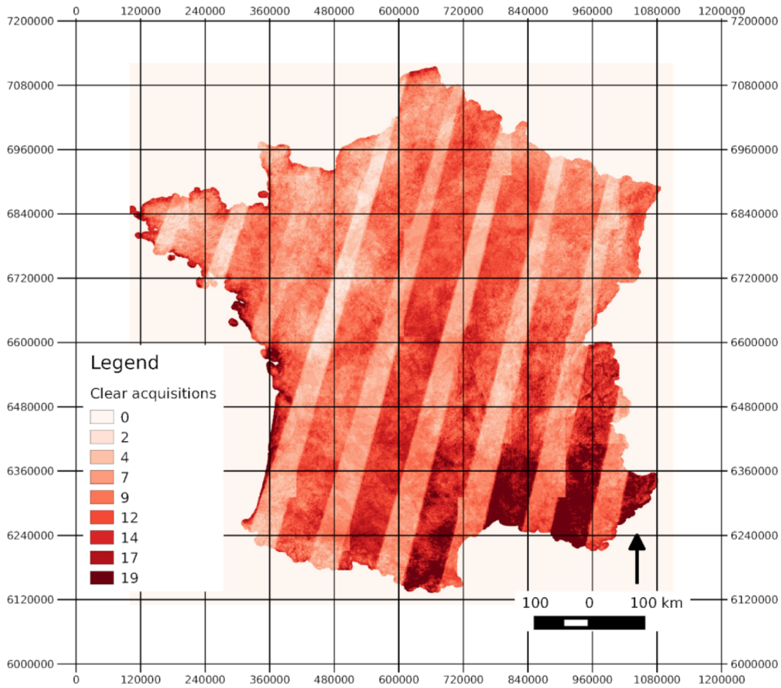
Authors :

- Omar Barrilero, CS Systemes d'Information
- Cécile Cazals, CS Systemes d'Information
- Mickaël Savinaud, CS Systemes d'Information

1 Introduction

The monitoring of the agriculture by using remote sensing techniques is a fundamental support for the EU's Common Agricultural Policy (CAP). The data provided by the Copernicus Sentinel missions thanks to their spatial and temporal resolution and global coverage has demonstrated to be very useful for these activities.

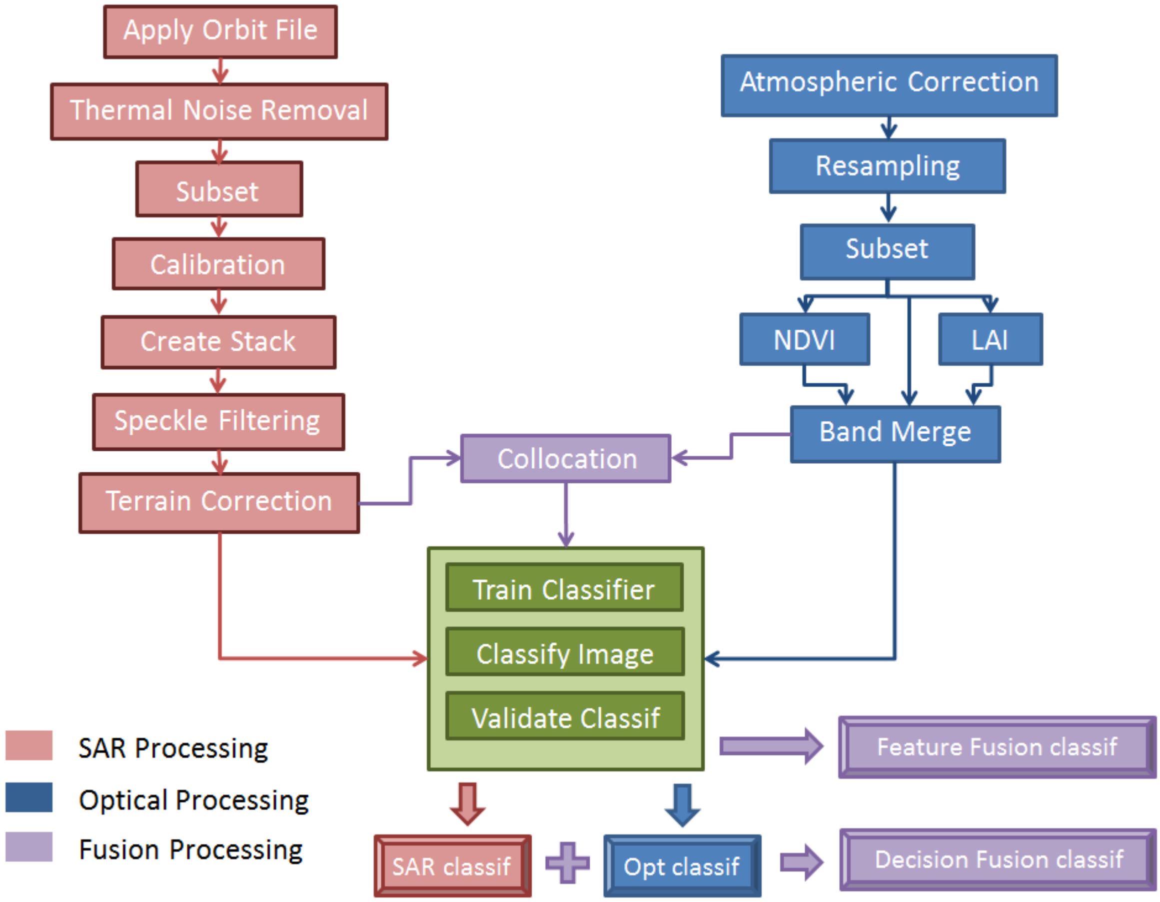
Sentinel-2 data provide accurate crop classifications if the cloud free acquisitions time series is large enough and if the acquisition dates are taken in the good agricultural period. But when the study area is very cloudy, the number of available optical images could be insufficient for this classification. SAR time series from Sentinel-1 can be used combined with the optical Sentinel-2 in these cases to improve the results.



Map of the number of times that every pixel sees the ground taking into account satellite revisit and cloud cover for Landsat during the year 2014. Inglada et al. 2017.

In this tutorial, we propose to illustrate two levels of optical and radar data fusion for agricultural parcel mapping in northern France: a fusion at the attribute level and a fusion at the decision level.

The workflow presented below presents the outlines of the tutorial processing chains. First, crops predictions are computed using optical images only through a supervised random forest classification. Next, the crop classification is conducted using SAR images only. Then the fusion of S1 and S2 data at the attribute level is conducted by computing a classification using both data. Finally, the fusion of optical and SAR classifications is processed, leading to a data fusion at the decision level.



2 Training

2.1 Data used

- Six Sentinel-1A IW GRDH images with VH & VV polarization acquired on March 3, April 25, May 19, June 19, July 18 and August 11, 2016 [downloadable @ https://scihub.copernicus.eu/]

S1A_IW_GRDH_1SDV_20160320T174002_20160320T174027_010456_00F839_F4BD.zip

S1A_IW_GRDH_1SDV_20160425T174004_20160425T174029_010981_0107DF_FC53.zip

S1A_IW_GRDH_1SDV_20160519T174005_20160519T174030_011331_01130E_EDE3.zip

S1A_IW_GRDH_1SDV_20160612T174006_20160612T174031_011681_011E3D_78B8.zip

S1A_IW_GRDH_1SDV_20160718T174008_20160718T174033_012206_012F1A_2538.zip

S1A_IW_GRDH_1SDV_20160811T174010_20160811T174035_012556_013AAC_34EA.zip

- Three Sentinel-2A images acquired on March 12, July 10 and September 8, 2016 [downloadable @ https://scihub.copernicus.eu/]

  S2A_USER_PRD_MSIL1C_PDMC_20160312T233952_R051_V20160312T105037_20160312T105037.SAFE

  S2A_USER_PRD_MSIL1C_PDMC_20160710T185720_R051_V20160710T105247_20160710T105247.SAFE

  S2A_USER_PRD_MSIL1C_PDMC_20160921T193747_R051_V20160908T105022_20160908T105416.SAFE

  The images used in the tutorial are the atmospherically corrected images (level 2A). For correcting them the Sen2Cor application has been used. See *Atmospheric Correction*

- Vector files as ESRI Shapefile for the classification training, validation and masking stored in:
  /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Vector_data

  These in situ data are provided by the Land-Parcel Identification Systems (LPIS), a geographic information system identifying the land use of agricultural parcels within each EU member country. The database is manually filled by farmers. A selection of the 20 majority cultures of the study area was carried out (see Crop table).

  The used data have been processed from the French *Registre Parcellaire Graphique* that is available in http://professionnel.ign.fr/index.html

- *Auxiliary data stored in:*
  /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Auxdata

2.2 Software

We will use :

- Internet browser,
- SNAP 6.0 + Sentinel-1 & Sentinel-2 Toolboxes,
- OTB 6.6.1 (OSGEO Live),
- Sen2Cor 2.5.5

You can install them easily. They are availble in

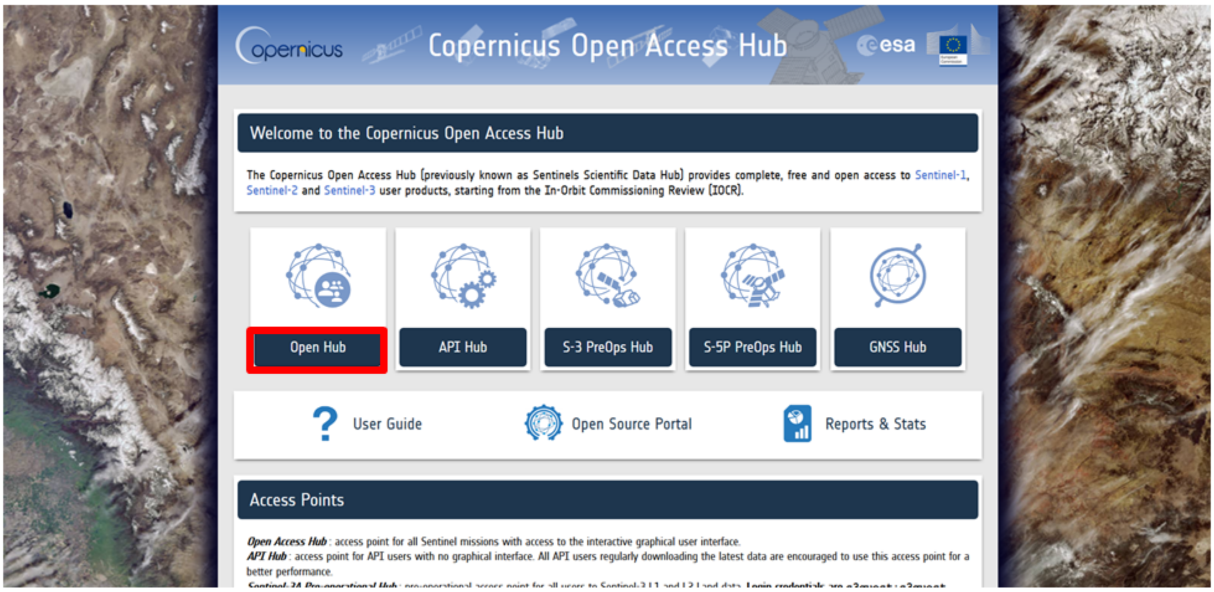
/path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_Software/

3 Step by step – Optical processing

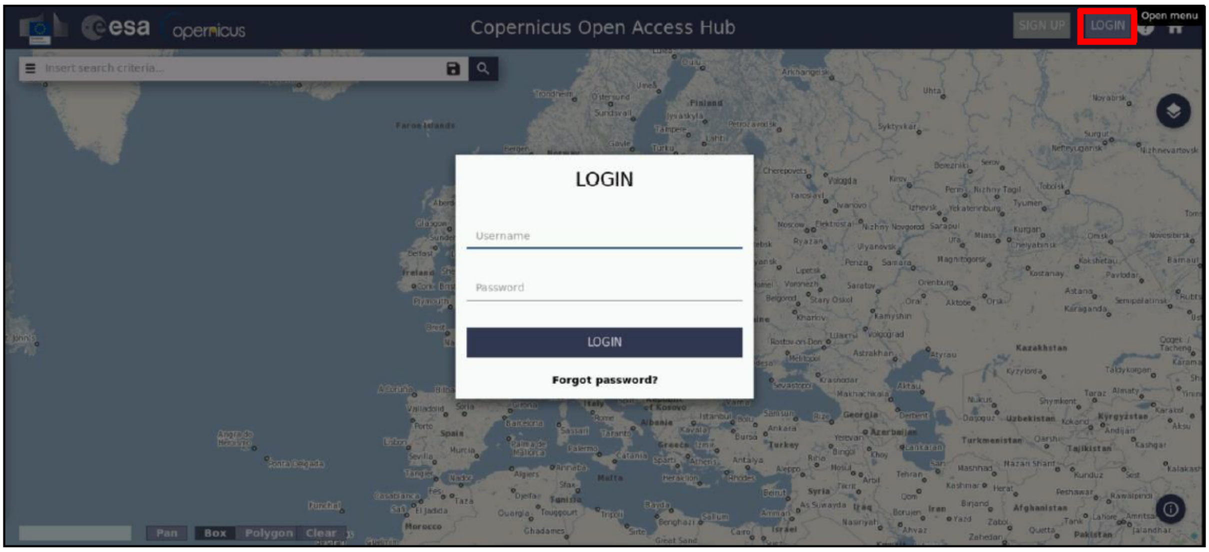
3.1 Data download – ESA SciHUB

In this step, we will download the Sentinel-2 scenes from the Copernicus Open Access Hub using the online interface (**Applications -> Network -> Web Browser,** or click the link below).

Go to https://scihub.copernicus.eu/

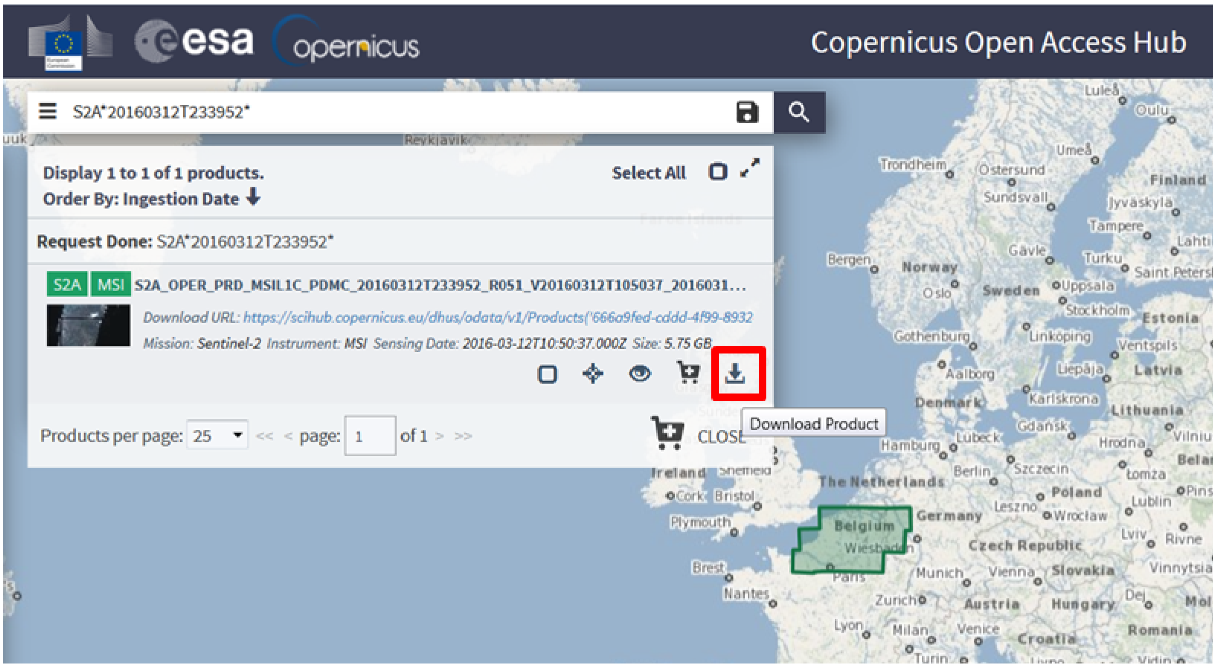


Go to Open HUB. If you already have an account proceed to login. If not, please register in the upper right corner.



Search the desired product by introducing part of the name and clicking on .

Download the product



Follow the same steps for the rest of the S2 images listed in Data used.

Data will be downloaded to your downloaf directory as zip archives. They are also available at:

/path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Original/S2

3.2 Atmospheric Correction

The first step for processing the S2 images is to carry out the atmospheric correction. For doing that, we are going to use Sen2Cor 2.5.5 (http://step.esa.int/main/third-party-plugins-2/sen2cor/)

Sen2Cor is a processor already integrated in the RUS environment which computes the Bottom-Of-Atmosphere (BOA) reflectances from the Top-Of-Atmosphere (TOA) reflectances from the L1C products.

It can be used from SNAP (**Optical -> Thematic Land Processing -> Sen2Cor**) or directly from the command line.

In this tutorial, we are going to do it from command line. Please, note that if you have already the L2A products, then you can skip this step.

As we are working with S2 products from 2016, they are in the old format: they could contain more than one granule per product. For this reason, we are going to apply the Sen2Cor only to the granules we are interested on in order to save time.

The basic command for running Sen2Cor is:

L2A_Process –resolution=XX [fullPathToTheGranule]

So, having into account that the needed granules are T31UDQ and T31UEQ, the commands should be:

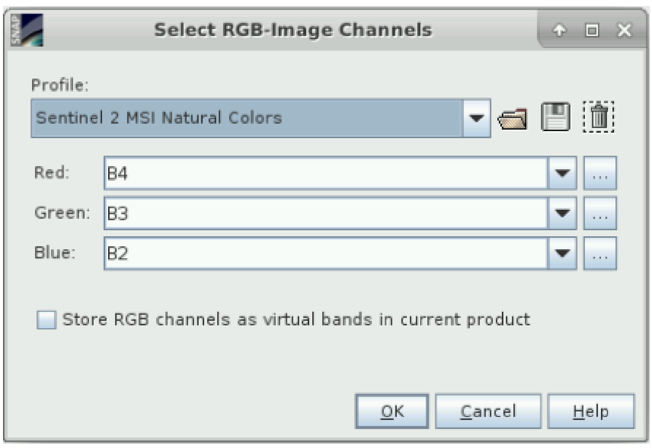
```
L2A_Process           --resolution=20           /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_Tutorial
Kit/S2A_OPER_PRD_MSIL1C_PDMC_20160312T233952_R051_V20160312T105037_20160312T105037.SAFE/GRANULE/S2A_OPER_MSI_L1C_TL_SGS__20160312T181201_A003766_T31UDQ_N02.01
```

Where the resolution has been set at 20 meters because the neural network that is going to be used later to compute the LAI has been trained at that resolution.

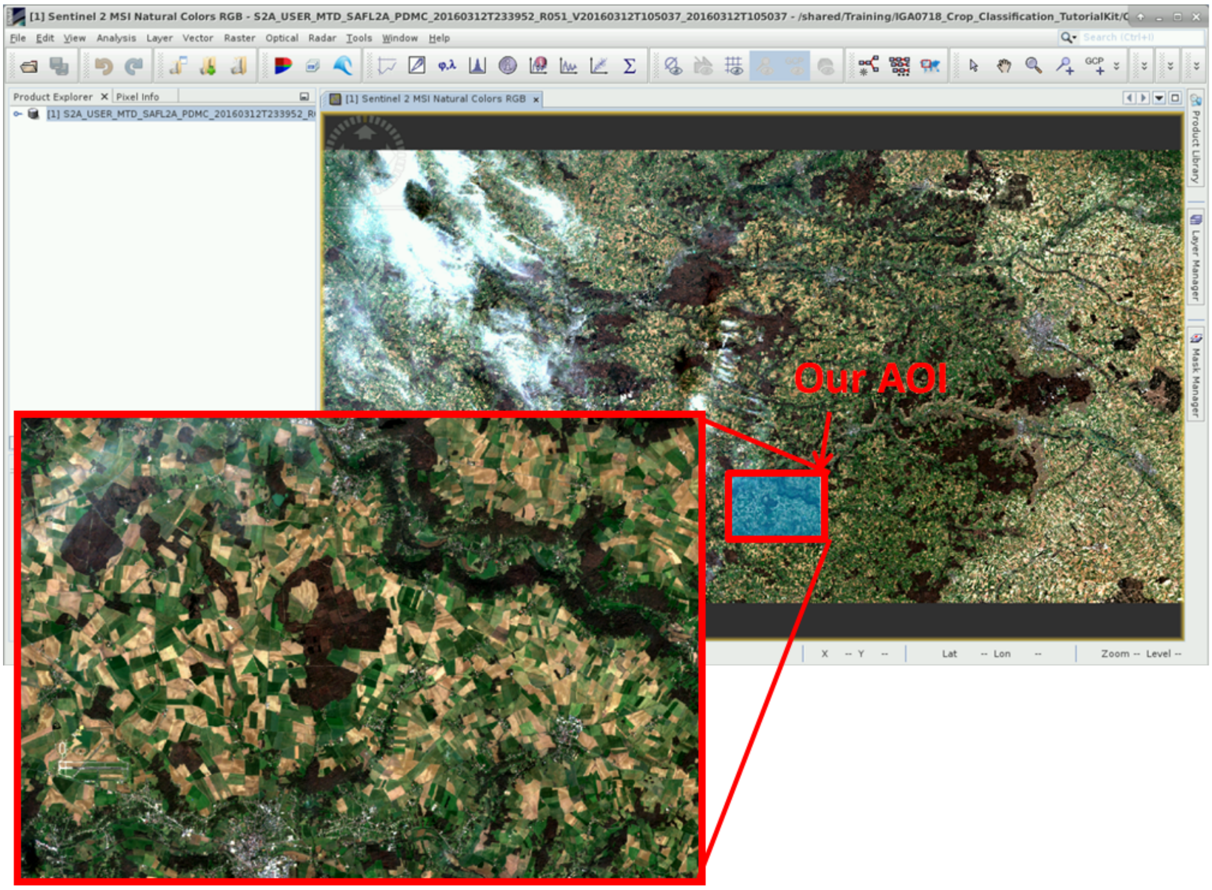
We have to execute it for all the granules T31UDQ and T31UEQ present in the L1C products.

