Ciências
ULisboa

Faculdade
de Ciências
da Universidade
de Lisboa

DEGGE

MULTISPECTRAL REMOTE SENSING

MASTER IN GEOSPATIAL ENGINEERING

2024/2025

FREE AND OPEN-SOURCE SOFTWARE TO BE INSTALLED

Orfeo ToolBox (OTB) **Version 9.1.0**	Orfeo ToolBox (OTB) version 9.1.0, released on 2024-09-27, is an open-source project for state-of-the-art remote sensing (https://www.orfeo-toolbox.org/). 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! (https://www.orfeo-toolbox.org/CookBook/QGISInterface.html#plugin-installation) All OTB's algorithms are accessible from QGIS, Python, the command line, or C++. QGIS is an easy-to-use visualization tool emphasizing hardware-accelerated rendering for high-resolution imagery (optical and SAR). With it, end-users can visualize huge raw imagery products and access all the toolbox applications. From resource-limited laptops to high-performance clusters, OTB is available on Linux, Windows, and macOS (via Docker for ARM Mac computers). It is community-driven, extensible, and heavily documented. Orfeo ToolBox is not a black box! (https://www.orfeo-toolbox.org/CookBook/)
QGIS **Version 3.40.3**	QGIS is a professional GIS application built on top of and proud to be itself Free and Open Source Software (FOSS). The current version is QGIS 3.40.3 'Bratislava' and was released on 2025-01-17 (https://qgis.org/download/). QGIS is available on Windows, macOS, Linux, and Android. QGIS has a lot of documentation. All documentation is in English, but documents such as the user guide are also available in other languages (https://qgis.org/resources/hub/).
SNAP **Version 11.0.0**	SeNtinel's Application Platform (SNAP) is an open-source common architecture for ESA Toolboxes ideal for Earth Observation data exploitation (https://step.esa.int/main/download/snap-download/). SNAP reunites all Sentinel Toolboxes to offer the most complex platform for this mission. The current version is 11.0.0 released on 2024-10-24.

PRACTICAL LESSON 1

EXERCISE 1.1

Free and open access for both Landsat and Sentinel data products is available to all users through the following links, respectively:

USGS WEBSITE

The USGS Earth Explorer data portal is your one stop shop for obtaining geo-spatial datasets from our extensive collections. Users can navigate via interactive map or text search to obtain Landsat satellite imagery, Radar data, UAS data, digital line graphs, digital elevation model data, aerial photos, Sentinel satellite data, some commercial satellite imagery including IKONOS and OrbView3, land cover data, digital map data from the National Map, and many other datasets. Users can search by exact location via the interactive map or input specific coordinates to view what data types are available.

https://www.usgs.gov/land-resources/nli/landsat/landsat-data-access

https://earthexplorer.usgs.gov/

CONVENTIONAL DATA ACCESS HUBS

One of the access points to a wide range of Earth observation data from the Copernicus Sentinel missions managed by ESA is the **Copernicus Data Space Ecosystem** that provides tools for easy discovery, visualization, and download which will be continuously upgraded.

https://www.copernicus.eu/en/access-data/conventional-data-access-hubs

https://dataspace.copernicus.eu/

DOWNLOAD THE DATA

Search for the following two Sentinel-2 Level-2A data products ID in the ONDA platform at: https://catalogue.onda-dias.eu/catalogue/ (login required, registration is free) and download them:

- S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433
- S2B_MSIL2A_20210817T112119_N0301_R037_T29SMC_20210817T130929

Once this process takes a while, download both images from the OneDrive platform (**folder DRM2024**).

The Level-2A processing includes a scene classification and an atmospheric correction applied to Top-Of-Atmosphere (TOA) Level-1C orthoimage products. Level-2A main output is an orthoimage Bottom-Of-Atmosphere (BOA) corrected reflectance product.

Additional outputs are an Aerosol Optical Thickness (AOT) map, a Water Vapour (WV) map, and a Scene Classification Map (SCM) together with Quality Indicators (QI) for cloud and snow probabilities at 60 m resolution. Level-2A output image products will be resampled and generated with an equal spatial resolution for all bands (10 m, 20 m or 60 m). Standard distributed products contain the envelope of all resolutions in three distinct folders:

10 m: containing spectral bands 2, 3, 4, 8, a True Colour Image (TCI) and an AOT and WV maps resampled from 20 m.

20 m: containing spectral bands 2 - 7, the bands 8A, 11 and 12, a True Colour Image (TCI), a Scene Classification map (SCL) and an AOT and WV map. The band B8 is omitted as B8A provides more precise spectral information.

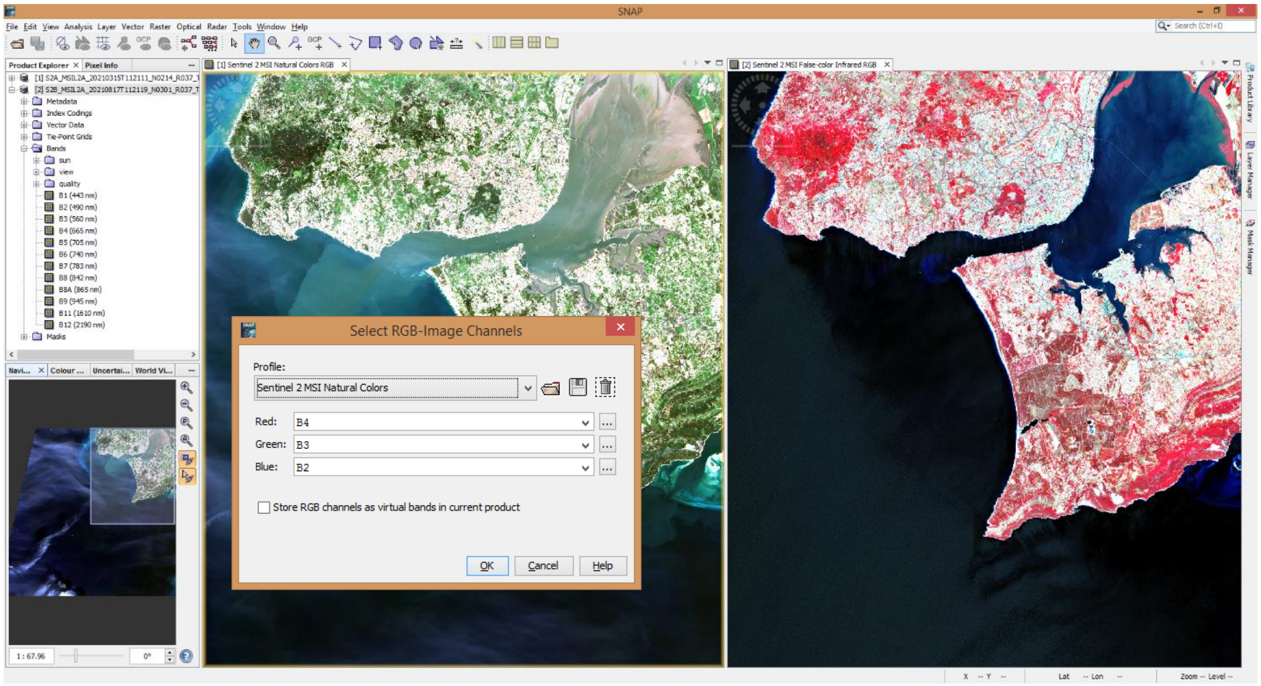
60 m: containing all components of the 20 m product resampled to 60 m and additionally the bands 1 and 9. ***The cirrus band 10 is omitted, as it does not contain surface information.***