3.3 SNAP – open and explore data

Launch SNAP Desktop ( ); click Open product and navigate to:
/path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Original/S2/
Navigate to each *.SAFE folder and open the corresponding xml file for each image. The name should be
`S2A_USER_MTD_SAFL2A_PDMC_2016XXXXTXXXXXX_R051_V2016XXXXTXXXXXX_2016XXXXTXXXXXX.xml`
(please, note that it is the xml file without the suffix “report”).
To visualize the products, right-click on them and select **Open RGB image window.**



Accept the proposed profile.



3.4 Optical processing graph

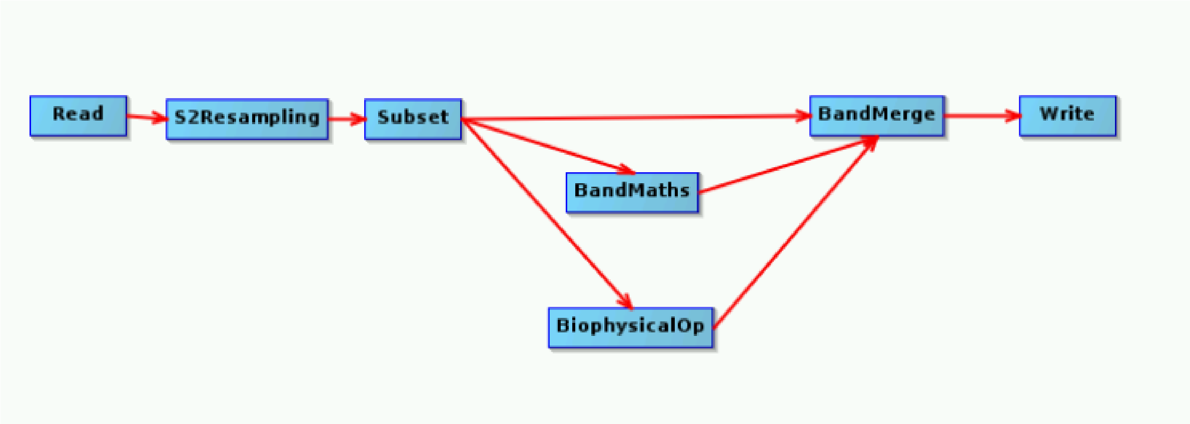
- NOTE 1: GPT

Usage:
 gpt <op>|<graph-file> [options] [<source-file-1> <source-file-2> ...]

Description:
This tool, included in SNAP, is used to execute SNAP raster data operators in batch-mode. The operators can be used stand-alone or combined as a directed acyclic graph (DAG). Processing graphs are represented using XML. More info about processing graphs, the operator API, and the graph XML format can be found in the SNAP documentation.

As we are going to apply the same preprocessing steps to all the S2 images, we have defined them into a graph and we will execute the graph directly from the command line by using gpt.

The processing chain is:



The xml graph can be found in:

/path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Scripts/OpticalPreprocessingGraph.xml

For creating it, we have to edit some parameters directly in the xml since there are some operators that are not fully supported currently by the Graph Builder in SNAP Desktop. Moreover, two parameters have been defined in the xml file in a ‘special’ way in order to allow modifying them when executing the graph from the command line. These parameters are the “inputFile” and “outputFile”. In the xml file they have been defined with a prefix ‘$’ in order to indicate that we have to define them when running the graph.

```
<graph id="Graph">
  <version>1.0</version>
  <node id="Read">
    <operator>Read</operator>
    <sources/>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <file>$inputFile</file>
    </parameters>
  </node>
  <node id="S2Resampling">
    <operator>S2Resampling</operator>
    <sources>
      <sourceProduct refid="Read"/>
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <resolution>20</resolution>
    </parameters>
  </node>
  <node id="Subset">
    <operator>Subset</operator>
    <sources>
      <sourceProduct refid="S2Resampling"/>
```

The modules in the graph are:

Read: To read the input. We can define directly the source or indicate $inputFile and select the input when executing.

```
<node id="Read">
  <operator>Read</operator>
  <sources/>
  <parameters class="com.bc.ceres.binding.dom.XppDomElement">
    <file>$inputFile</file>
  </parameters>
</node>
```

S2Resampling: to transforme the S2 product to a single-size product. The differences with the generic resampling operator are explained in Resampling in SNAP. The output resolution selected is 20 m, the rest of the configuration parameters are the default ones.

```xml
<node id="S2Resampling">
  <operator>S2Resampling</operator>
  <sources>
    <sourceProduct refid="Read"/>
  </sources>
  <parameters class="com.bc.ceres.binding.dom.XppDomElement">
    <resolution>20</resolution>
  </parameters>
</node>
```

Subset: to work only with our interest area (this way all the process will be faster) and to keep only some bands (for using less disk space). The configuration is:

Spatial Subset:
- Scene start X: 4917
- Scene start Y: 3919
- Width: 1169 (Scene end X: 6085)
- Height: 795 (Scene end Y: 4713)

Band Subset:
- the reflectance bands: B2, B3, B4, B5, B6, B7, B8A, B11, B12
- the sun angles and the mean viewing angles.

```xml
<node id="Subset">
  <operator>Subset</operator>
  <sources>
    <sourceProduct refid="S2Resampling"/>
  </sources>
  <parameters class="com.bc.ceres.binding.dom.XppDomElement">
    <sourceBands>B2,B3,B4,B5,B6,B7,B8A,B11,B12,view_zenith_mean,view_azimuth_mean,sun_zenith,sun_azimuth</sourceBands>
    <region>4917,3919,1169,795</region>
    <geoRegion/>
    <subSamplingX>1</subSamplingX>
    <subSamplingY>1</subSamplingY>
    <fullSwath>false</fullSwath>
    <tiePointGridNames/>
    <copyMetadata>true</copyMetadata>
  </parameters>
</node>
```

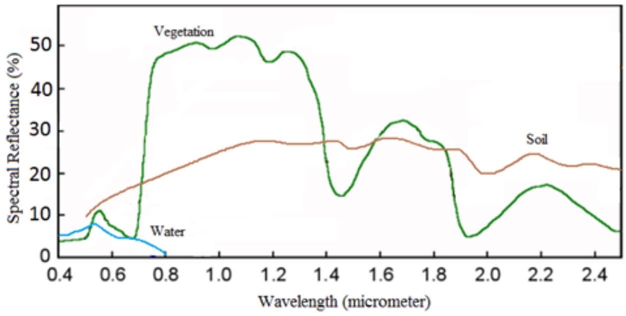
BandMaths: to compute the NDVI

- NOTE 2: NDVI

Radiometric indices are a convenient way to resume information. They allow exploiting the particular spectral properties of vegetation.

One of the most used for vegetation monitoring is the NDVI, which composes a measurement for the photosynthetic activity and is strongly in correlation with density and vitality of the vegetation. It is based on the abrupt rise of the reflection level at 0,7 µm showed in the spectral signature of vegetation. So much more active the chlorophyll of the plants, so much bigger is the rise.

$$\mathrm{NDVI} = \frac{(\mathrm{NIR} - \mathrm{Red})}{(\mathrm{NIR} + \mathrm{Red})}$$

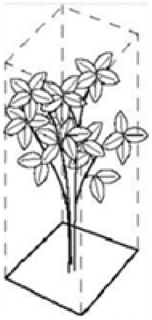


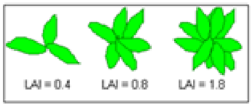

```xml
<node id="BandMaths">
  <operator>BandMaths</operator>
  <sources>
    <sourceProduct refid="Subset"/>
  </sources>
  <parameters class="com.bc.ceres.binding.dom.XppDomElement">
    <targetBands>
      <targetBand>
        <name>NDVI</name>
        <type>float32</type>
        <expression>(B8A-B4)/(B8A+B4)</expression>
        <description/>
        <unit/>
        <noDataValue>0.0</noDataValue>
      </targetBand>
    </targetBands>
    <variables/>
  </parameters>
</node>
```

BiophysicalOp: to compute the LAI. See http://step.esa.int/main/toolboxes/sentinel-2-toolbox/sentinel-2-toolbox-features/ for more information about the LAI algorithm integrated in SNAP.

- NOTE 3: LAI

The Leaf Area Index is the half of the total developed area of green leaves per unit of ground horizontal surface area. It is an indication of canopy density and can be used to predict productivity and crop growth.






```
<node id="BiophysicalOp">
  <operator>BiophysicalOp</operator>
  <sources>
    <sourceProduct refid="Subset"/>
  </sources>
  <parameters class="com.bc.ceres.binding.dom.XppDomElement">
    <computeLAI>true</computeLAI>
    <computeFapar>false</computeFapar>
    <computeFcover>false</computeFcover>
    <computeCab>false</computeCab>
    <computeCw>false</computeCw>
  </parameters>
</node>
```

BandMerge: to prepare the output product we have to merge the reflectance bands, the NDVI and the LAI in a single product.

```
<node id="BandMerge">
  <operator>BandMerge</operator>
  <sources>
    <sourceProduct refid="Subset"/>
    <sourceProduct.1 refid="BandMaths"/>
    <sourceProduct.2 refid="BiophysicalOp"/>
  </sources>
  <parameters class="com.bc.ceres.binding.dom.XppDomElement">
    <sourceBands>B2,B3,B4,B5,B6,B7,B8A,B11,B12,NDVI,lai</sourceBands>
    <geographicError>1.0E-5</geographicError>
  </parameters>
</node>
```

Write: to write the output in BEAM-DIMAP format.

```
<node id="Write">
  <operator>Write</operator>
  <sources>
    <sourceProduct refid="BandMerge"/>
  </sources>
  <parameters class="com.bc.ceres.binding.dom.XppDomElement">
    <file>$outputFile</file>
    <formatName>BEAM-DIMAP</formatName>
  </parameters>
</node>
```

3.5 Running the optical preprocessing

To run the graph, follow the steps below:

Close SNAP and open a terminal:

Execute the command for the first S2 image:

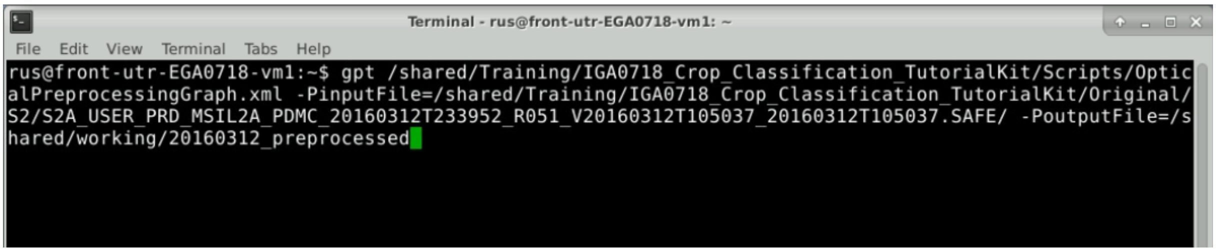
```
gpt [graph] –PinputFile=[input] –PoutputFile=[output]
```

Where:

```
[graph] = /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Scripts/OpticalPreprocessingGraph.xml
```

```
[input] = /path/to/FOSS4G2019_SNAP&OTB-Crop-Classification_TutorialKit/Original/S2/S2A_USER_PRD_MSIL2A_PDMC_20160312T233952_R051_V20160312T105037_20160312T105037.SAFE/
```

```
[output] = /path/to/20160312_preprocessed
```

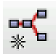


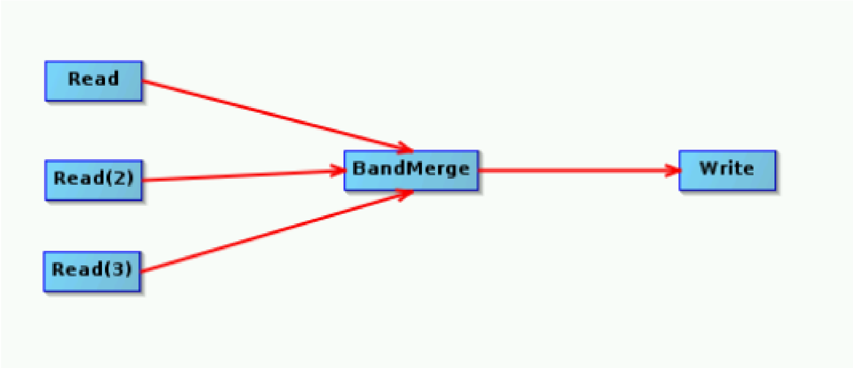
Repeat the process for the rest of the images:

- Second image:
  ```
  [graph] = /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Scripts/OpticalPreprocessingGraph.xml
  [input] = /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Original/S2/S2A_USER_PRD_MSIL2A_PDMC_20160710T185720_R051_V20160710T105247_20160710T105247.SAFE/
  [output] = /shared/working/20160710_preprocessed
  ```
- Third image:
  ```
  [graph] = /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Scripts/OpticalPreprocessingGraph.xml
  [input] = /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Original/S2/S2A_USER_PRD_MSIL2A_PDMC_20160921T193747_R051_V20160908T105022_20160908T105416.SAFE/
  [output] = /shared/working/20160921_preprocessed
  ```

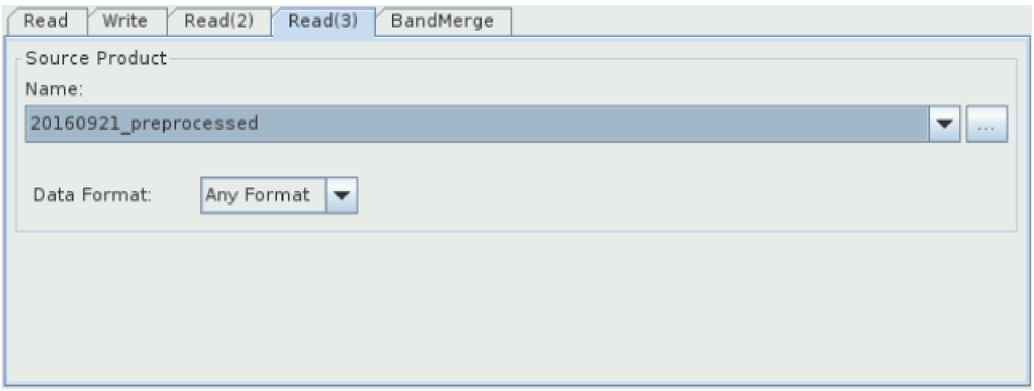
3.6 Prepare data for the classification

Now, we have the three S2 images preprocessed in individual files, but we need all the bands together in an only file (geotiff) in order to be able to process it easily in OTB. For doing that, we are going to open SNAP again and we will create and execute a graph:

Open the Graph Builder  and reproduce the following graph:

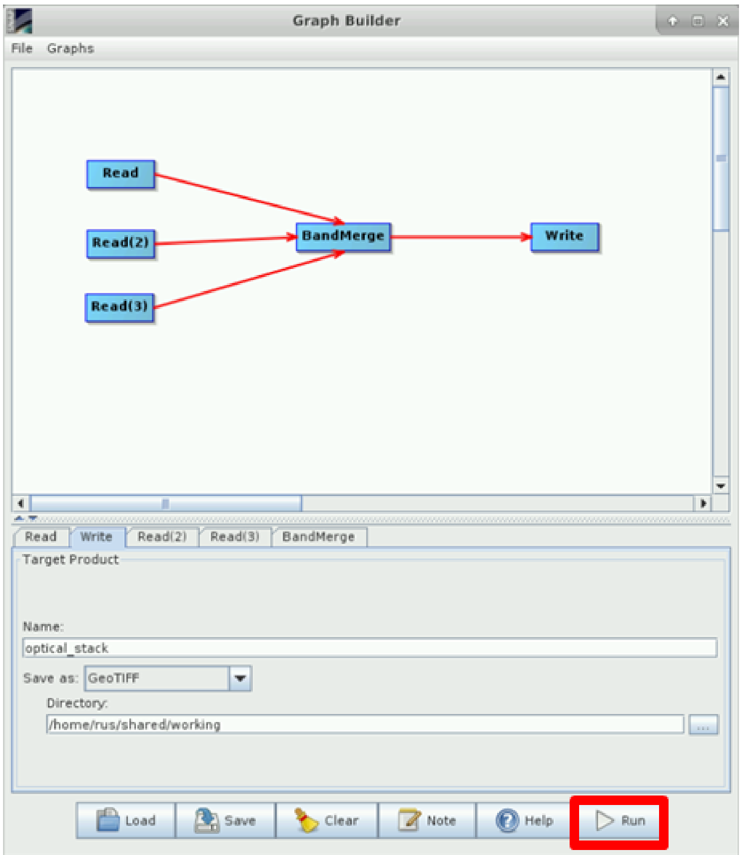


In each read tab, select one of the preprocessed product:



Choose the name in the write tab and select GeoTIFF format:

Run the graph and close SNAP when it has finished.



3.7 Classification in OTB

Supervised classifications are widely used to map land cover using remote sensing data. Classification process consists of 3 main stages:

- Training of the classifier by using the in situ data. The output of this step will be the classification model.
- The classification is applied to the whole image. The output will be the classified image and a confidence map.
- The classification prediction is evaluated over a large area. The output of this step will be a confusion matrix.

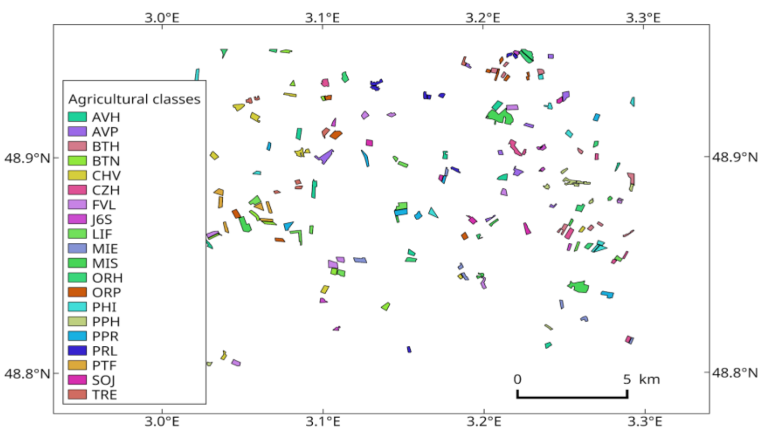
The Orfeo ToolBox classification framework is used in this part.

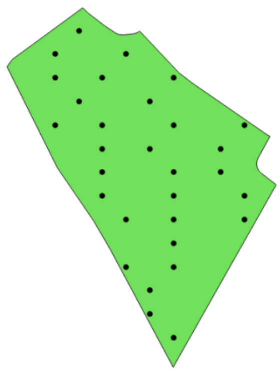
- NOTE 2: OTB

Orfeo ToolBox (OTB) is an open-source project for state-of-the-art remote sensing. Built on the shoulders of the open-source geospatial community, it can process high resolution optical, multispectral and radar images at the terabyte scale. A wide variety of applications are available: from ortho-rectification or pansharpening, all the way to classification, SAR processing, and much more!
All of OTB's algorithms are accessible from Monteverdi, QGIS, Python, the command line or C++.
https://www.orfeo-toolbox.org/

3.7.1 Train image classifier

The training samples are available in the folder *Vector_data/Train/train_plots_samples.shp.* It consists of 779 points for each of the 20 agricultural classes.
The following map presents the spatial coverage of the training samples. Agricultural codes (CODE_CULT) details are available in Crop table.