SENTINEL-2 **2A launched June 23, 2015** **2B launched March 7, 2017**	**Bands**	**Central Wavelength (micrometers)**	**Resolution (meters)**
	Band 1 - Coastal Blue	0.443	60
	Band 2 - Blue	0.490	10
	Band 3 - Green	0.560	10
	Band 4 - Red	0.665	10
	Band 5 - Red Edge	0.705	20
	Band 6 - Red Edge	0.740	20
	Band 7 - Red Edge	0.783	20
	Band 8 - NIR	0.842	10
	Band 8A - Red Edge	0.865	20
	Band 9 - Water Vapor	0.945	60
	Band 10 – SWIR (Cirrus)	1.375	60
	Band 11 - SWIR	1.610	20
	Band 12 - SWIR	2.190	20

OPEN THE PRODUCTS

In SNAP, use the Open Product button in the top toolbar (or **File > Open Product …**) and browse for the location of the downloaded data. Select the two downloaded zip files and press Open Product.

Select an image and, using the right mouse button, click on “Open RGB Image Window”, and choose one of the available color composites: Natural Colors or False-color Infrared.



Select RGB-Image Channels

Profile: Sentinel 2 MSI Natural Colors

Red: B4

Green: B3

Blue: B2

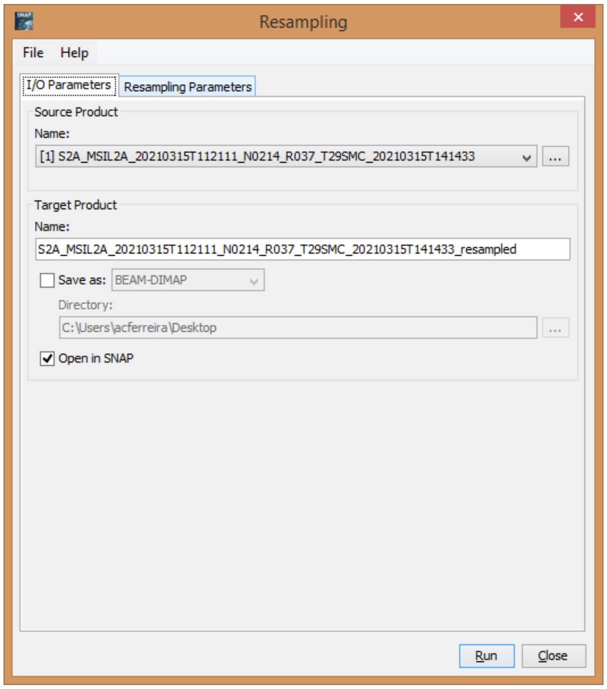
☐ Store RGB channels as virtual bands in current product

EXERCISE 1.2

Pre-processing of satellite images might include some basic image operations, such as image resampling, extraction of a region of interest (ROI), band arithmetic operations, multispectral band merging, and multitemporal images collocation.

RESAMPLE THE PRODUCTS

In SNAP, use **Raster > Geometric > Resampling** to resample a multi-size product to a single-size product. A multi-size product is a product in which bands are of different sizes and/or resolutions. This can be useful for those instances when a SNAP feature is not supported for a multi-size product.



Source Product

Name: [1] S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433

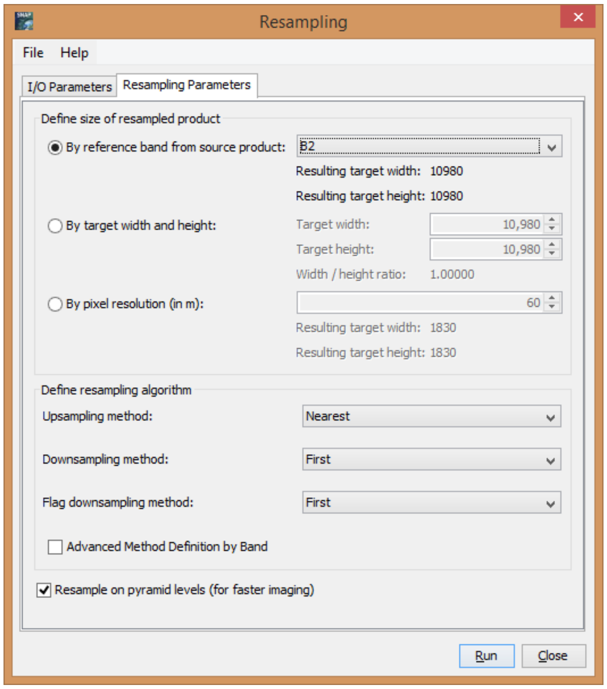
Target Product

Name: S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resampled

□ Save as: BEAM-DIMAP

Directory: C:\Users\acferreira\Desktop

☑ Open in SNAP



Define size of resampled product

● By reference band from source product: B2

Resulting target width: 10980

Resulting target height: 10980

○ By target width and height:

Target width: 10,980

Target height: 10,980

Width / height ratio: 1.00000

○ By pixel resolution (in m): 60

Resulting target width: 1830

Resulting target height: 1830

Define resampling algorithm

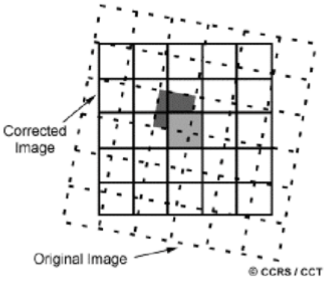
Upsampling method: Nearest

Downsampling method: First

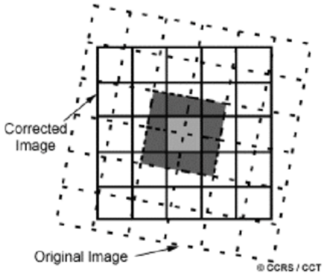
Flag downsampling method: First

□ Advanced Method Definition by Band

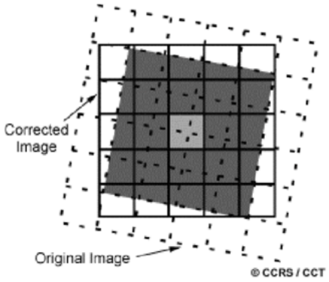
☑ Resample on pyramid levels (for faster imaging)



nearest neighbor method



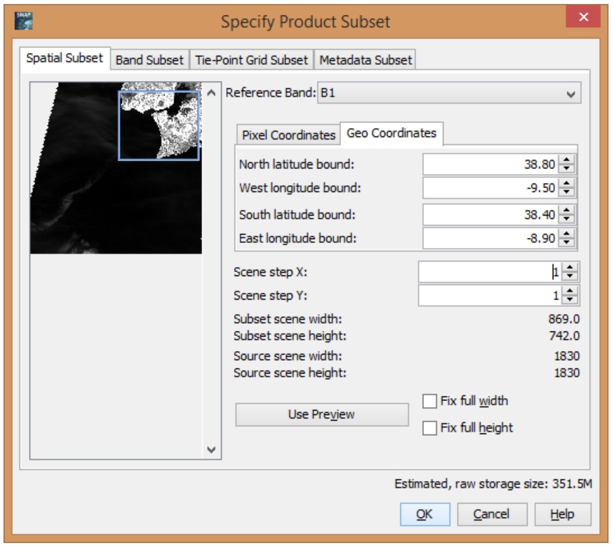
bilinear interpolation method



cubic convolution method

CREATE A SUBSET

In SNAP, use **Raster > Subset** to create either spatial and/or spectral subsets of a data product. The spatial subset may be given by pixel positions or a geographical polygon.

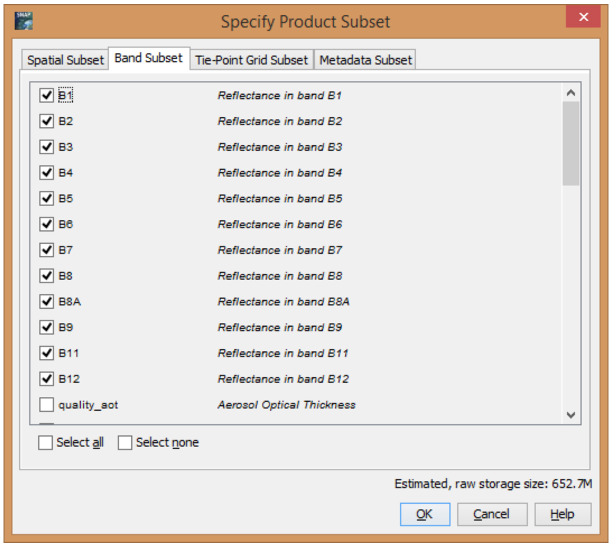


Reference Band: B1

North latitude bound:	38.80
West longitude bound:	-9.50
South latitude bound:	38.40
East longitude bound:	-8.90
Scene step X:	1
Scene step Y:	1
Subset scene width:	869.0
Subset scene height:	742.0
Source scene width:	1830
Source scene height:	1830

☐ Fix full width
☐ Fix full height

Estimated, raw storage size: 351.5M



☑ B1	Reflectance in band B1
☑ B2	Reflectance in band B2
☑ B3	Reflectance in band B3
☑ B4	Reflectance in band B4
☑ B5	Reflectance in band B5
☑ B6	Reflectance in band B6
☑ B7	Reflectance in band B7
☑ B8	Reflectance in band B8
☑ B8A	Reflectance in band B8A
☑ B9	Reflectance in band B9
☑ B11	Reflectance in band B11
☑ B12	Reflectance in band B12
☐ quality_aot	Aerosol Optical Thickness

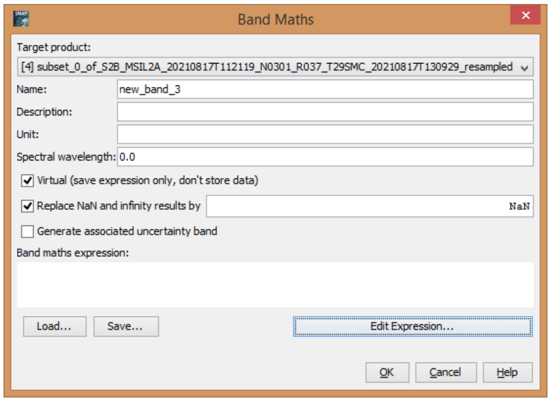
☐ Select all ☐ Select none

Estimated, raw storage size: 652.7M

CREATE A RATIO BAND

In SNAP, use **Raster > Band Maths** to create new image sample values derived from existing bands, tie-point grids, and flags. The source data can originate from all currently open and spatially compatible input products. The source data is combined with arbitrary mathematical expressions to generate the target data. By default, a new image view is automatically opened for the new sample values.

In this case, consider a simple ratio between Band 8 (NIR) and Band 4 (Red). ***A new band with the result is added to the product.***



Band Maths

Target product: [4] subset_0_of_S2B_MSIL2A_20210817T112119_N0301_R037_T29SMC_20210817T130929_resampled

Name: new_band_3

Description:

Unit:

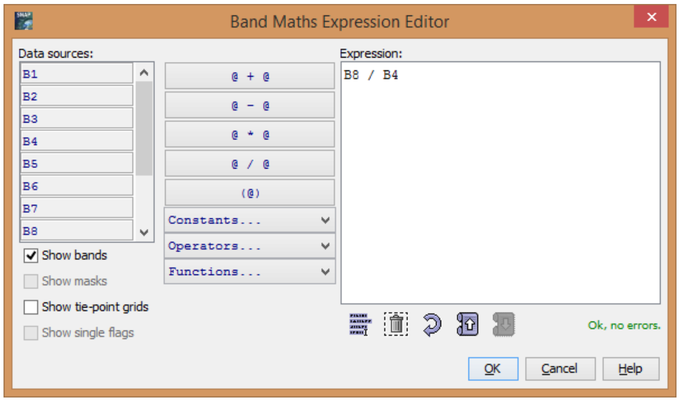
Spectral wavelength: 0.0

☑ Virtual (save expression only, don't store data)

☑ Replace NaN and infinity results by NaN

☐ Generate associated uncertainty band

Band maths expression:



Band Maths Expression Editor

Data sources: B1, B2, B3, B4, B5, B6, B7, B8

☑ Show bands

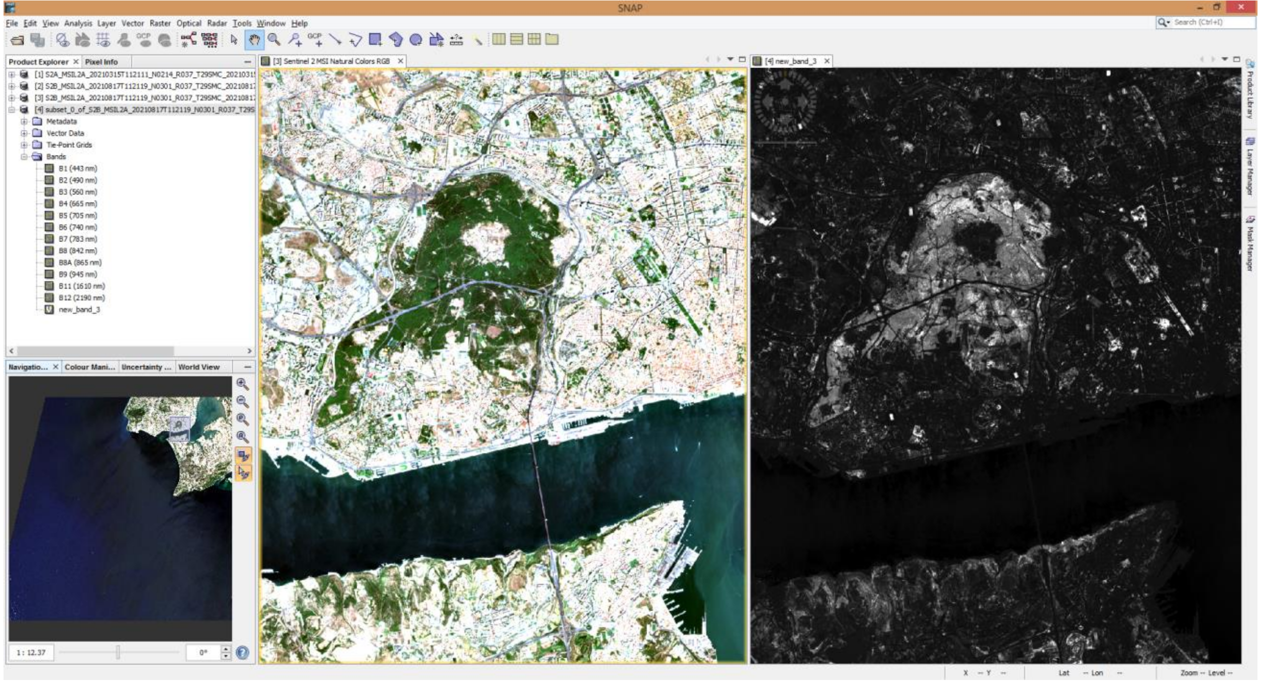
☐ Show masks

☐ Show tie-point grids

☐ Show single flags

Expression: B8 / B4

Ok, no errors.

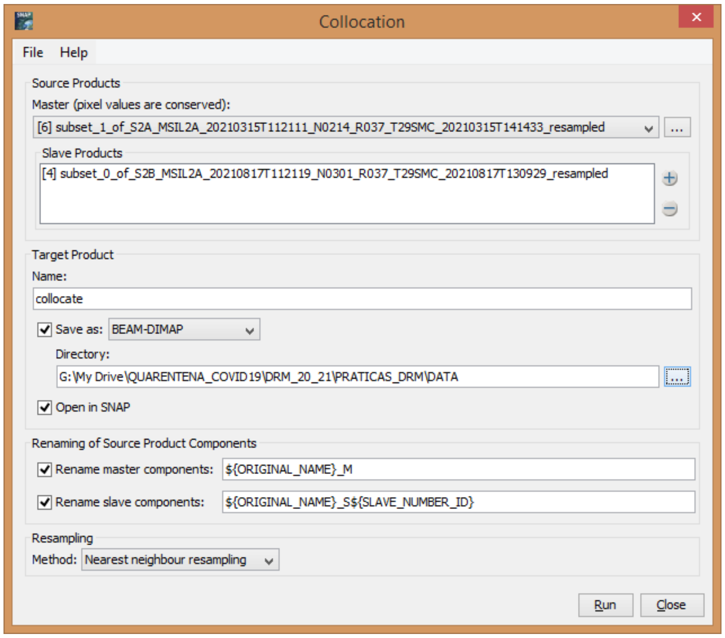


SNAP
File Edit View Analysis Layer Vector Raster Optical Radar Tools Window Help
Product Explorer
Pixel Info
[1] S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_2021031
[2] S2B_MSIL2A_20210817T112119_N0301_R037_T29SMC_202108
[3] S2B_MSIL2A_20210817T112119_N0301_R037_T29SMC_202108
[4] subset_0_of_S2B_MSIL2A_20210817T112119_N0301_R037_T29S
Metadata
Vector Data
Tie-Point Grids
Bands
B1 (443 nm)
B2 (490 nm)
B3 (560 nm)
B4 (665 nm)
B5 (705 nm)
B6 (740 nm)
B7 (783 nm)
B8 (842 nm)
B8A (865 nm)
B9 (945 nm)
B11 (1610 nm)
B12 (2190 nm)
new_band_3
[3] Sentinel 2 MSI Natural Colors RGB
[4] new_band_3
Navigation
Colour Mani...
Uncertainty ...
World View
Product Library
Layer Manager
Mask Manager

COLLOCATE OVERLAPPING PRODUCTS

In SNAP, use **Raster > Geometric > Collocation** to collocate spatially overlapping products. Collocating products implies that the pixel values of slave products are resampled into the geographical raster of the master product.

To avoid naming conflicts, the Collocation Tool allows renaming both master and slave components such as bands, flag codings, and bitmask definitions according to a user-defined pattern.



Source Products

Master (pixel values are conserved): [6] subset_1_of_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resampled

Slave Products: [4] subset_0_of_S2B_MSIL2A_20210817T112119_N0301_R037_T29SMC_20210817T130929_resampled

Target Product

Name: collocate

☑ Save as: BEAM-DIMAP

Directory: G:\My Drive\QUARENTENA_COVID19\DRM_20_21\PRATICAS_DRM\DATA

☑ Open in SNAP

Renaming of Source Product Components

☑ Rename master components: ${ORIGINAL_NAME}_M

☑ Rename slave components: ${ORIGINAL_NAME}_S${SLAVE_NUMBER_ID}

Resampling

Method: Nearest neighbour resampling

Alternatively, one can use **Radar > Coregistration > Stack Tools > Create Stack** to collocate two spatially overlapping products. Collocating two products imply that the pixel values of one product (the secondary) are resampled into the geographical raster of the other (the reference). When two products are collocated, the band data of the secondary product is resampled into the geographical raster of the reference product.