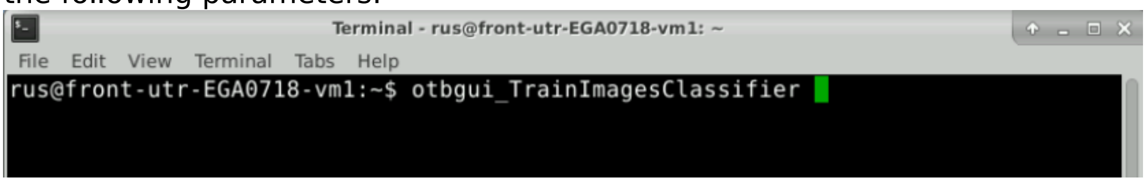


Spatial repartition of training sample inside plots

Random Forest classifier is well known for being a good and quick classifier; we will use it for this application.
Orfeo ToolBox library propose a classifier training function that can be launched both in a graphical interface and in command line.

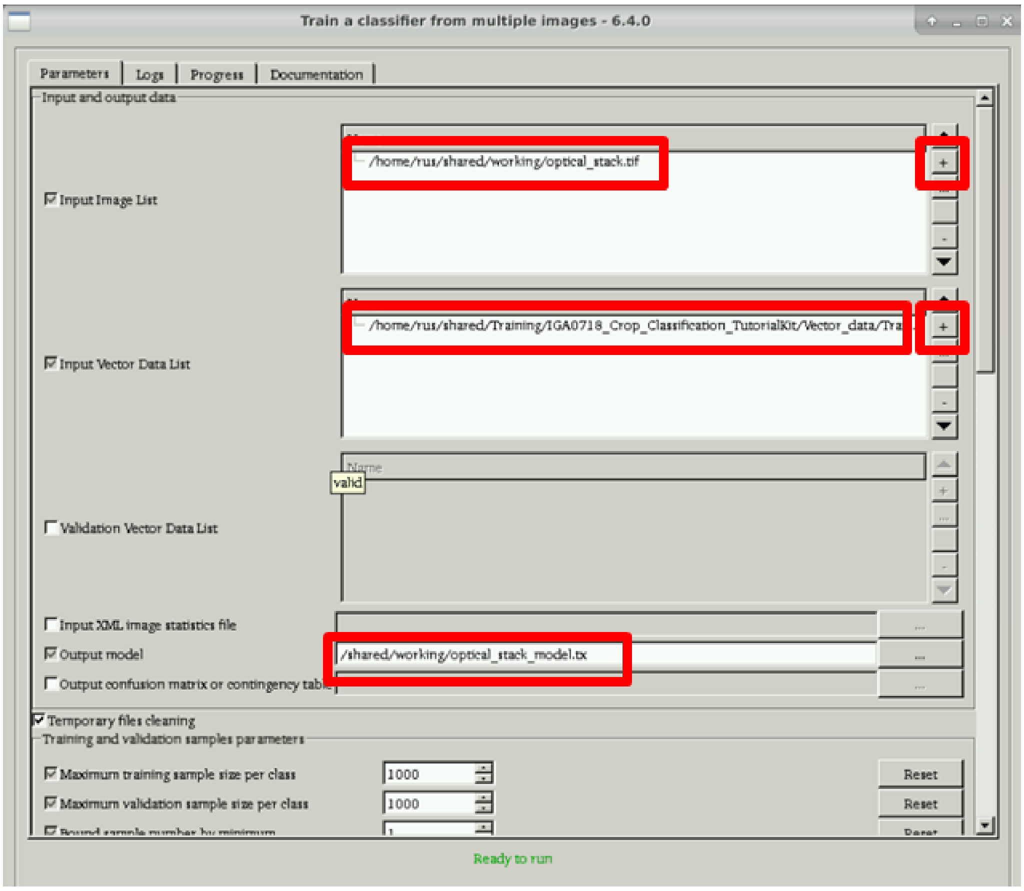
- Graphical interface:

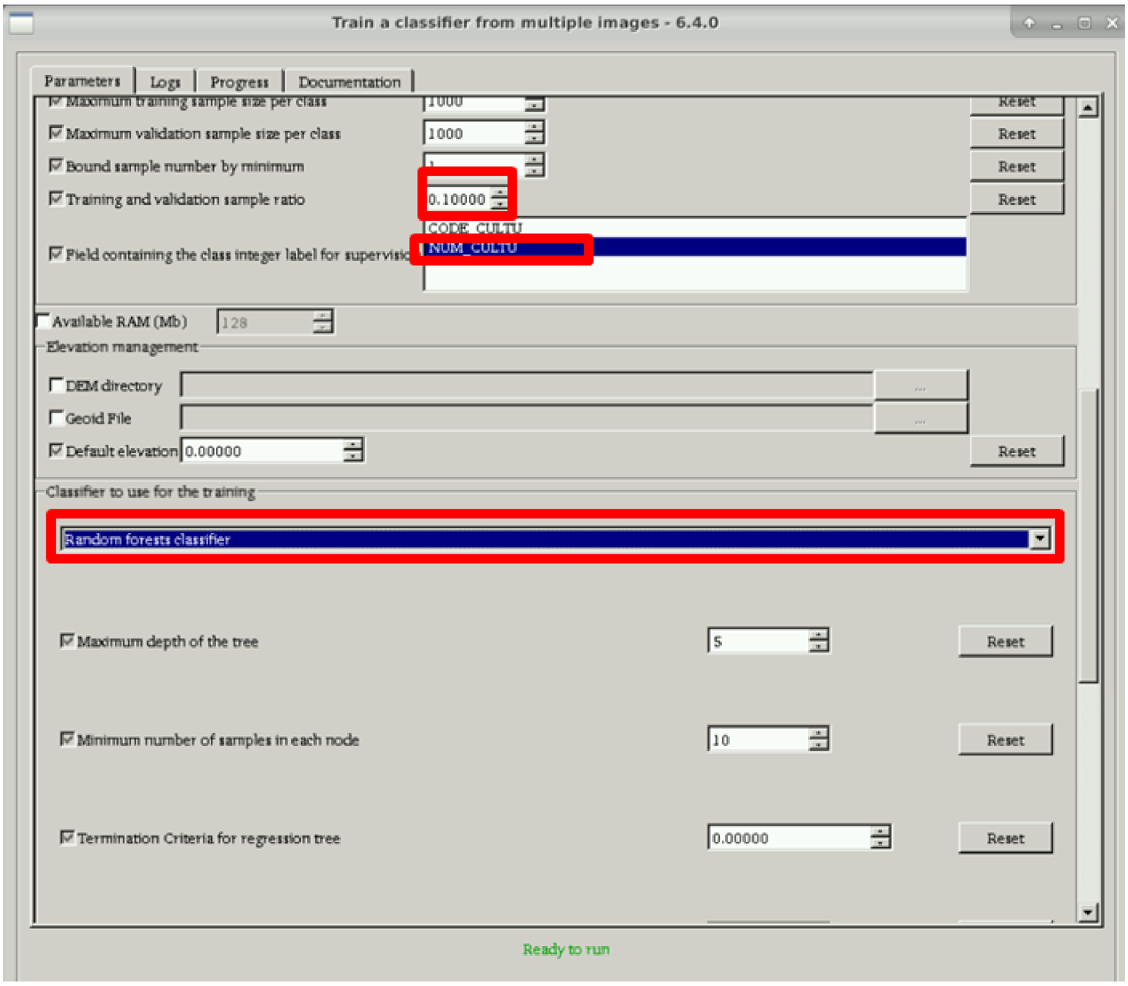
  Open a terminal and run **otbgui_TrainImagesClassifier**, in the interface, set the following parameters:



  - Input images list : by clicking on the "+" add the image optical_stack.tif
  - Input vector list : by clicking on the "+" add the file vector_data/Train/train_plot_samples.shp
  - Output model : save the classifier model in a file optical_stack_model.txt

- Field containing the class integer label for supervision : “NUM_CULTU”
- Training and validation sample ratio: 0.1, in order to keep samples for the training, a separate cross-correlation will be conducted.
- Classifier to use for the training: “Random Forest Classifier”. We will use the default parameters.





- Command line:

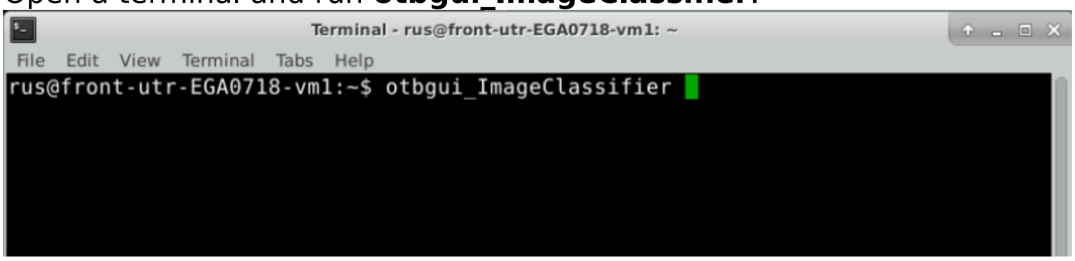
Launch the following command in a terminal:

```
otbcli_TrainImagesClassifier -io.il /shared/working/optical_stack.tif
-io.vd /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Vector_data/Train/train_plots_samples.shp
-sample.vfn NUM_CULTU -sample.vtr 0.1 -classifier rf
-classifier.rf.nbtrees 100 -io.out /shared/working/optical_stack_model.txt
```

3.7.2 Image classification

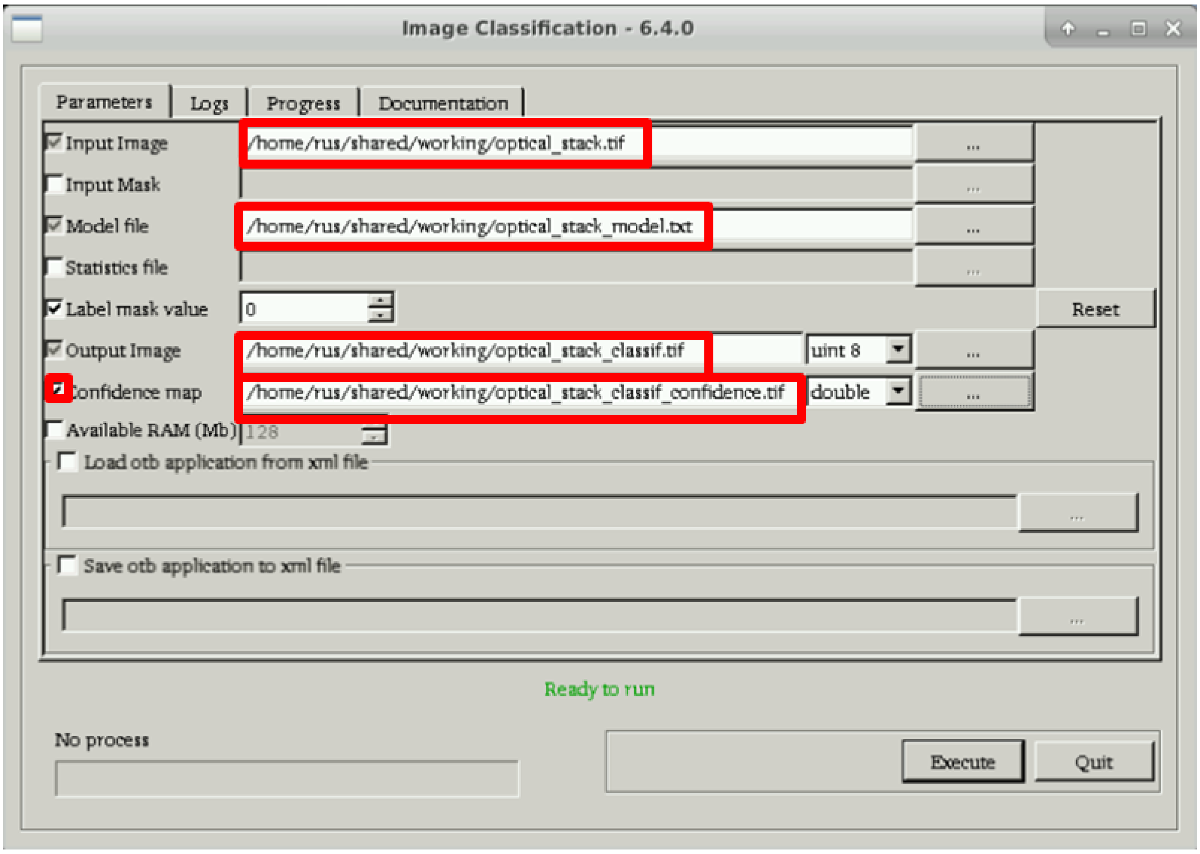
The classification model built in the training step is now applied to the whole image. Depending on the classifier, the confidence map can also be provided by this step. This map will help us to analyse the classification result.

- Graphical interface:
  Open a terminal and run **otbgui_ImageClassifier**.



Set the following parameters:

- Input images list : by clicking on the “+” add the image optical_stack.tif
- Model file : by clicking on the “+” add the image optical_stack_model.txt
- Output image : save the prediction results in optical_stack_classif.tif
- Confidence map : check the box and save the confidence result in optical_stack_classif_confidence.tif

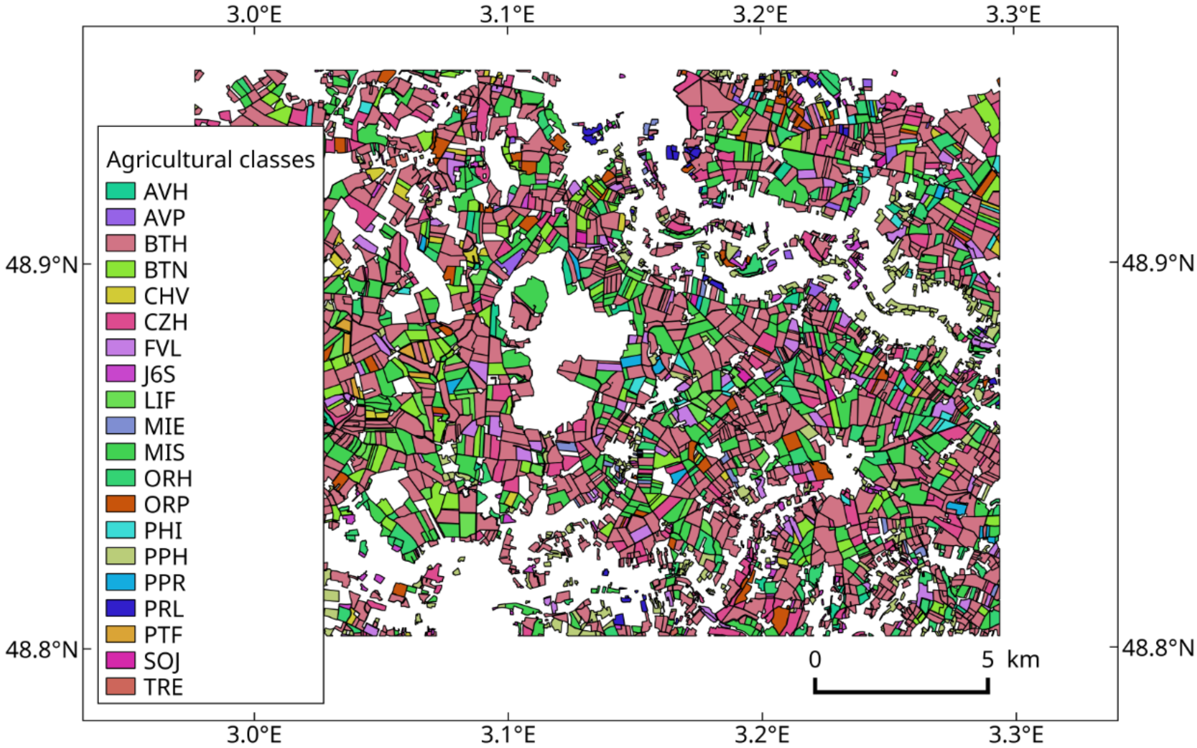


- Command line :

```
otbcli_ImageClassifier
-in /path/to/working_dir/optical_stack.tif
-model /path/to/working_dir/optical_stack_model.txt
-out /path/to/working_dir/optical_stack_classif.tif
-confmap /path/to/working_dir/optical_stack_classif_confidence.tif
```

3.7.3 Result analysis

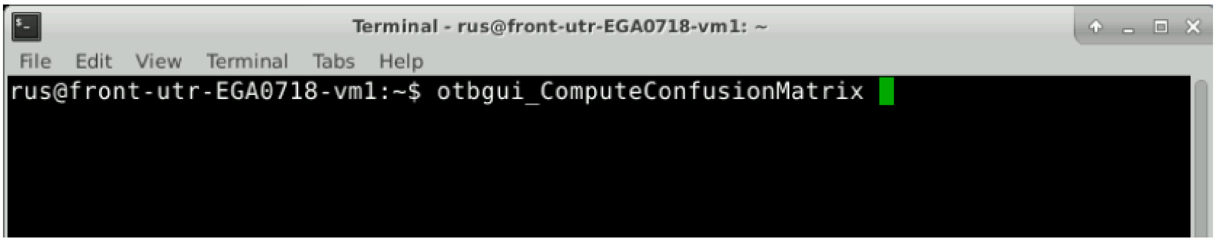
The predictions of the classification have to be compared to a larger ground truth data set. The vector file Vector_data/Valid/valid_plots.shp contains parcels and crop types for the 20 agricultural classes over the whole subset area.
The following map presents the spatial coverage of the validation samples. Agricultural codes (CODE_CULT) details are available in Crop table.



A cross-validation can be conducted using those data in OTB with the *ComputeConfusionMatrix* application.