The collocation algorithm requires accurate geopositioning information for both reference and secondary products (select Product Geolocation). The metadata for the stacked product is copied from the metadata of the reference product. ***When the reference image (Master) extent is selected as the output extent for the collocated images, "None" must be used for the resampling type.***

EXERCISE 1.3

Perform the operations in exercise 1.2 using the **Graph Builder** and **Batch Processing** tools.

Graphs can be created visually with the Graph Builder, processed directly from the DAT, and then saved as XML files. These saved graphs can then also be used as the input for the command line Graph Processing Tool (GPT) with a different set of input data products.

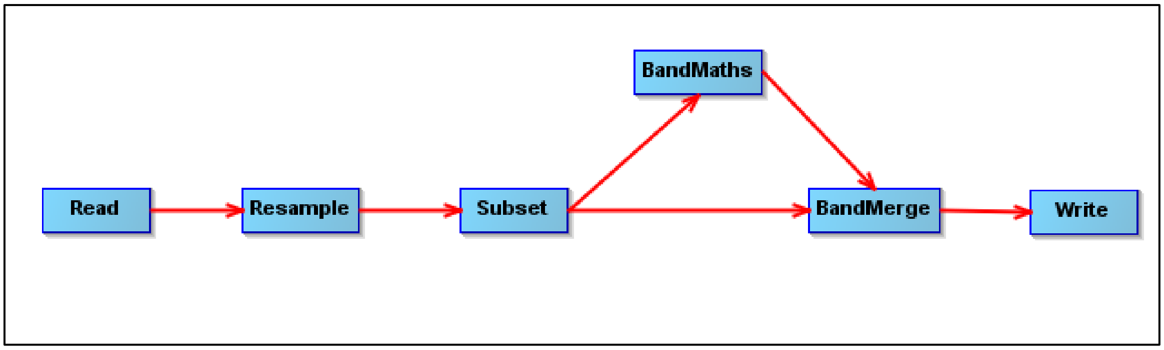
The Graph Builder allows the user to assemble graphs from a list of available operators and connect operator nodes to their sources.

The Batch Processing tool available via the DAT allows you to execute a single reader/writer graph for a set of products.

CREATE A GRAPH

In SNAP, use **Tools > Graph Builder** to create a graph with a sequence of operators.

In this case, the operations are **Resample**, **Subset**, and **Band Maths**. ***However, in Graph Builder, Band Maths only outputs the calculated band (instead of adding it to the original product, as mentioned in exercise 1.2), therefore another operator (Band Merge) has to be added to merge the new band and the original bands of the product.***



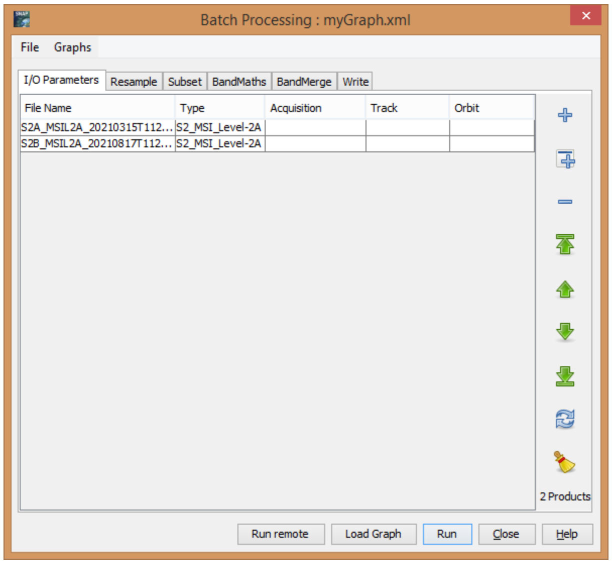
When using Graph Builder, the subset operation requires the insertion of the Region Of Interest (ROI) as a WKT String. For that purpose, use the shapefile ROI.shp that is in the OneDrive platform (**folder DRM2024**). Open this file in QGIS, select the polygon, and extract the WKT string using the WKT Plugin.

MultiPolygon (((-9.5 38.79999999999999716, -8.90000000000000036 38.79999999999999716, -8.90000000000000036 38.39999999999999858, -9.5 38.39999999999999858, -9.5 38.39999999999999858, -9.5 38.79999999999999716)))

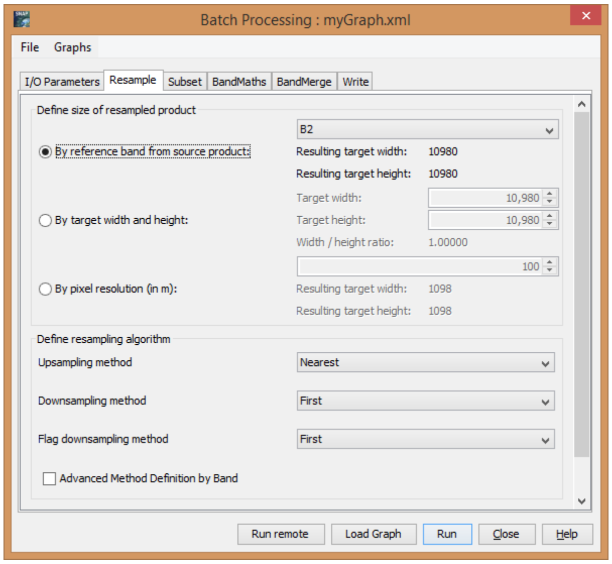
RUN THE GRAPH IN BATCH PROCESSING

In SNAP, use **Tools > Batch Processing** to execute a single reader/writer graph for a set of products. Press the "Load" button to browse for a previously saved graph. Next, add products in the IO tab by pressing the "Add" button or dragging and dropping a ProductSet or Products from the Project or Products views. Set the target folder where the output will be written to and then press "Run".

To avoid all target products having the same name, in the Write tab replace the name of the Target Product with “target”.



File Name	Type	Acquisition	Track	Orbit
S2A_MSIL2A_20210315T112...	S2_MSI_Level-2A			
S2B_MSIL2A_20210817T112...	S2_MSI_Level-2A			



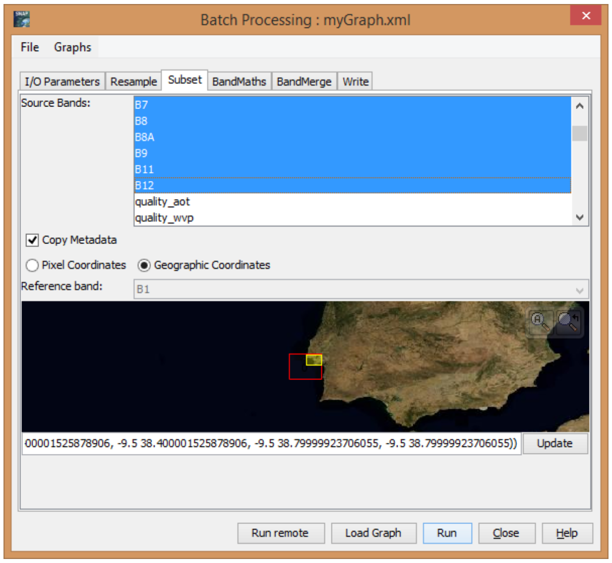
Define size of resampled product

● By reference band from source product: B2
Resulting target width: 10980
Resulting target height: 10980

○ By target width and height:
Target width: 10,980
Target height: 10,980
Width / height ratio: 1.00000

○ By pixel resolution (in m): 100
Resulting target width: 1098
Resulting target height: 1098

Define resampling algorithm
Upsampling method: Nearest
Downsampling method: First
Flag downsampling method: First
☐ Advanced Method Definition by Band



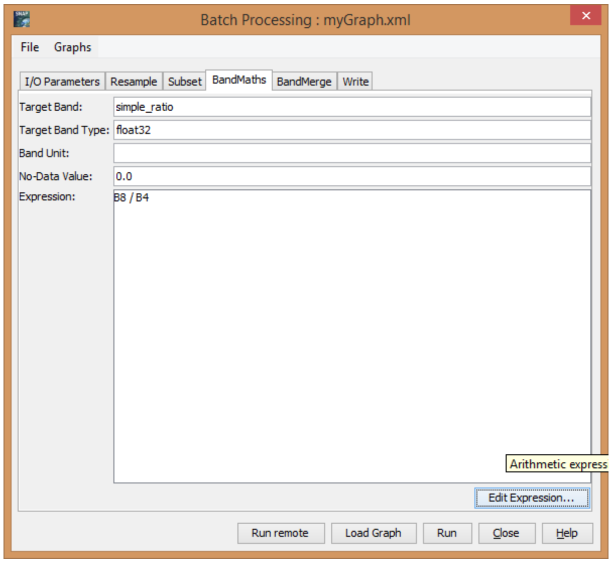
Source Bands: B7, B8, B8A, B9, B11, B12, quality_aot, quality_wvp

☑ Copy Metadata

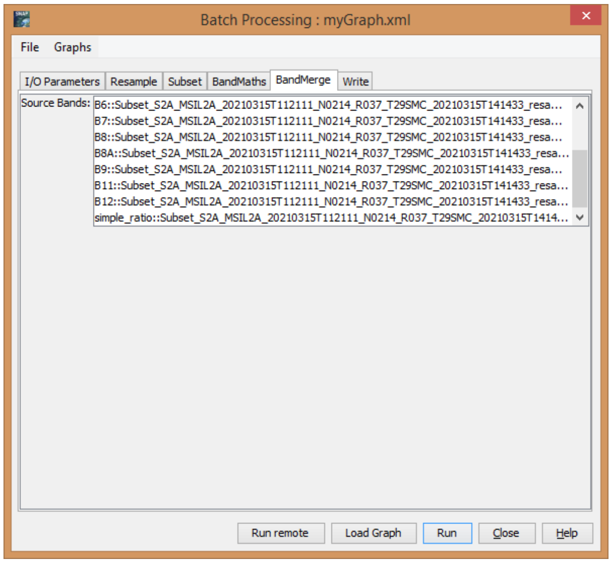
○ Pixel Coordinates ● Geographic Coordinates

Reference band: B1

00001525878906, -9.5 38.400001525878906, -9.5 38.79999923706055, -9.5 38.79999923706055))

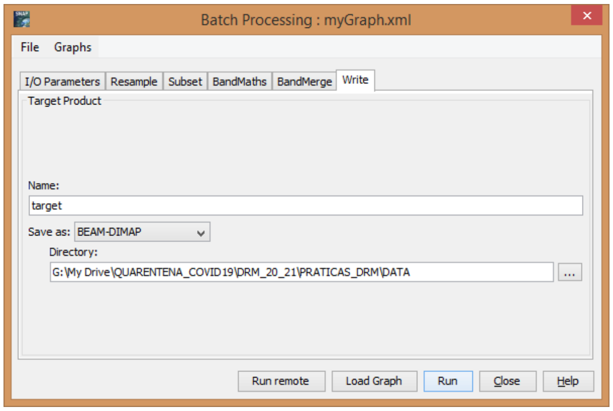


Target Band: simple_ratio
Target Band Type: float32
Band Unit:
No-Data Value: 0.0
Expression: B8 / B4



Source Bands:

```
B6::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resa...
B7::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resa...
B8::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resa...
B8A::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resa...
B9::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resa...
B11::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resa...
B12::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resa...
simple_ratio::Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T1414...
```



Target Product

Name: target

Save as: BEAM-DIMAP

Directory: G:\My Drive\QUARENTENA_COVID19\DRM_20_21\PRATICAS_DRM\DATA

PRACTICAL LESSON 2

EXERCISE 2.1

Apply several techniques for dimensionality reduction, such as spectral indices (vegetation, built-up, soil, and water indices, for example), principal component analysis, texture, and temporal metrics.

CALCULATE SPECTRAL INDICES

In SNAP, use **Raster > Band Maths** to create new image sample values derived from existing bands. Calculate the NDVI[1] (Normalized Difference Vegetation Index) band, the NDWI[2] (Normalized Difference Water Index) band, the SAVI[3] (Soil-Adjusted Vegetation Index), and the NDBI[4] (Normalized Difference Built-up Index).

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

$$NDWI = \frac{(Green - NIR)}{(Green + NIR)}$$

$$SAVI = \frac{1.5 * (NIR - Red)}{(NIR + Red + 0.5)}$$

$$NDBI = \frac{(SWIR1 - NIR)}{(SWIR1 + NIR)}$$

Alternatively, use **Optical > Thematic Land Processing** to calculate several radiometric indices included in the:

- **Soil Radiometric Indices**
- **Vegetation Radiometric Indices** (NDVI and SAVI)
- **Water Radiometric Indices** (NDWI2)

CALCULATE THE PRINCIPAL COMPONENTS OF AN IMAGE

In SNAP, use **Raster > Image Analysis > Principal Component Analysis** to generate the principal component images from a stack of co-registered detected images. The Principal Component Analysis (PCA) consists of a remapping of the information of the input co-registered images into a new set of images. The output images are scaled to prevent negative pixel values.