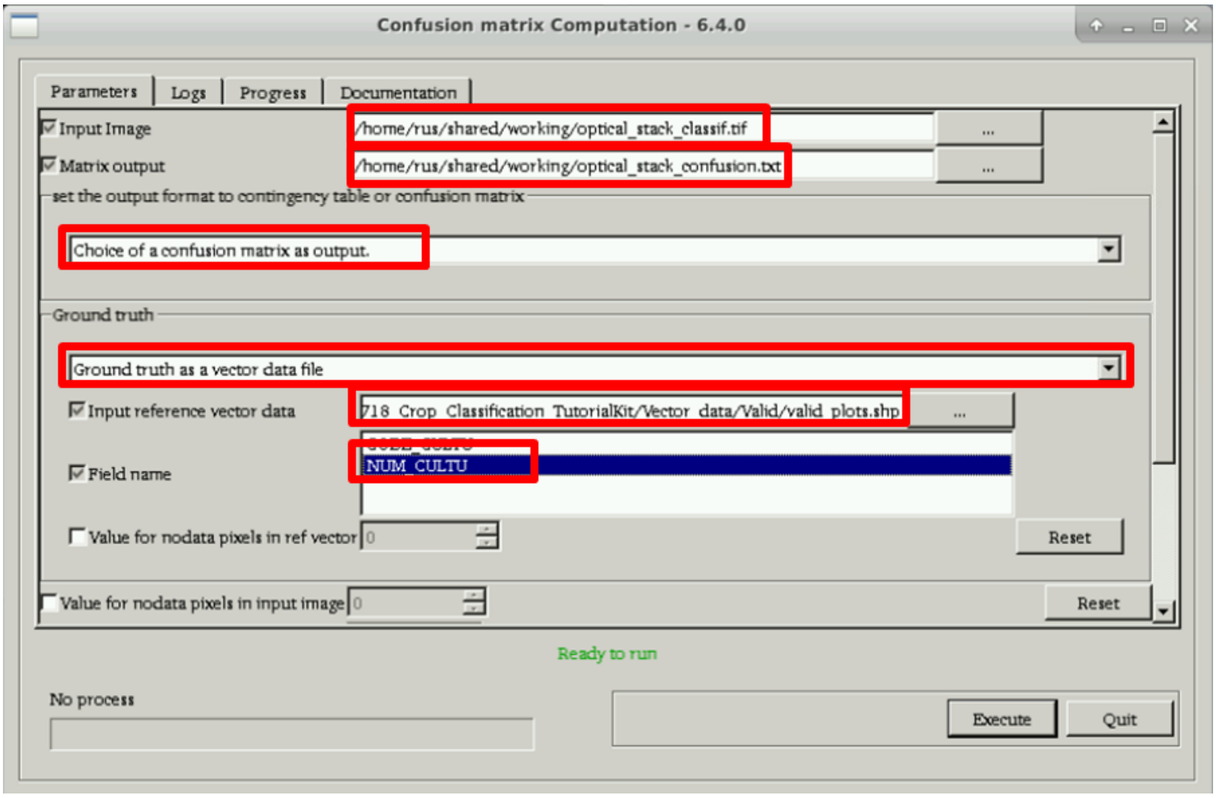
- Using the graphical interface:

Open a terminal and run *otbgui_ComputeConfusionMatrix*:

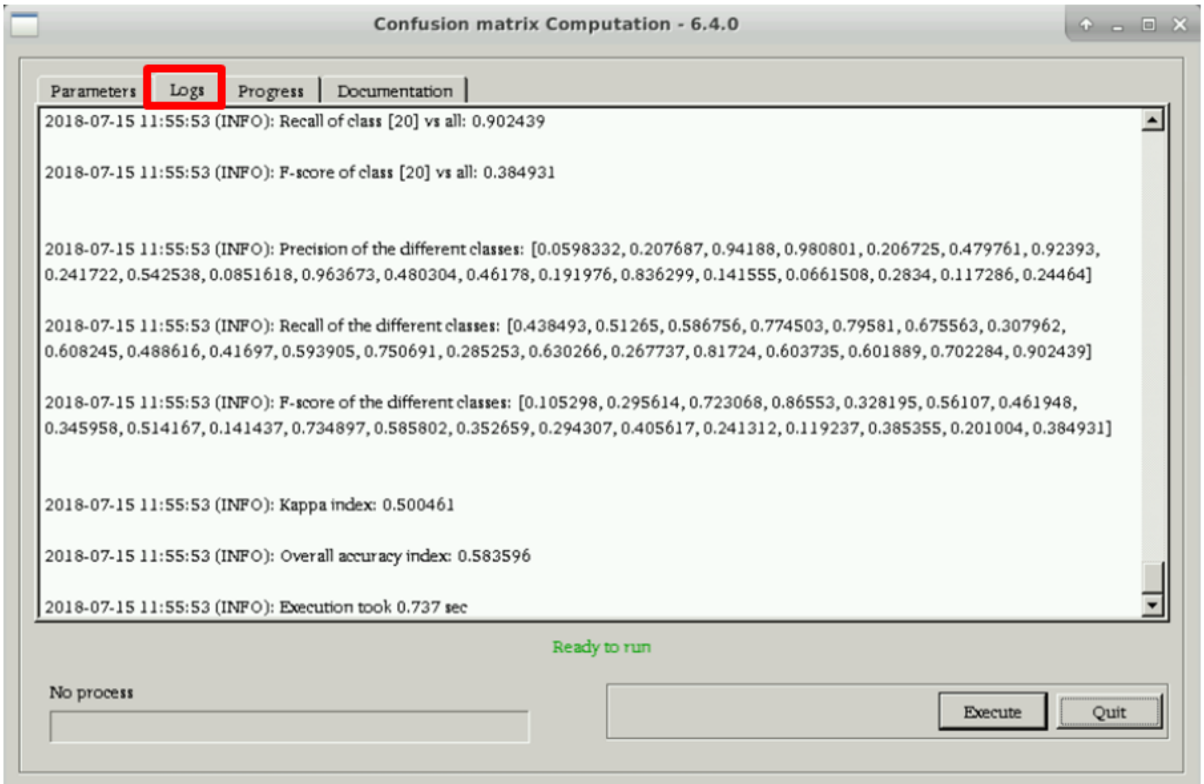


Set the following parameters:

- Input image: by clicking on the "+" add the image optical_stack_classif.tif
- Output matrix : save the confusion matrix to optical_stack_confusion.txt
- Choice of a confusion matrix as output
- Ground truth as a vector data file
- Input reference vector data: /path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/ Vector_data/Valid/valid_plots.shp
- Field name: NUM_CULTU



In the *Logs* tab, some statistics are summarized.



Global statistics:
Overall Accuracy: Number of correct predictions / Total number of samples
Kappa: More advanced prediction quality index which measures inter-rate agreement for qualitative (categorical) items. The following table 2 presents arbitrary guidelines to characterize kappa index.

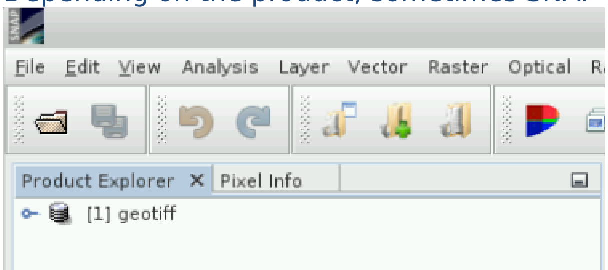
κ **value**	**Interpretation result**
< 0	Disagreement
0.0 — 0.20	Very weak agreement
0.21 — 0.40	Weak agreement
0.41 — 0.60	Moderate agreement
0.61 — 0.80	Strong agreement
0.81 — 1.00	Almost perfect agreement

Statistics by classes:
Precision: Number of correct predictions of class A / number of samples predicted as class A.
Recall: Number of correct predictions of class A / number of samples that belong to class A

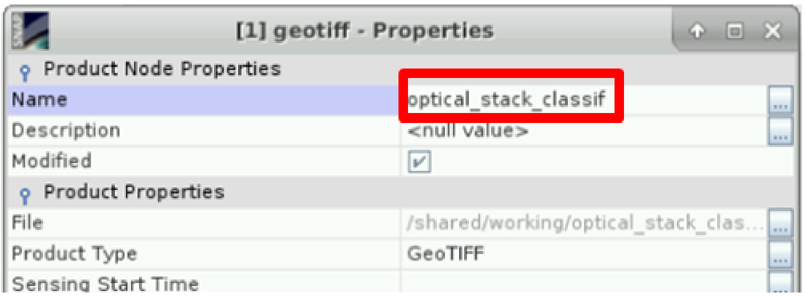
3.8 Visualize the results in SNAP

Open again SNAP and open the optical_stack_classif.tif and optical_stack_classif_confidence.tif

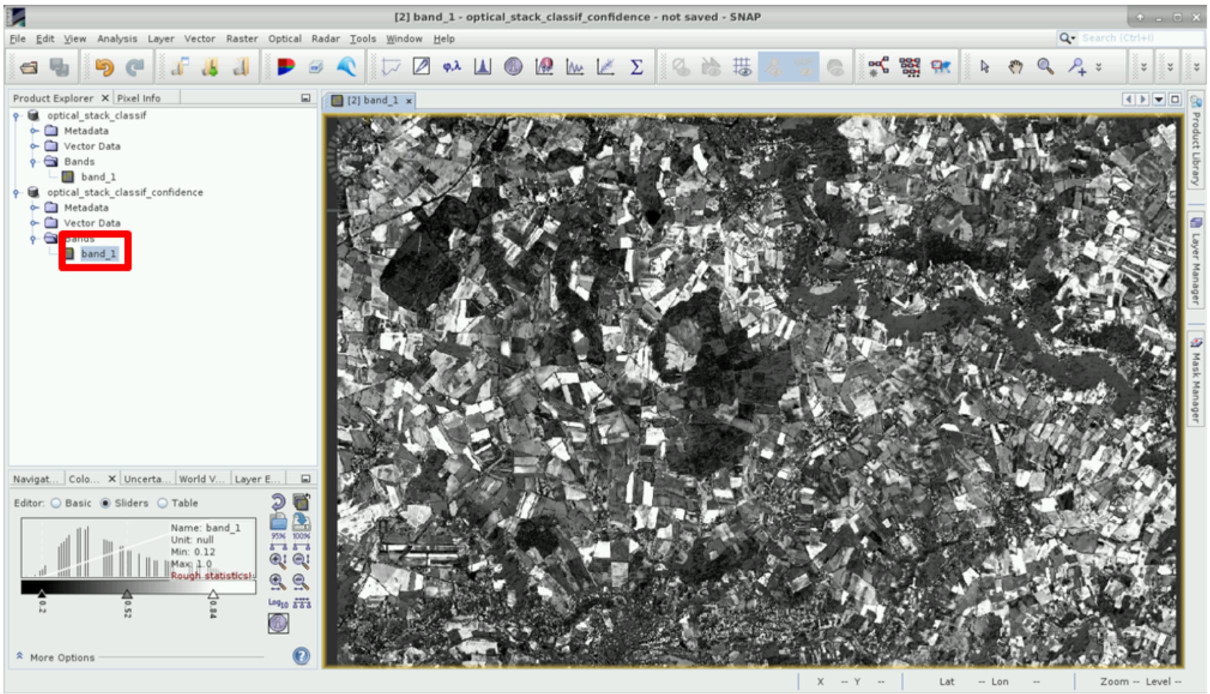
TIP 1:
Depending on the product, sometimes SNAP does not load properly the name:



You can modify it: right-click on the product -> Properties

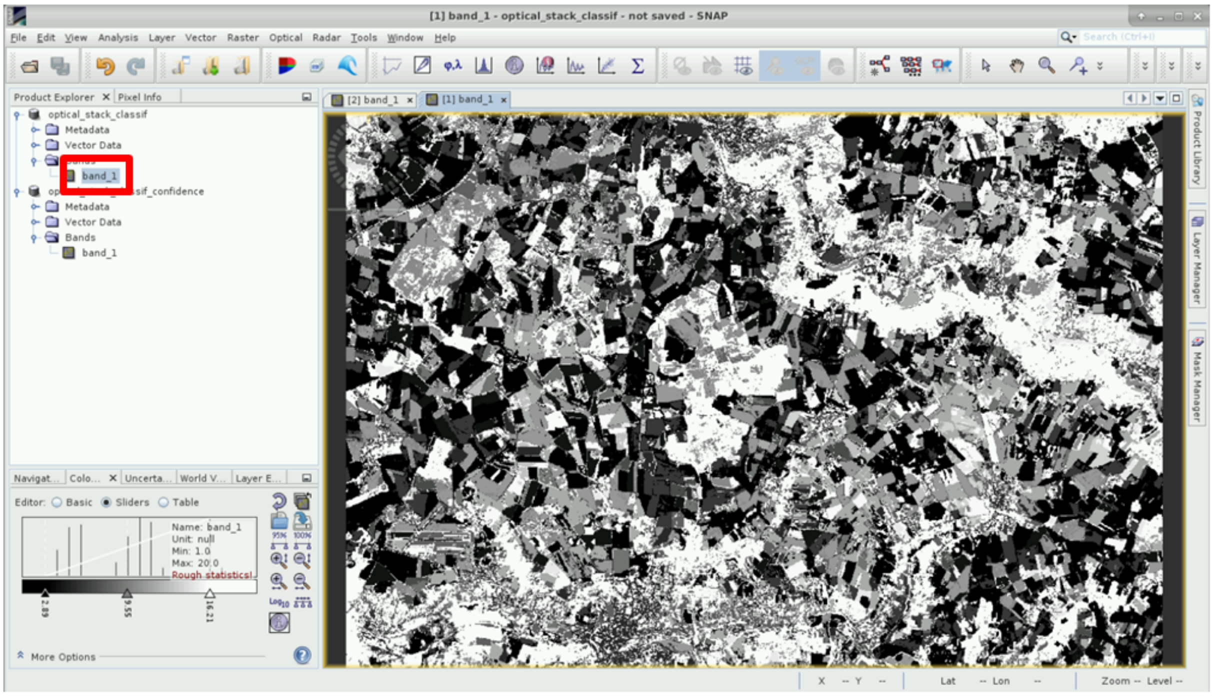


Open the confidence band: double-click on band_1



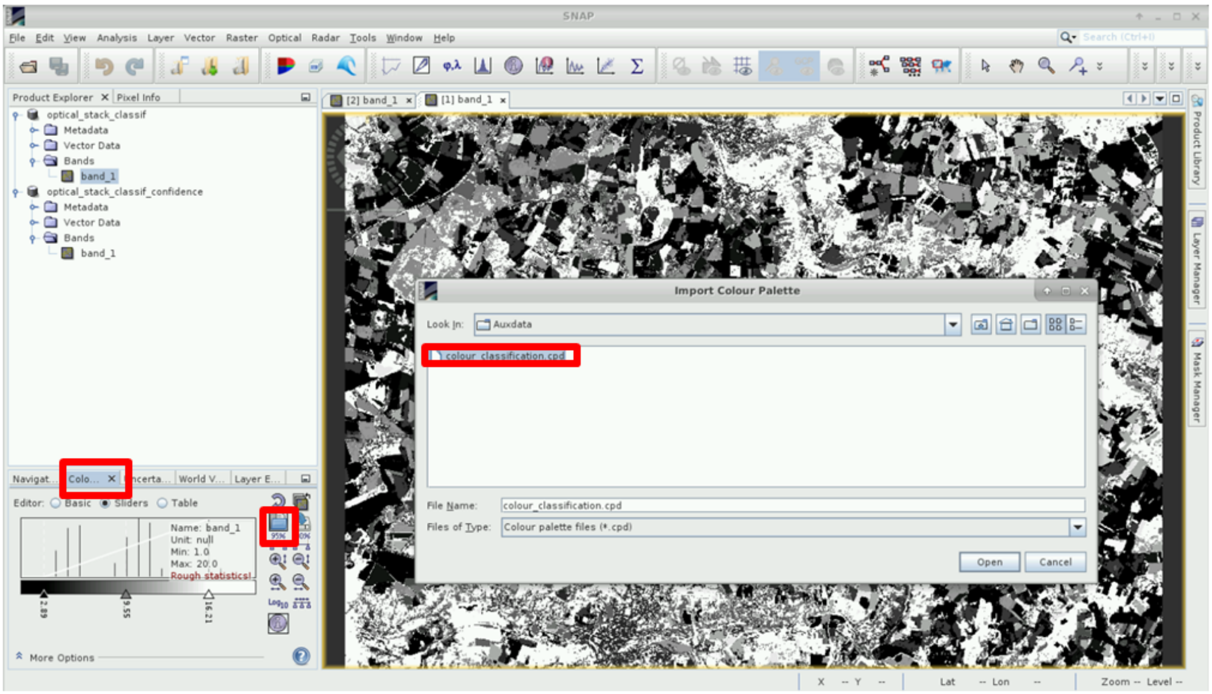
The values are between 0 and 1 (closer to 1: more confidence)

Open the classification band: double-click on band_1

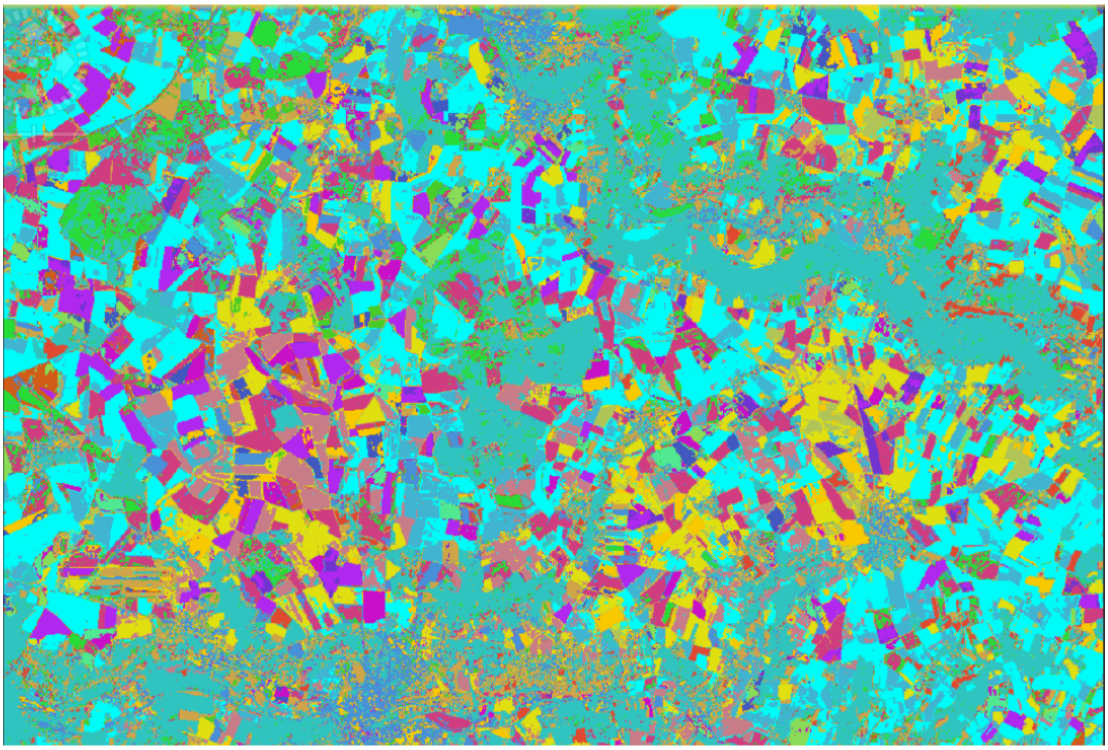


In this band, the classes are represented. To improve the visualization, go to the Colour Manipulation window and import the colour palette in

/path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Auxdata/colour_classification.cpd



If it asks for “Automatically distribute...” select No (it is already well distributed).



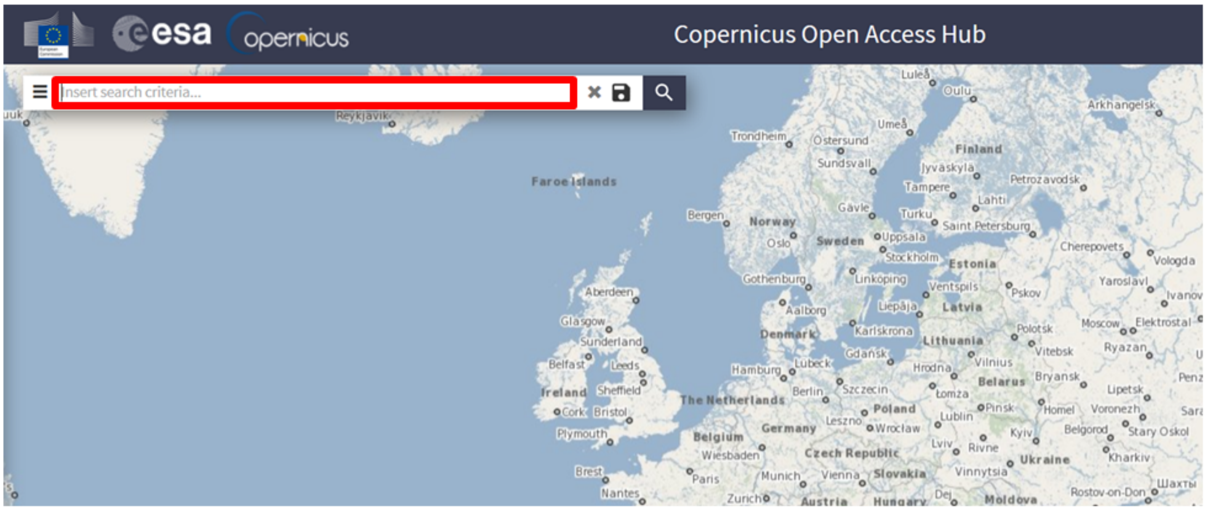
1 Step by step – SAR processing

To carry out the classification with the Sentinel-1 time series, a preprocessing chain is also necessary. In the tutorial, we are going to use a graph for the preprocessing of every S1 image. After that, all acquisitions will be gathered into a stack, filtered and terrain corrected to a ground geometry. Then, a classification process will provide a crop map.

1.1 Data download – ESA SciHUB

In this step, we will download the Sentinel-1 scenes from the Copernicus Open Access Hub using the online interface (**Applications -> Network -> Web Browser,** or click the link below).
Go to https://scihub.copernicus.eu/

By using the searching tool, download the following images:



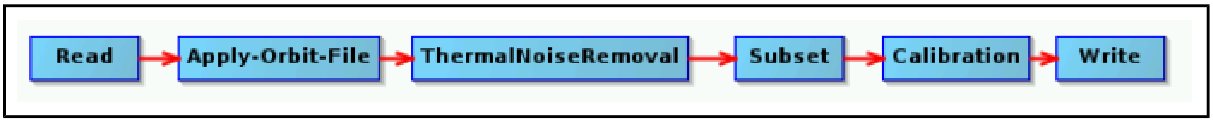
- *S1A_IW_GRDH_1SDV_20160320T174002_20160320T174027_010456_00F839_F4BD.zip*
- *S1A_IW_GRDH_1SDV_20160425T174004_20160425T174029_010981_0107DF_FC53.zip*
- *S1A_IW_GRDH_1SDV_20160519T174005_20160519T174030_011331_01130E_EDE3.zip*
- *S1A_IW_GRDH_1SDV_20160612T174006_20160612T174031_011681_011E3D_78B8.zip*
- *S1A_IW_GRDH_1SDV_20160718T174008_20160718T174033_012206_012F1A_2538.zip*
- *S1A_IW_GRDH_1SDV_20160811T174010_20160811T174035_012556_013AAC_34EA.zip*

Data will be downloaded to /home/rus/Downloads as zip archives. Move the archives to:
/shared/Training/IGA0718_Earthquake_Damage_Detection_Iran_TutorialKit/Original/S1/ .

1.2 SAR Preprocessing

As seen before, the use of a graph allows to concatenate several processing in memory without writing the intermediate files, saving memory and time.

A graph summarizing the SAR preprocessing is given in the folder Scripts/SARPreprocessingGraph.xml



The **Read** module provides an image reader at the beginning of the processing.
The module **Apply-Orbit-File** downloads accurate satellite orbit parameter, allowing a more precise geolocation of image. All the default settings are used.
The **Thermal Noise Removal** module applies a LUT to remove a backscatter gradient (sometimes present between Sentinel-1 sub-swaths). All the default settings are left.
In order to reduce computing time and memory used, a spatial **Subset** is conducted to limit the processing to our Region Of Interest (ROI). A Well-Known Text (WKT) geometry of the the ROI can be given through the geo-coordinates tab.

Finally, the **Calibration** module transforms digital numbers of the sensor into a backscattering coefficient $\sigma°$. This step allows comparing data from different dates and sensors. The default settings are left.
The **Write** module provides an image writer that saves the output image.

The graph can be executed from SNAP Desktop or from command line, as seen before. In this tutorial, we are going to execute it via command line using gpt but through a bash script. This way, it is possible to launch it for all the images in a folder at the same time and it is not needed to execute a different command for each product.

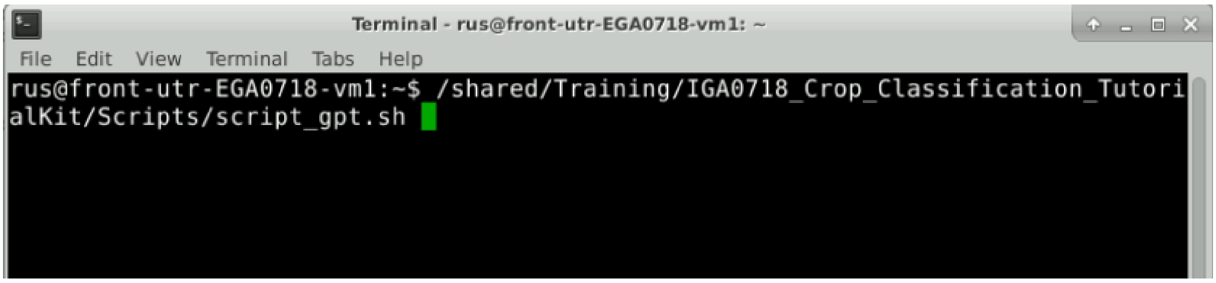
The bash script is in /Scripts/script_gpt.sh

```
#!/bin/bash

oldEnd=.zip
newEnd=_Cal.dim

for i in $(ls -d -1 /shared/Training/IGA0718_Crop_Classification_TutorialKit/Original/S1/S1*.zip)
do
        date
        gpt /shared/Training/IGA0718_Crop_Classification_TutorialKit/Scripts/SARPreprocessingGraph.xml -PinputFile=$i -PoutputFile=${i%$oldEnd}$newEnd
        date
done
```

It runs the graph for all the S1 products in the folder */path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Original/S1* and writes the output to the same folder.

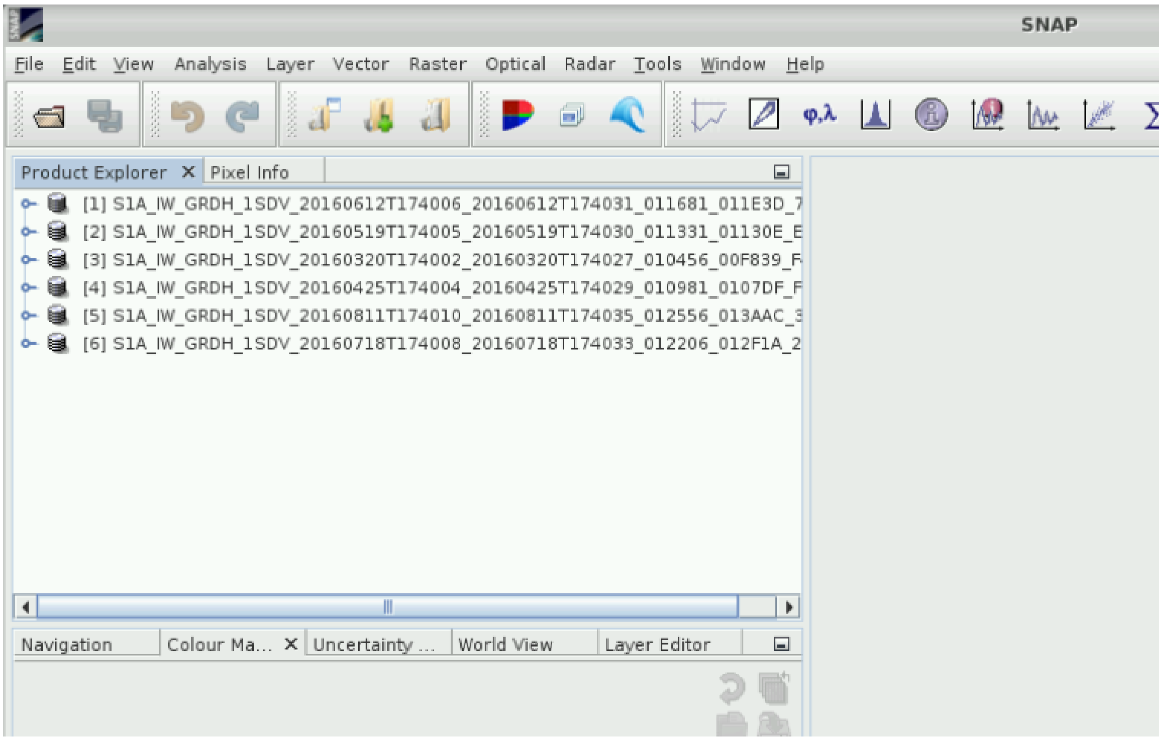
To execute the bash script, open a terminal and execute the command:

```
/path/to/FOSS4G-2019_SNAP&OTB-Crop-Classification_TutorialKit/Scripts/script_gpt.sh
```



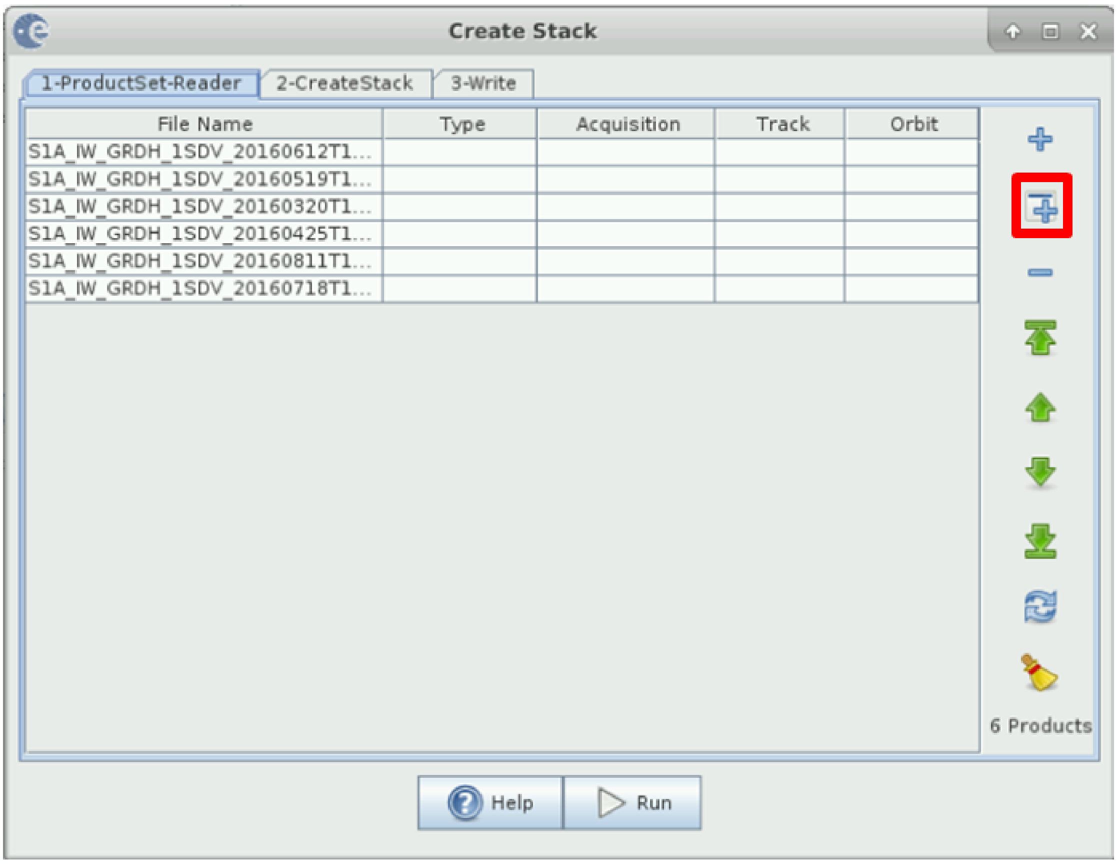
1.3 Create stack

Now that all the input SAR images are preprocessed, open all of them in SNAP.

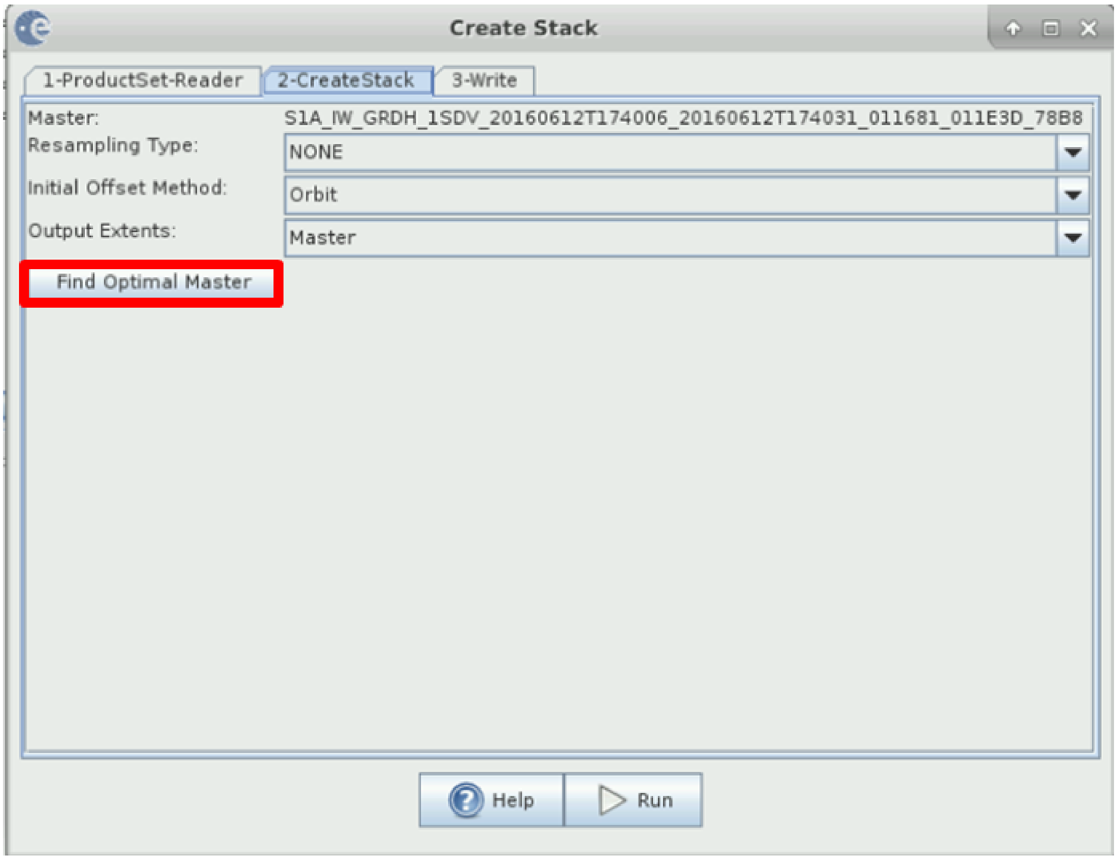


In this step, all SAR images are gathered into a single multiband image by using the Create Stack processor (**Radar -> Coregistration -> Stack Tools -> Create Stack**).

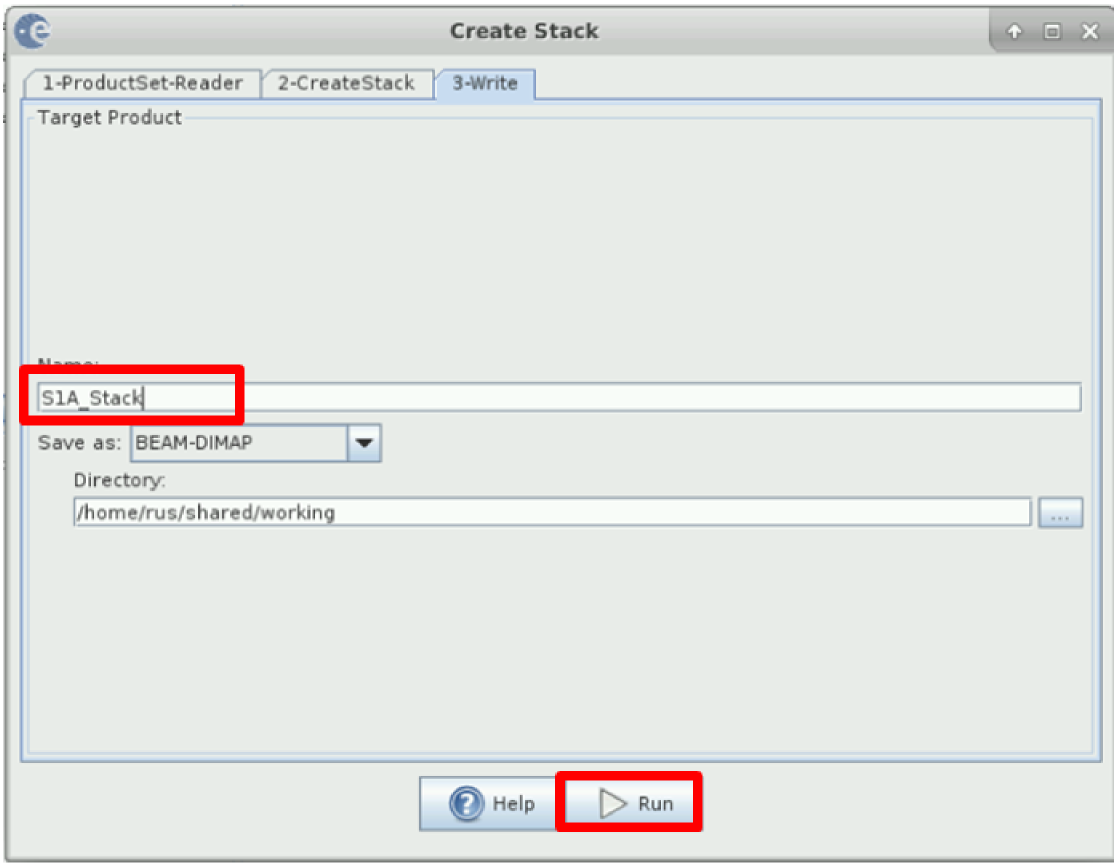
Click on  for adding all the products:



In the **Create Stack** tab, choose the optimal master image by clicking on **Find optimal master**: the acquisition with the baseline closest to all other baselines is selected to be master.



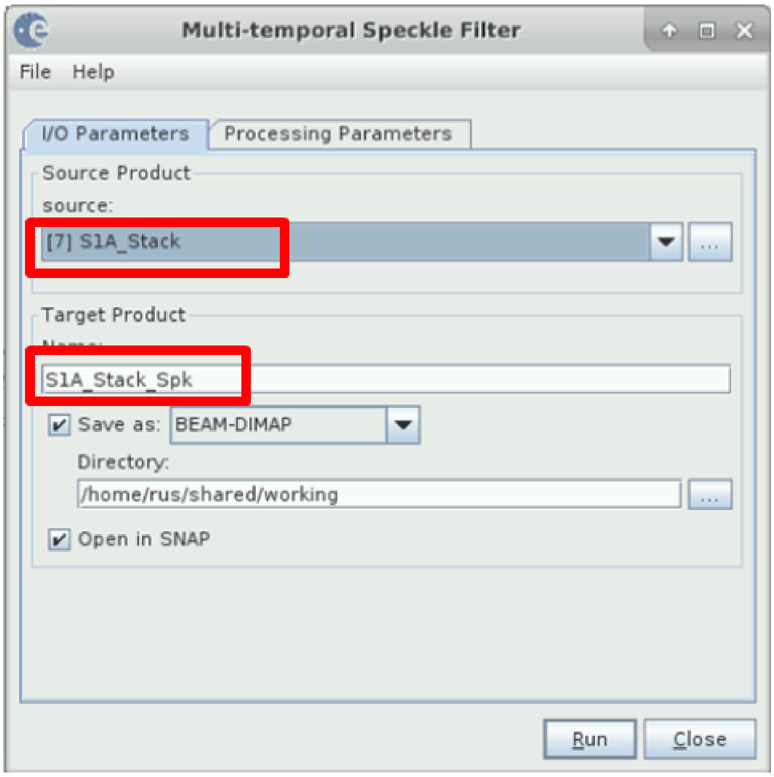
Select the name in the **Write** tab and Run.



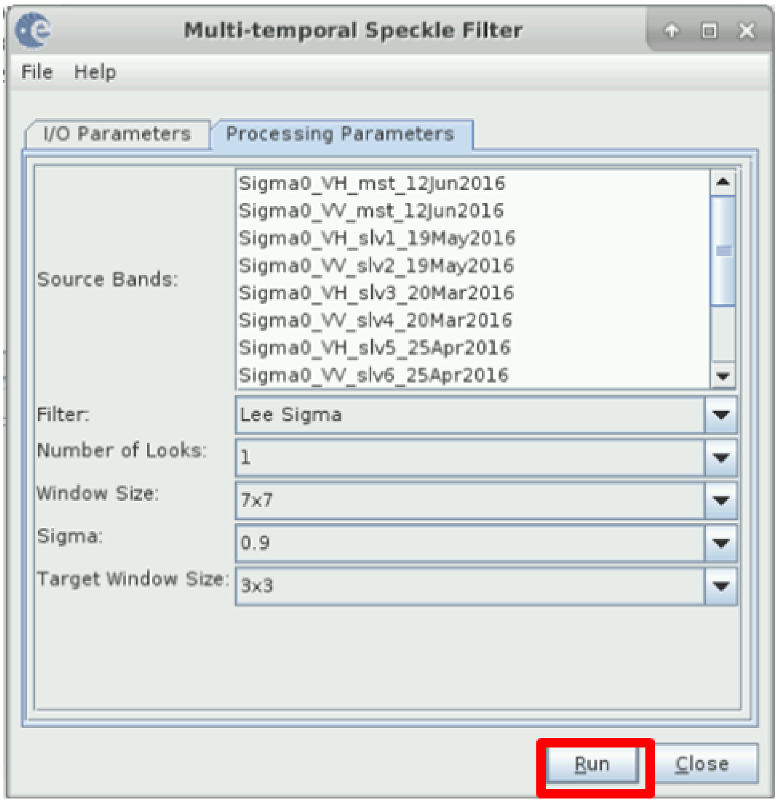
1.4 Speckle filtering

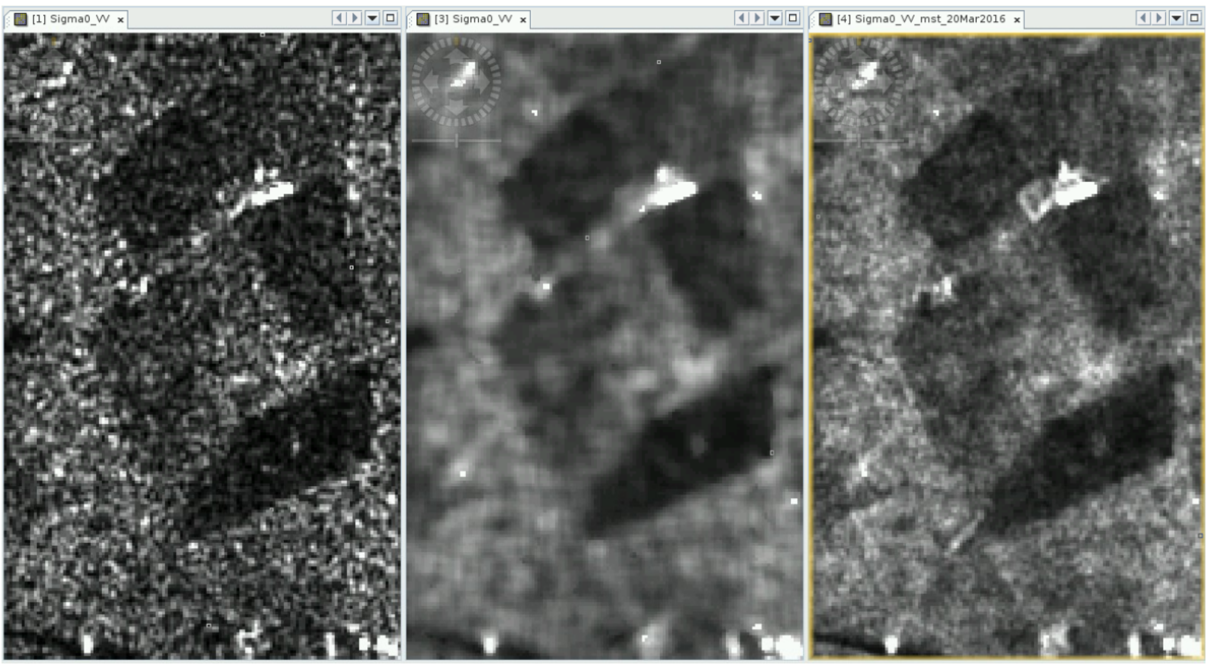
SAR images are affected by the speckle. This phenomenon hinder their use in a classification. In order to smooth the speckle, a filter has to be applied. The filtering procedure mainly consists in averaging pixels, this average can be conducted either in the spatial and the temporal dimension. The **multi-temporal speckle filtering** proposed in SNAP (**Radar->Speckle Filtering->Multi-Temporal Speckle Filtering**) smooths pixels with homogeneous behaviour in space and time and preserves heterogeneous areas and temporal behaviour.
Several adaptive filters can be used in the spatio-temporal filtering process, the Lee Sigma filter is widely used and allow preservation of parcels edges.