[1] Rouse, J.W., Jr., Haas, R. H., Schell, J. A., & Deering, D. W. (1974). Monitoring Vegetation Systems in the Great Plains with Erts, NASA Special Publication. *Proceedings of the Third Earth Resources Technology Satellite- 1 Symposium*, 309–317.

[2] McFeeters, S. K. (1996). The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *International Journal of Remote Sensing*, *17*(7), 1425–1432. https://doi.org/10.1080/01431169608948714.

[3] Huete, A. R. (1988). A soil-adjusted vegetation index (SAVI). *Remote Sensing of Environment*, *25*(3), 295–309. https://doi.org/10.1016/0034-4257(88)90106-X.

[4] Zha, Y., J. Gao, and S. Ni (2003). "Use of Normalized Difference Built-Up Index in Automatically Mapping Urban Areas from TM Imagery." *International Journal of Remote Sensing* 24, no. 3:,583-594.

The PCA operator consists of the following major steps:

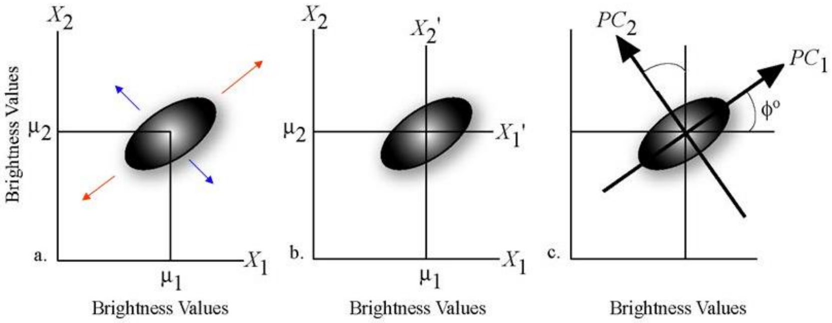
(1) Average the pixels across the input images to compute a mean image;

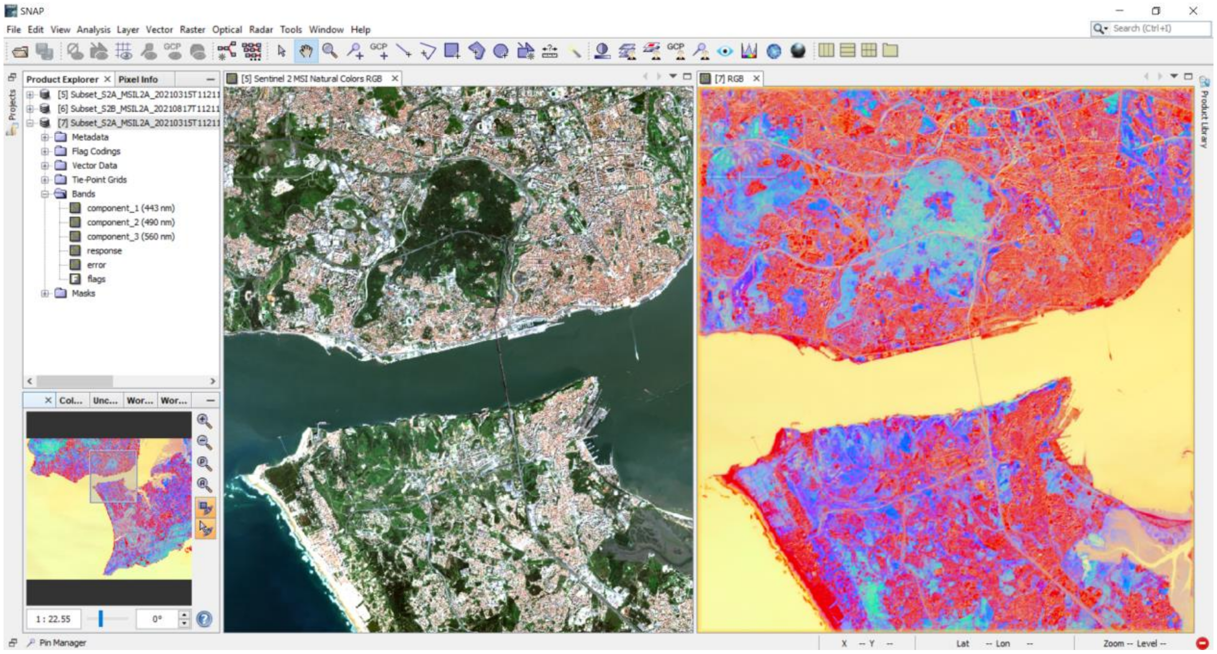
(2) Subtract the mean value of each input image (or image from step 1) from itself to produce zero-mean images;

(3) Compute the covariance matrix from the zero-mean images given in step 2;

(4) Perform eigenvalue decomposition of the covariance matrix;

(5) Compute PCA images by multiplying the eigenvector matrix by the zero-mean images given in step 2.





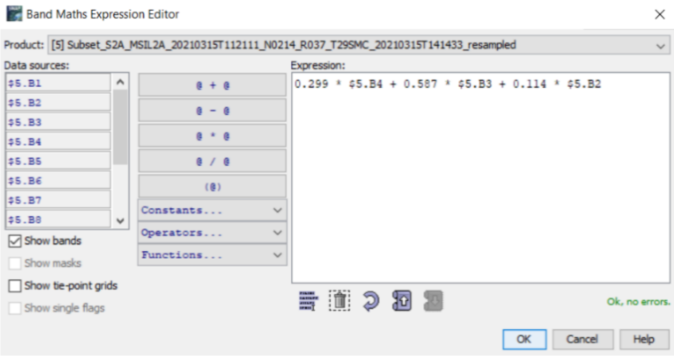
CALCULATE THE TEXTURE OF A BAND

In SNAP, use **Raster > Image Analysis > Texture Analysis > Grey Level Co-occurrence Matrix** to produce GLCM[5] (Gray Level Co-occurrence Matrix) texture features by making use of spatial information inherent in the image. The texture is the pattern of intensity variations in an image and can be a valuable tool in improving land-cover classification accuracy. Texture information involves the information from neighboring pixels which is important to characterize the identified objects or regions of interest in an image.

The GLCM is one of the most widely used methods to compute second-order texture measures. Several texture features can be computed from the GLCM matrix, each modeling different properties of the statistical relation of pixels co-occurrence estimated within a given moving window and along predefined directions and inter-pixel distances. The GLCM is a measure of the probability of occurrence of two grey levels separated by a given distance in a given direction. The features can be categorized into three groups, i.e. contrast group (contrast, dissimilarity, and homogeneity), orderliness group (angular second moment, maximum probability, and entropy), and statistics group (mean, variance, and correlation).