Set the S1A_stack as source



Execute the operator (the rest of the parameters can be left by default)





The images above show agricultural plots:

- Left: raw image
- centre: result of Lee Sigma filter applied only in the spatial domain
- right: result of Lee Sigma filter applied in the spatio-temporal domain.

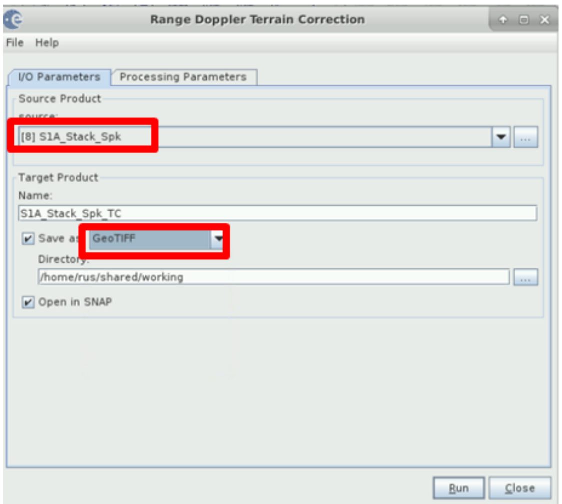
1.5 Terrain correction

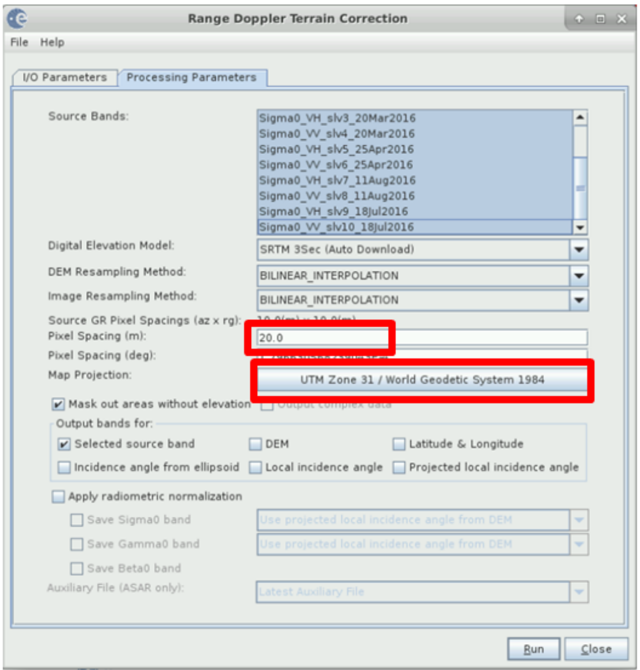
The filtered stack is still in radar geometry: the image orientation corresponds to the sensor azimuth. A projected coordinate system is more convenient to combine the vector, optical and SAR data.

To geocode and correct radar geometry the **Range Doppler Terrain Correction** module can be used (**Radar -> Geometric -> Terrain Correction -> Range Doppler Terrain Correction**).

Configure the module as following:

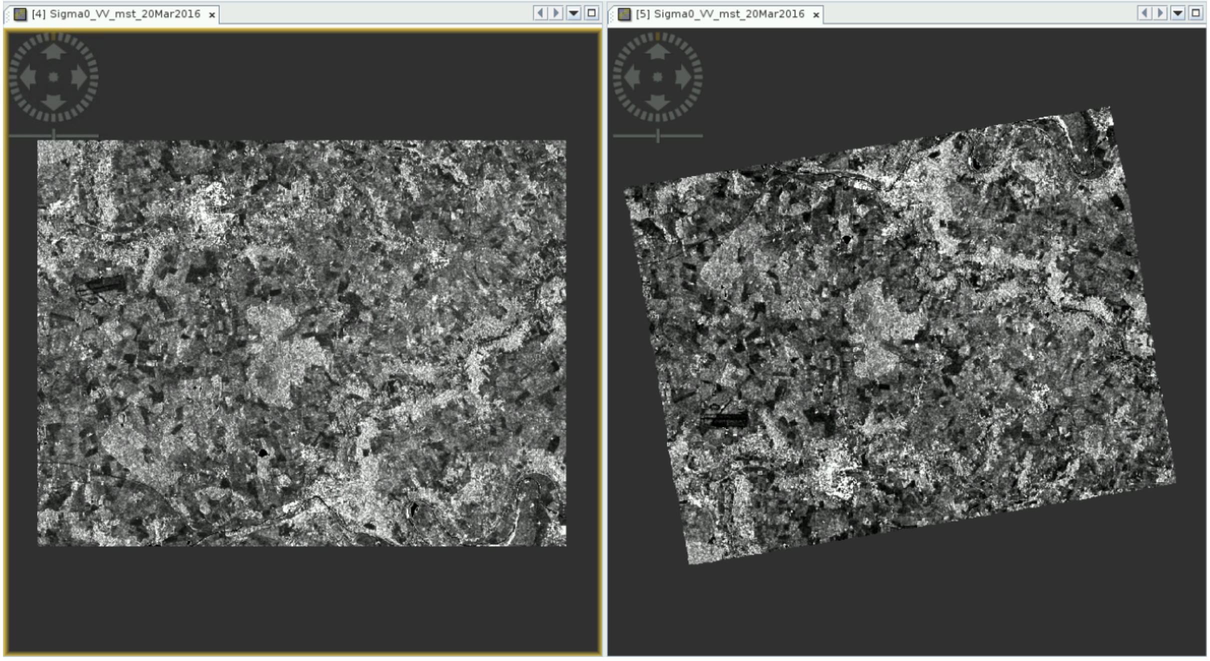
- The pixel spacing is 10m, change this parameter to 20m to have the same size as optical stack.
- Change the Map Projection to the projection UTM / WGS84 (automatic), it should automatically project to UTM 31N zone, the same as optical data.
- Save as GeoTIFF





Run the processor.

The image below shows the difference between SAR geometry (left image) and ground geometry (right). We can notice that up and down image are flipped: in sensor geometry in ascending orbit, the South of the image is acquired first, so it is on top of the image. As the orbit is not exactly oriented along the North-South axis buy with an angle of 98.18°, the geocoded image is not exactly North oriented.



1.6 Classification

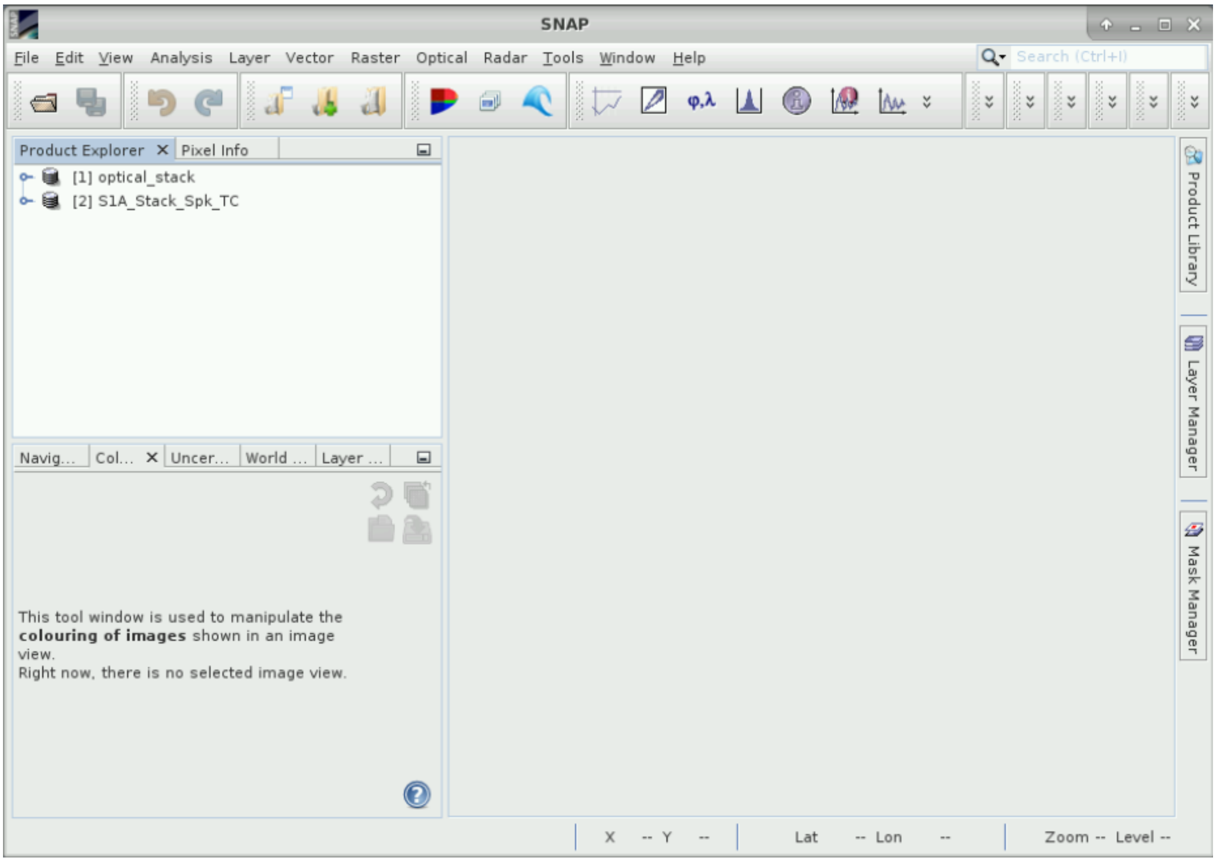
The input for the SAR classification is the S1A_Stack_Spk_TC.tif. The procedure is the same followed in the S2 processing chain. See Classification in OTB and reproduce the steps for obtaining:

- S1A_Stack_Spk_TC_model.txt
- S1A_Stack_Spk_TC_classif.tif
- S1A_Stack_Spk_TC_classif_confidence.tif
- Validation statistics

2 S1-S2 Fusion

2.1 Fusion preparation

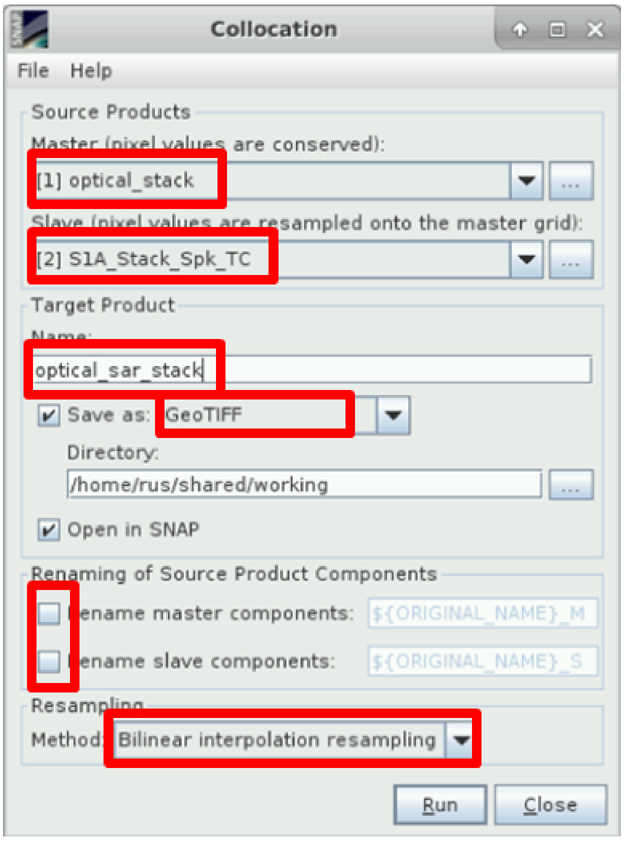
Close all the products in SNAP and open the two geotiff stacks: the optical (optical_stack) and the SAR (S1A_Stack_Spk_TC).



In order to be able to combine the data, they may have exactly the same geometry. For that, we are going to use the Collocation operator in SNAP (**Menu -> Raster -> Geometric Operations -> Collocation**)

Configure the collocation:

- the optical stack as master and the sar stack as slave.
- save the product in GeoTIFF format
- uncheck the renaming options
- Choose bilinear as resampling method



2.2 Feature level fusion

In this step, we are going to carry out a classification taking into account all the optical and sar data at the same time.

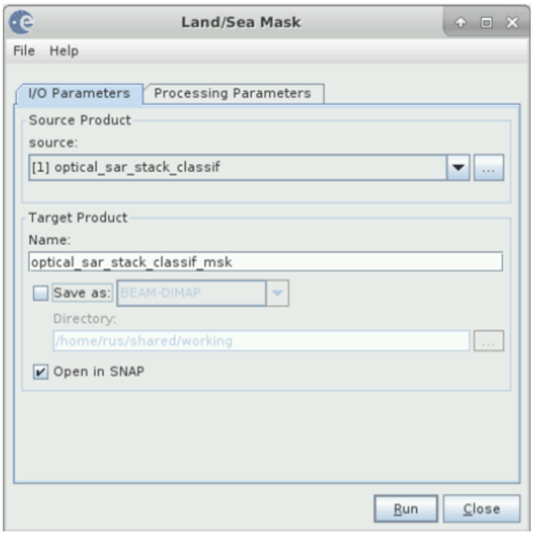
Follow the classification steps in OTB (Classification in OTB) using as input the optical_sar_stack for obtaining:

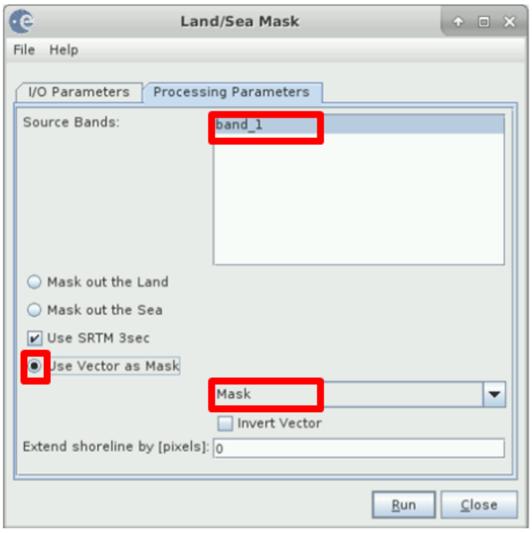
- optical_sar_stack_model.txt
- optical_sar_stack_classif.tif
- optical_sar_stack_classif_confidence.tif
- Validation statistics

Open the optical_sar_stack_classif.tif in SNAP.

Import the vector mask: /Vector_data/PlotsMask/Mask.shp:

- Select the optical_sar_stack_classif product
- **Vector -> Import -> ESRI Shapefile** and select the Mask shapefile
- Apply the mask: **Raster -> Masks -> Land/Sea Mask**

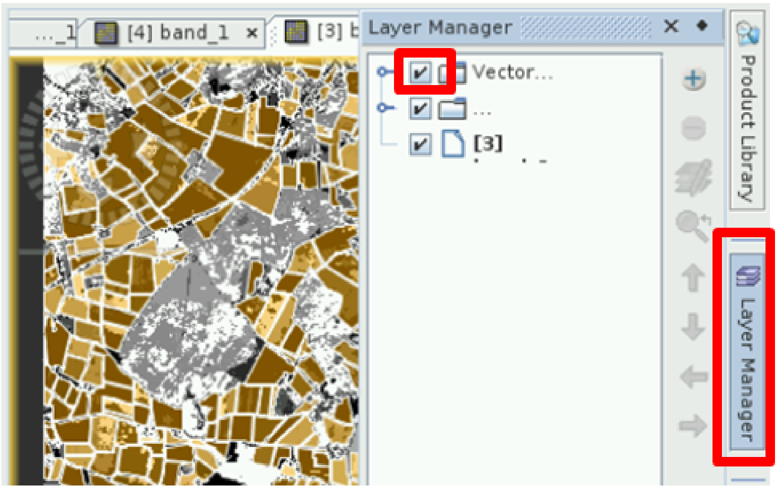


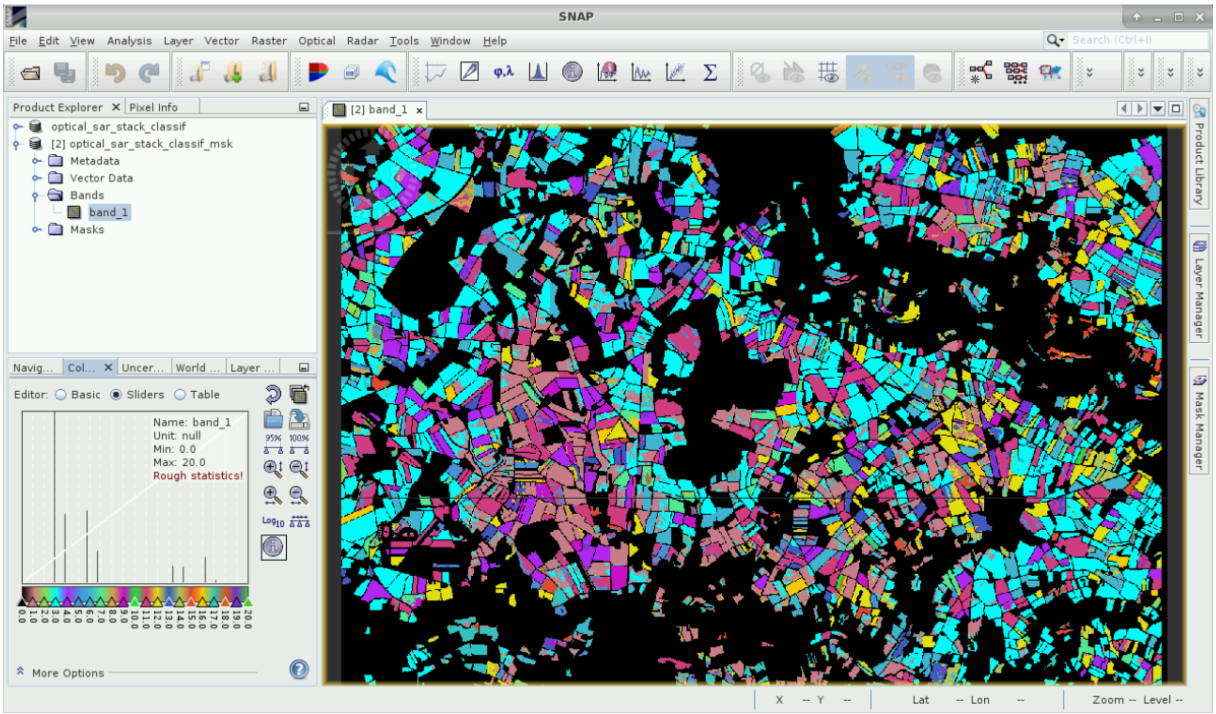


Open the masked band and apply the colour palette "Auxdata/colour_classification.cpd"

TIP 2:

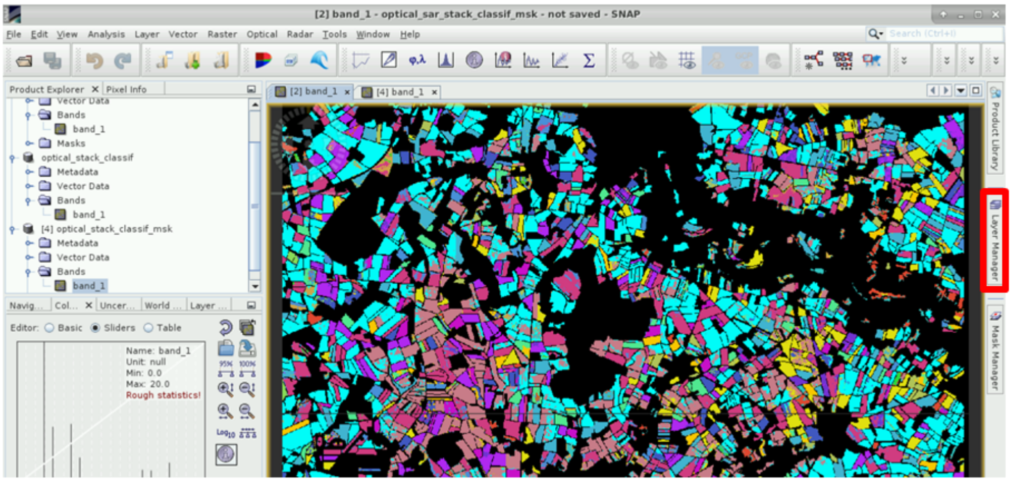
When there are vectors in the product, by default, they are shown when opening a band. It is possible to remove them from the view using the Layer Manager: uncheck the vectors:



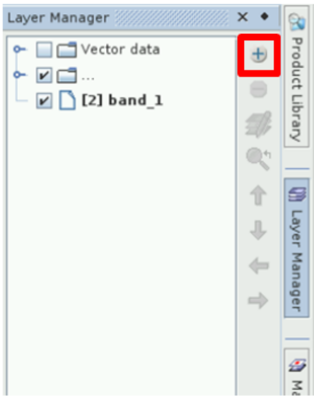


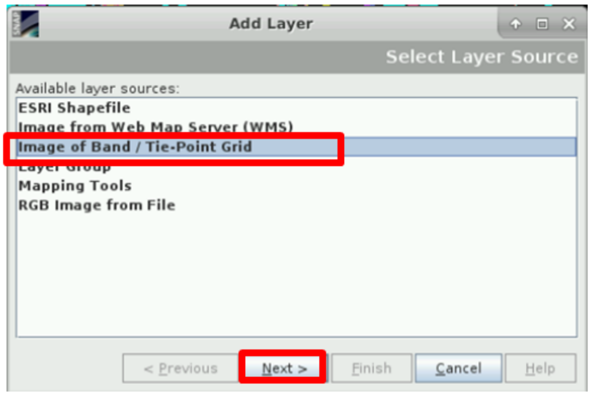
It is possible to compare visually the classification with the optical classification obtained before:

- Open the optical_stack_classification.tif in SNAP
- Import to it the mask vector and apply it as before.
- Apply the same colour palette
- Add the optical classification layer to the fusion classification view using the layer manager:
    - o Go to the view of the processed fusion classification and click on the Layer Manager

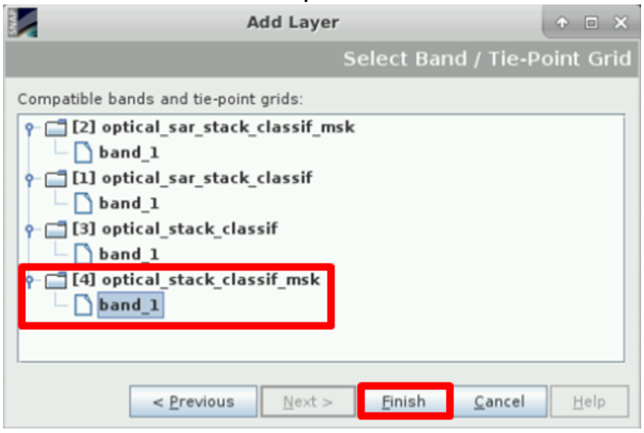


    - o Add Layer -> Image of a band

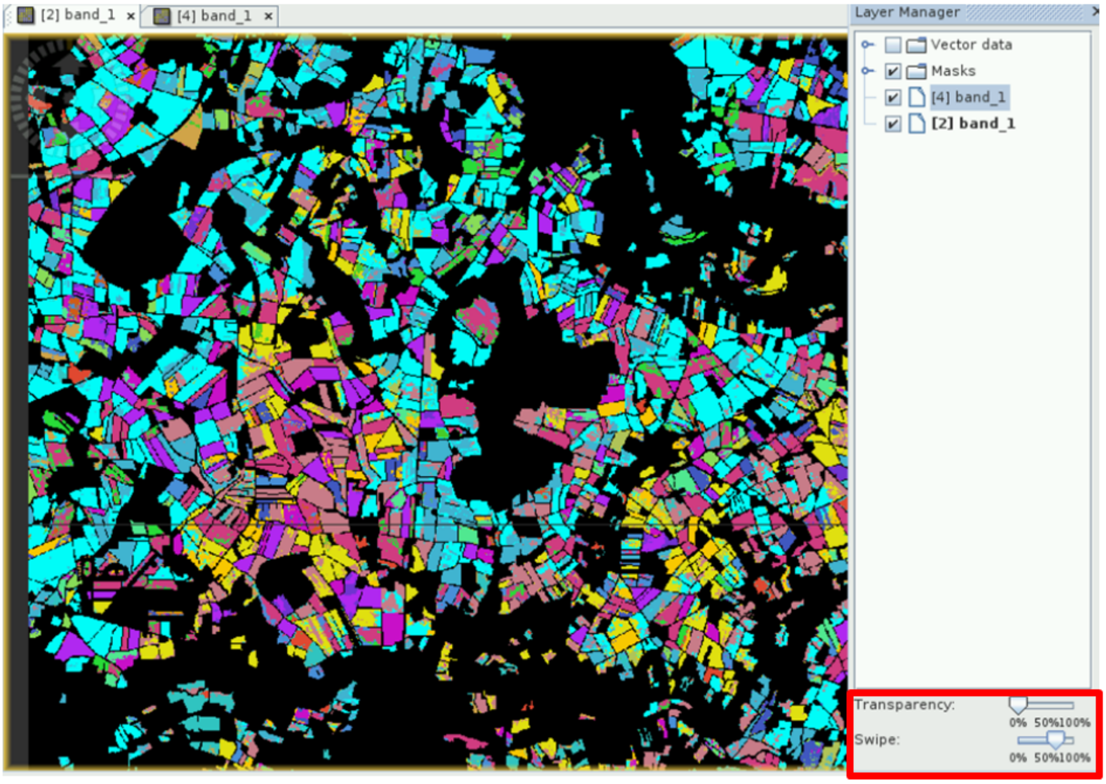




- o Select the masked optical classification



- o Use the transparency and swipe options to see the differences



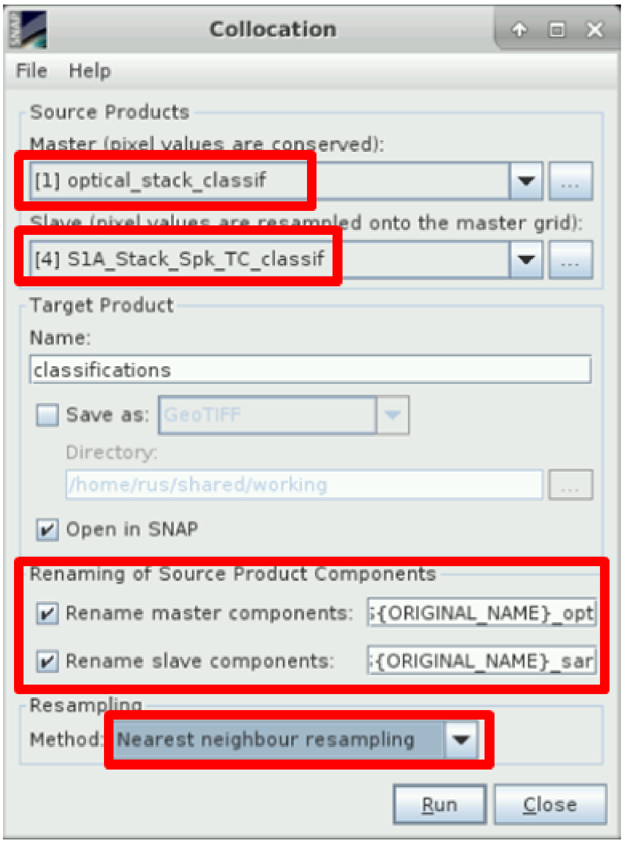
2.3 Decision level

Another kind of fusion that could be used in this case is what is known as “decision level” fusion. The input for this fusion is the optical and sar classification instead of the reflectance and sigma bands.
When we made the classification, we obtained also an output that was the confidence. We can use this information to create a new classification where we select for every pixel the classification with more confidence from those available.

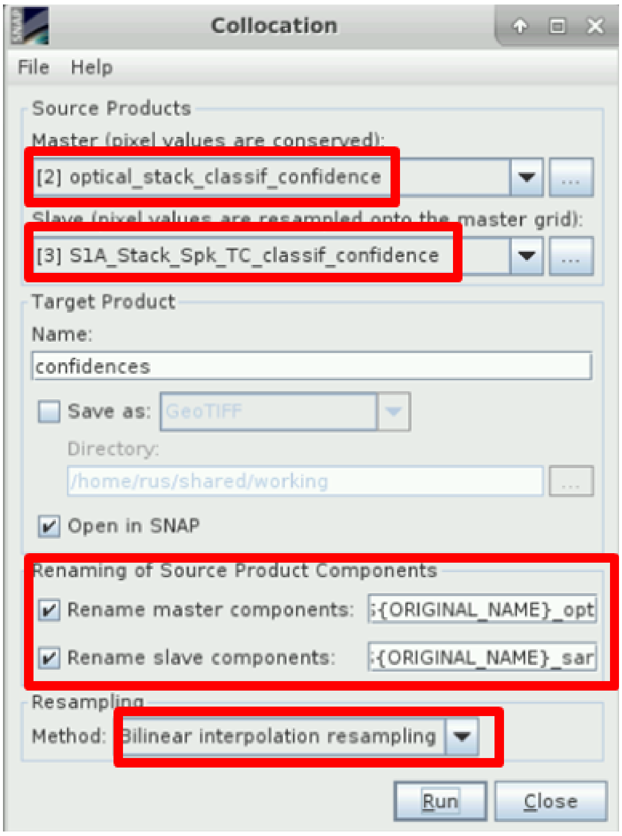
The steps are:

- Open the classification and confidence images obtained in the optical and in the sar processing chains.

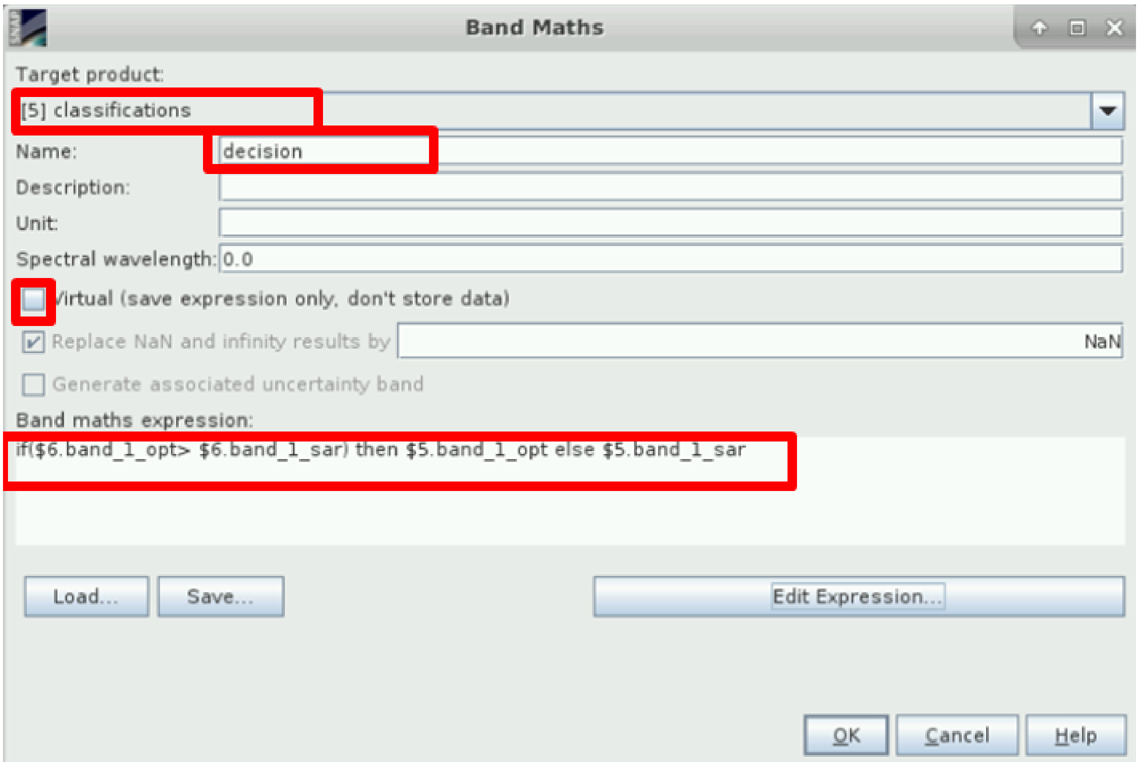
- Collocate (Raster -> Geometric Operations -> Collocation) the sar classification (slave) with the optical classification (master) using Nearest neighbour method.



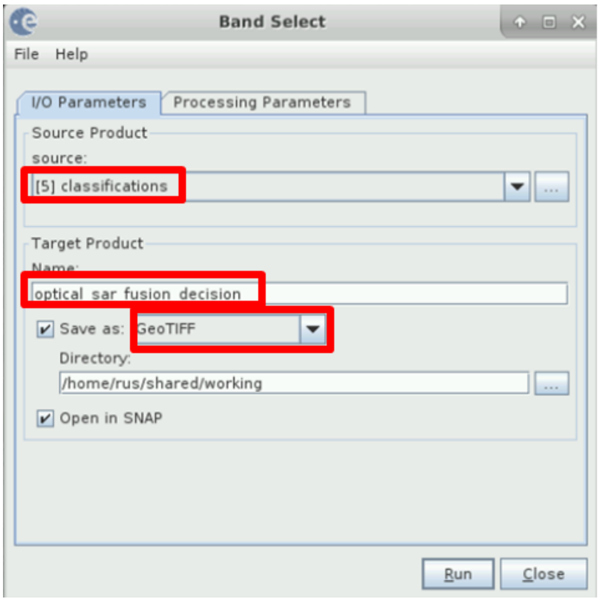
- Collocate (Raster -> Geometric Operations -> Collocation) the sar confidence (slave) with the optical confidence (master) using bilinear as resampling method.

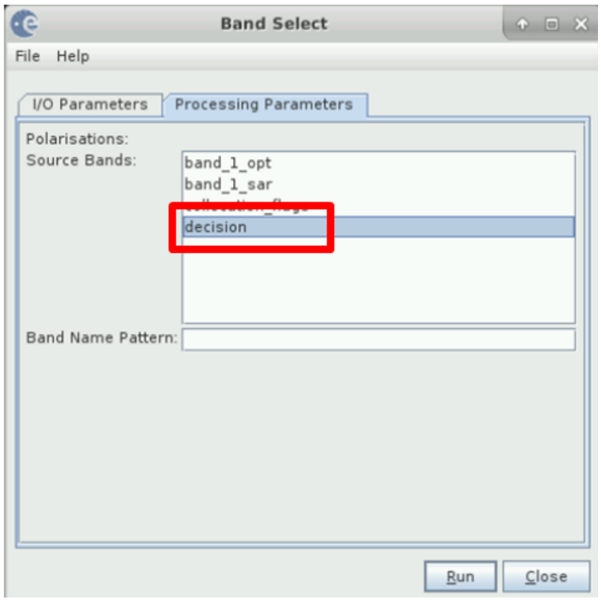


- Use Band Maths on the classification products to define a new band with the expression:

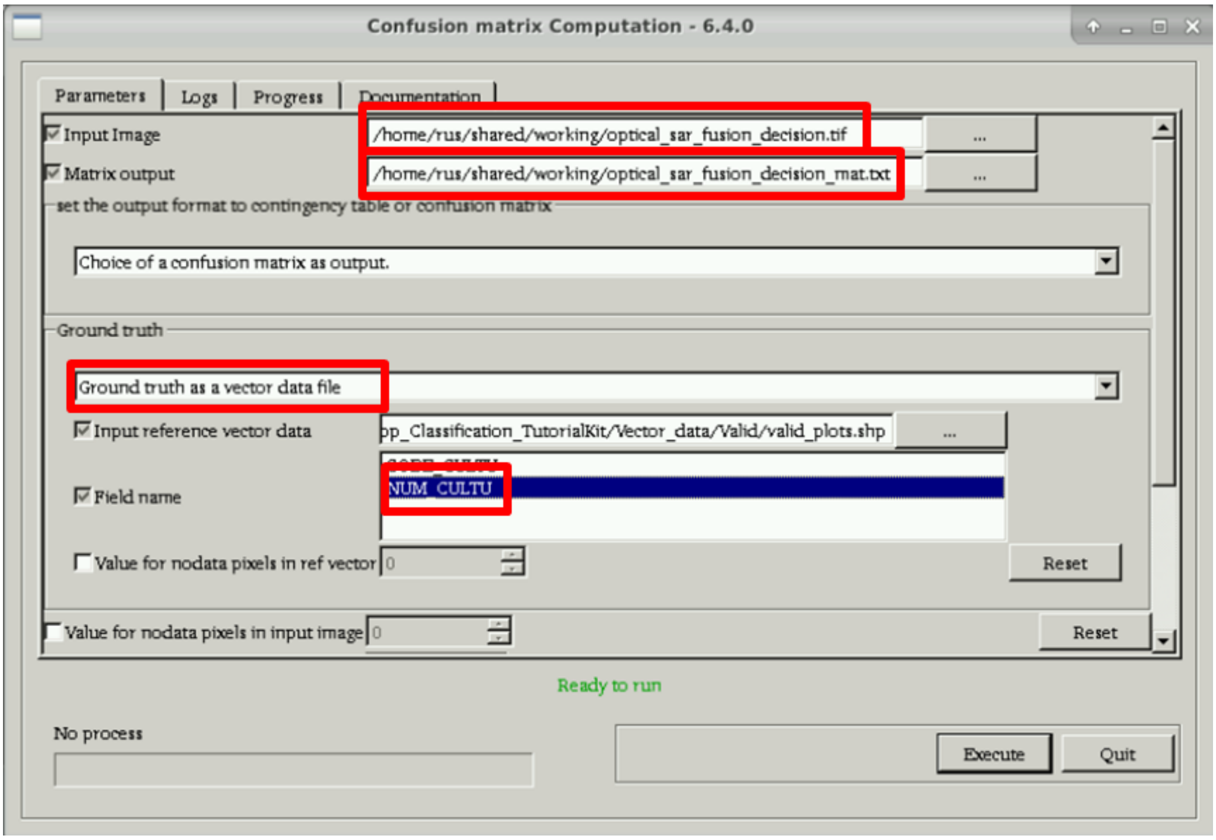


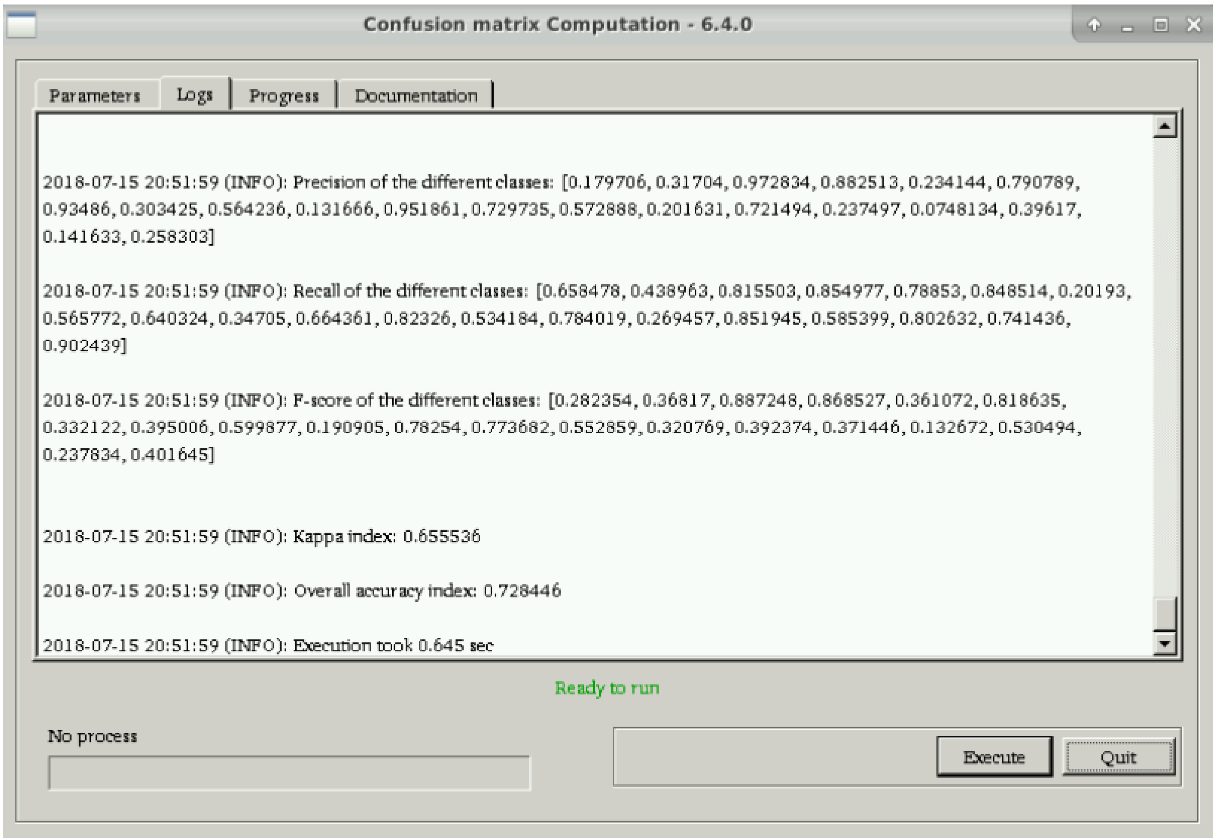
- Export this new classification to a GeoTIFF format by using the Band Select operator (Raster -> Data Conversion -> Band Select)





- Check the statistics of the 'decision' classification in OTB with ComputeConfusionMatrix application.





3 Extra steps

Some open questions/ideas to continue the analysis:

- Fusion of several classifiers:

We could carry out the study with more classifiers than Random Forest and with different configurations and use all of them as input for the “decision level” fusion. See the **FusionOfClassification** application in OTB, for example.

- Deep analysis:

With the available information in the confusion matrix, it could be done an analysis about what kinds of crops are classified better, which are not.

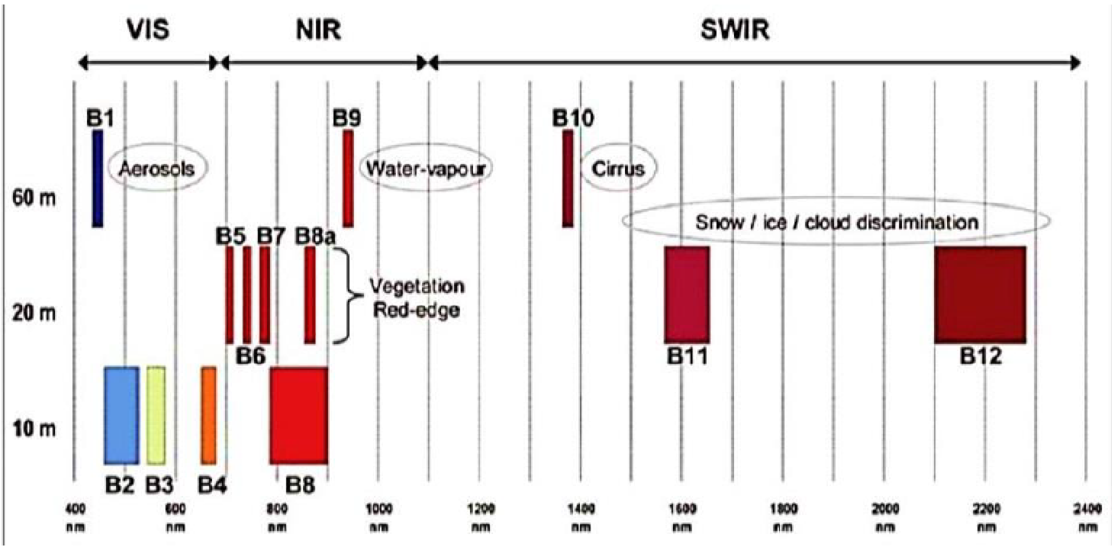
- Analyse the influence of the selected features in the classification

What happen if we use only the reflectance bands in the optical classification? Is it worth computing the LAI and the NDVI? Does the classification improve if we add more vegetation indices?
For SAR: does it help if we use some extra band, for example the VV/VH ratio?

4 Annex

4.1 Resampling in SNAP

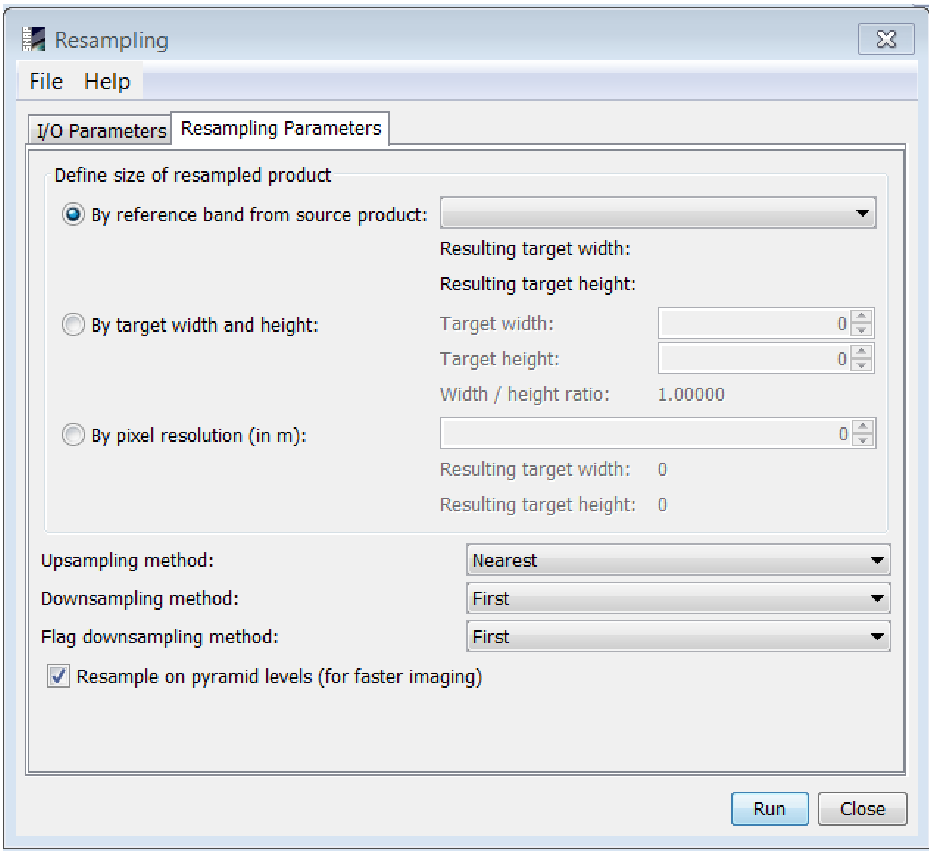
This operator resamples a multi-size source product to a single-size target product.
This is usually needed when working with Sentinel-2 products since they have bands at different resolution: 10, 20 and 60m. (and 5000 m.)



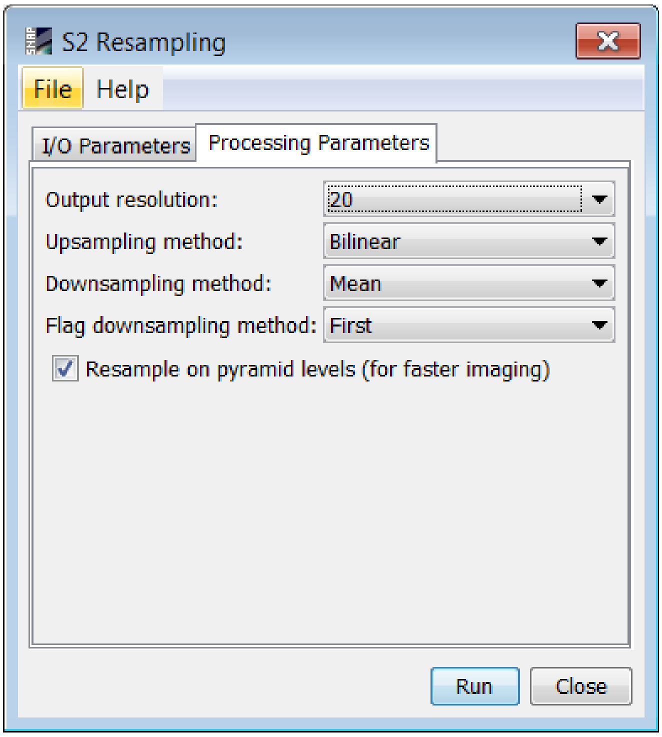
Credits: ESA 2015

Some operators in SNAP require this previous step: for example, it is not allowed to apply a band math operation by using bands of different size.
There are two different operators in SNAP for resampling the Sentinel-2 images:

- **Menu -> Raster -> Geometric Operations -> Resampling** : the generic one

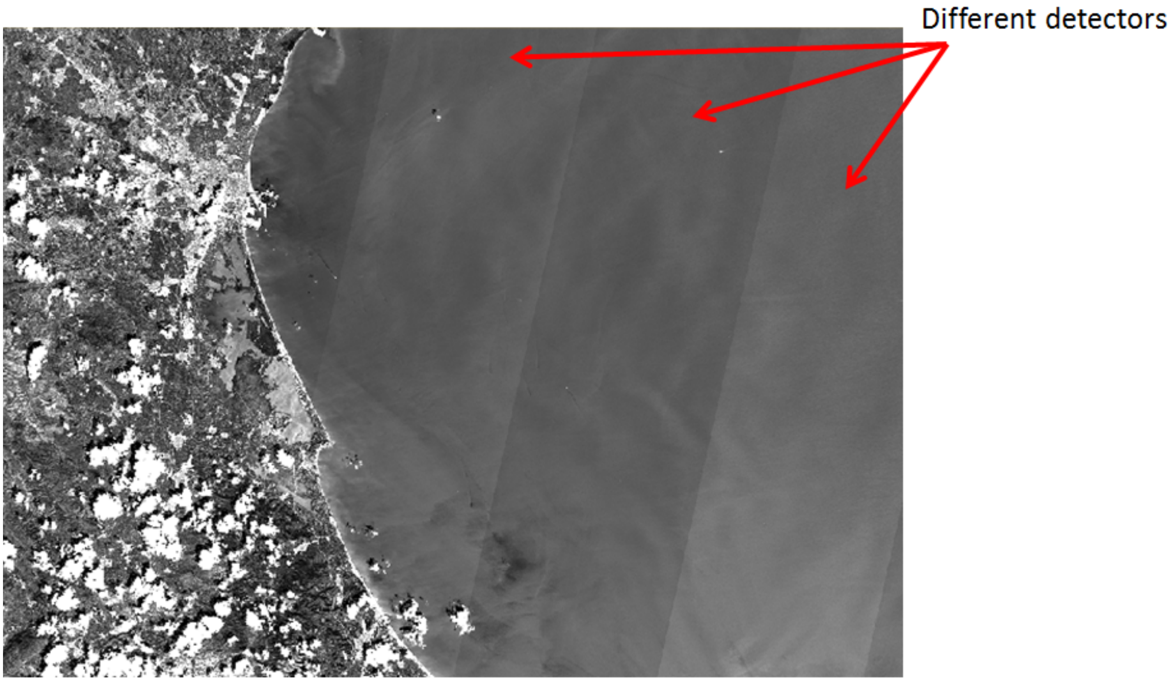


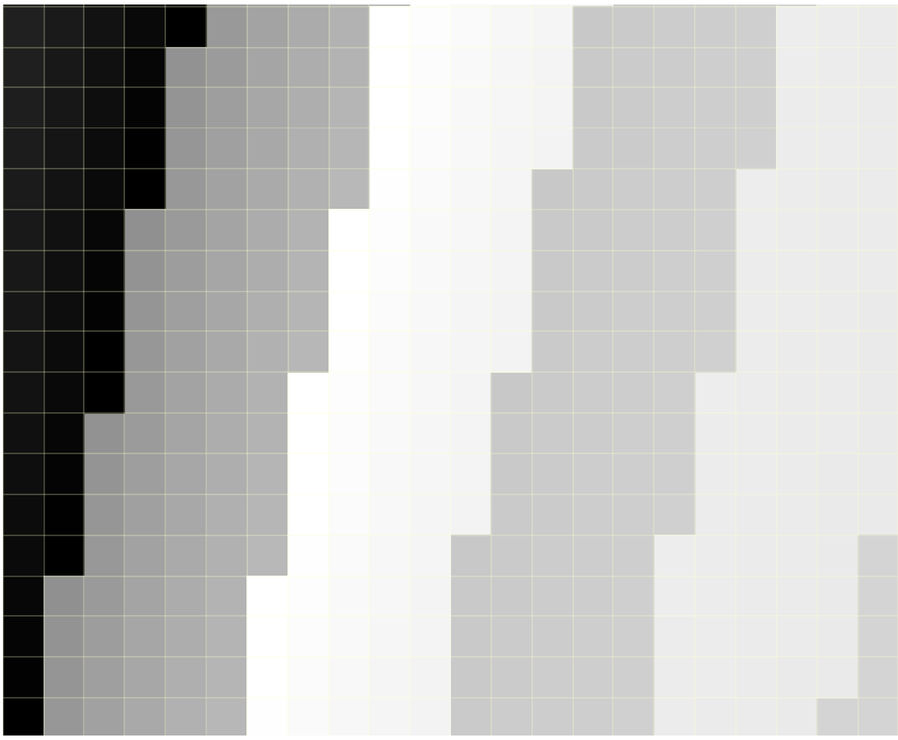
- **Menu -> Optical -> Geometric -> S2 Resampling Processor:** a resampling method specific for Sentinel-2 images.



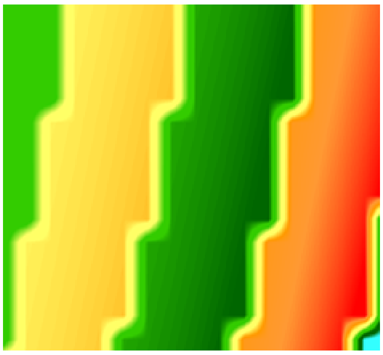
The difference between them is basically how the angles bands are managed: the S2 specific operator is able to read the detector footprint in order to interpolate the angles properly depending on from which detector is taken the pixel.

The S2 sensor, the Multi-Spectral Imager (MSI), is composed by 12 detectors with different orientation angles, which causes discontinuities in the viewing angles in an image (clearly visible over the sea):

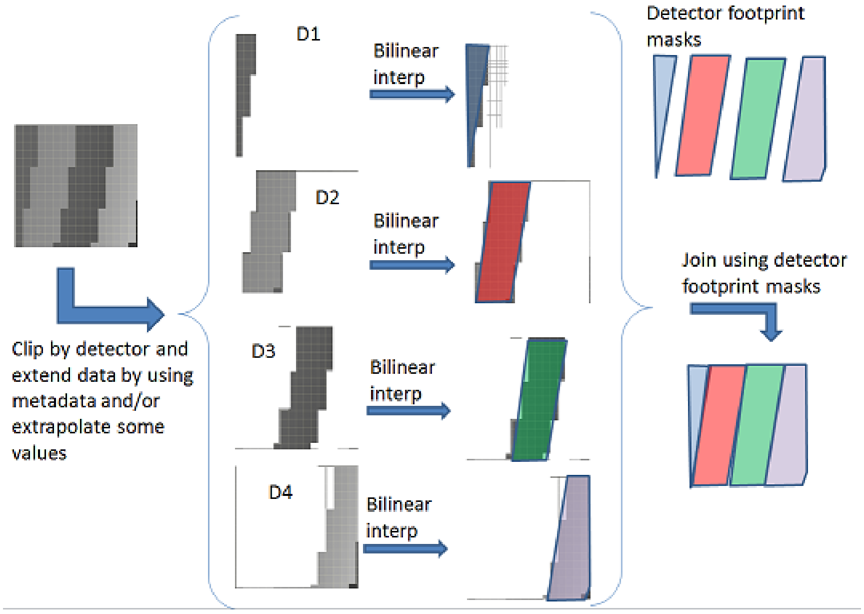


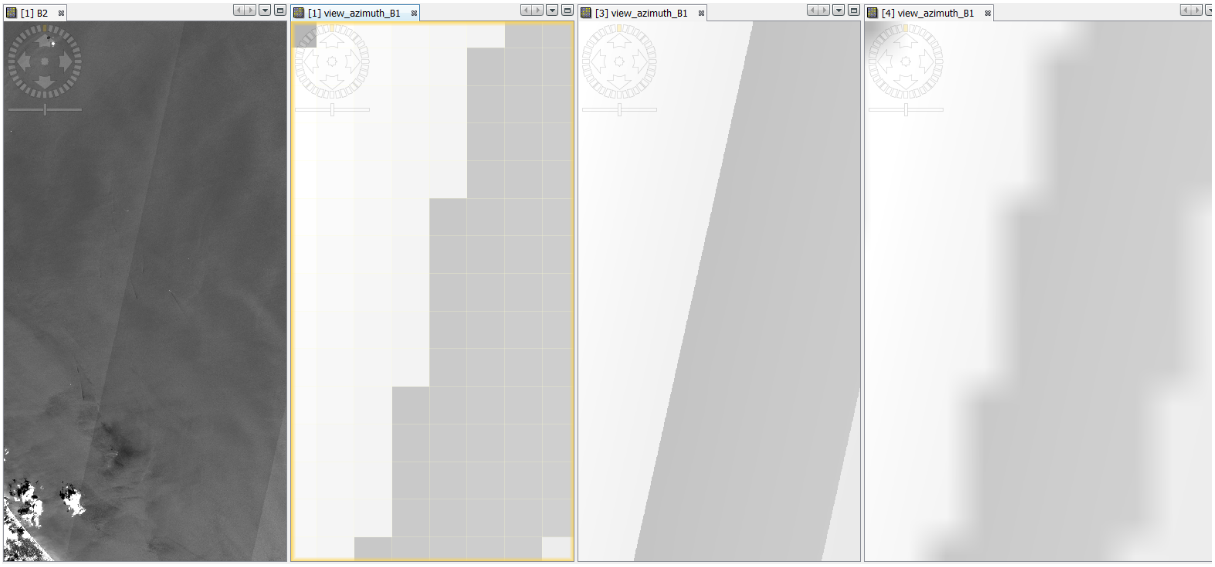

Azimuth viewing angles of B2.

Because of the discontinuity of the angles when the detector changes, the result of the resampling of the azimuth angle bands is not good from the physics point of view, since it blurs the border between detectors when we use a bilinear method. It is needed an specific processor to carry out the resampling of these bands in a "special" way.



The specific operator uses not only the angles bands but also the detector footprints. By using the footprints, it is possible to "isolate" the angles of each detector and interpolate them without using the angle information of the adjacent detectors (which causes the blurring). In the pixels closest to the border detector-detector, it is necessary to extrapolate the data since not all data is available to carry out the interpolation with bilinear method. Finally, the angles are joined again.





Left: Band B2, centre-left: original angles, centre-right: angles with S2Resampling, right: angles with the generic bilinear resampling.

4.2 Crop table

Crop code	Crop type	NUM_CULTU	Area (ha)
AVH	Winter oats	1	99.31
AVP	Spring oats	2	73.00
BTH	Winter sweet wheat	3	56.17
BTN	Non-fodder beet	4	74.13
CHV	Hemp	5	73.94
CZH	Rape	6	65.71
FVL	Bean planted before 31/05	7	84.21
J6S	Fallow 6 years or older	8	35.45
LIF	Linen fibers	9	86.35
MIE	Silage corn	10	58.54
MIS	Corn	11	147.59
ORH	Winter barley	12	79.77
ORP	Spring barley	13	63.21
PHI	Winter peas	14	63.22

PPH	Permanent meadow - predominant grass	15	41.94
PPR	Spring pea sown before 31/05	16	97.6
PRL	Long rotation meadow (6 years old or more)	17	46.03
PTF	Starch potato	18	97.46
SOJ	Soy	19	35.84
TRE	Other clover	20	40.77

LINKS:

- SNAP:
  - Main site: https://step.esa.int/main/
  - Forum: https://forum.step.esa.int/
  - Sen2Cor: https://step.esa.int/main/third-party-plugins-2/sen2cor/
- OTB:
  - Main site: https://www.orfeo-toolbox.org/
  - Forum:
- CS contact: otb at c minus s dot fr