Before calculating the GLCM texture features, convert the RGB image to a grayscale image[6] using the following expression and **Raster > Band Maths**:

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$


Band Maths Expression Editor

Product: [5] Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resampled

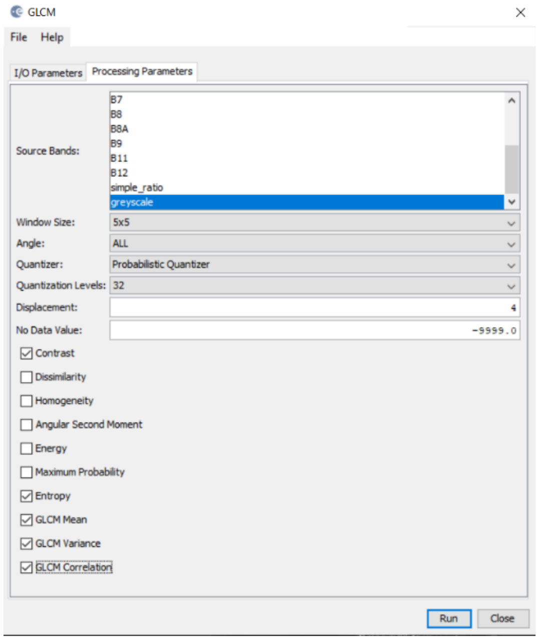
Data sources: $5.B1, $5.B2, $5.B3, $5.B4, $5.B5, $5.B6, $5.B7, $5.B8

☑ Show bands
□ Show masks
□ Show tie-point grids
□ Show single flags

Expression:

```
0.299 * $5.B4 + 0.587 * $5.B3 + 0.114 * $5.B2
```

Ok, no errors.


GLCM

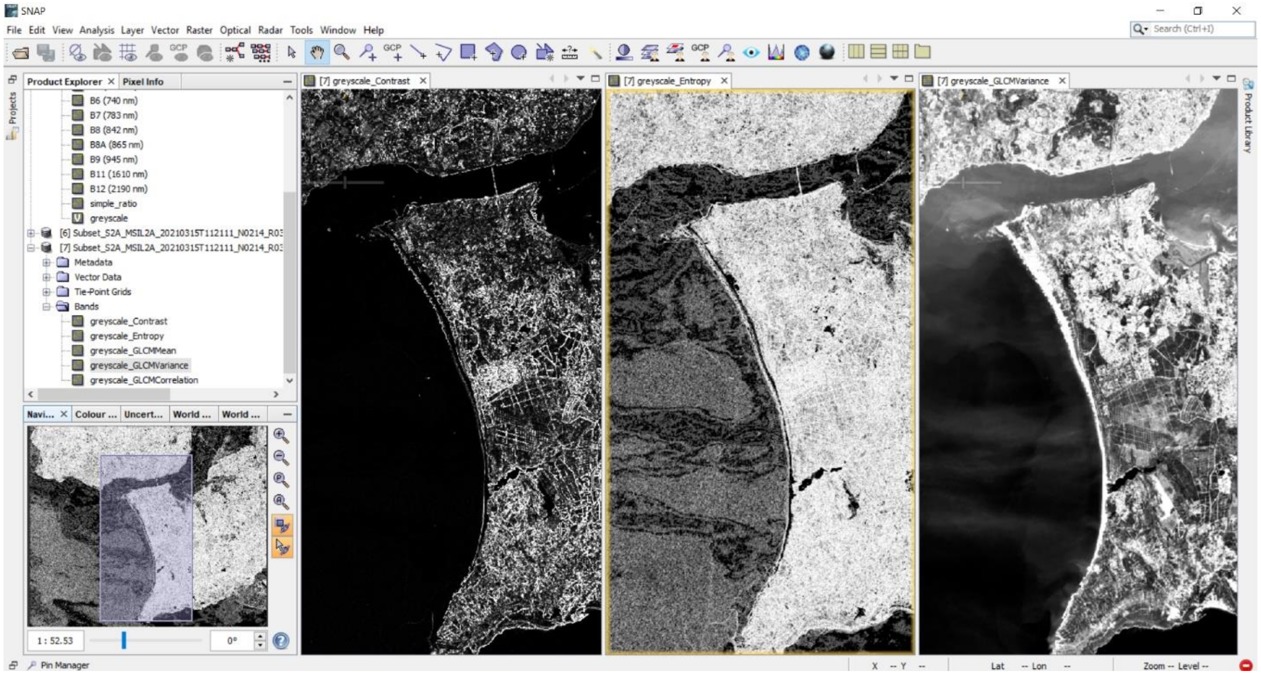
Source Bands: B7, B8, B8A, B9, B11, B12, simple_ratio, greyscale (selected: greyscale)

Window Size: 5x5
Angle: ALL
Quantizer: Probabilistic Quantizer
Quantization Levels: 32
Displacement: 4
No Data Value: -9999.0

☑ Contrast
□ Dissimilarity
□ Homogeneity
□ Angular Second Moment
□ Energy
□ Maximum Probability
☑ Entropy
☑ GLCM Mean
☑ GLCM Variance
☑ GLCM Correlation

[5] Haralick, R. M., Dinstein, I., & Shanmugam, K. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-3*(6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314.

[6] https://mmuratarat.github.io/2020-05-13/rgb_to_grayscale_formulas

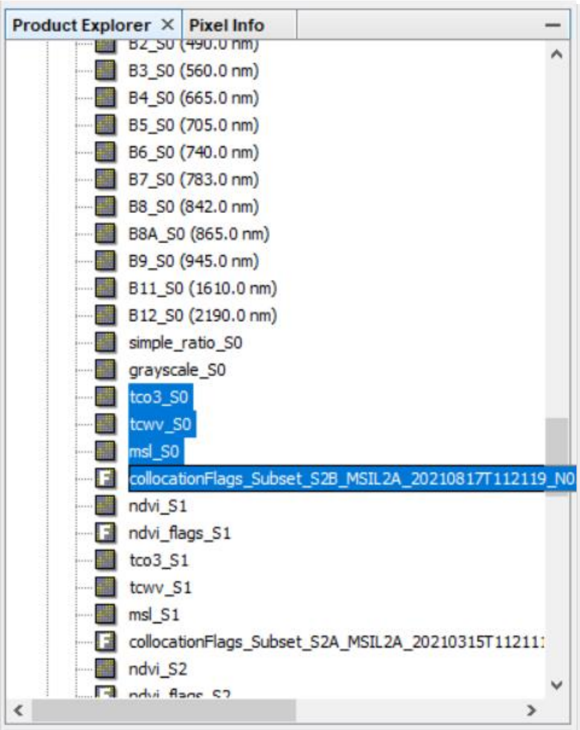


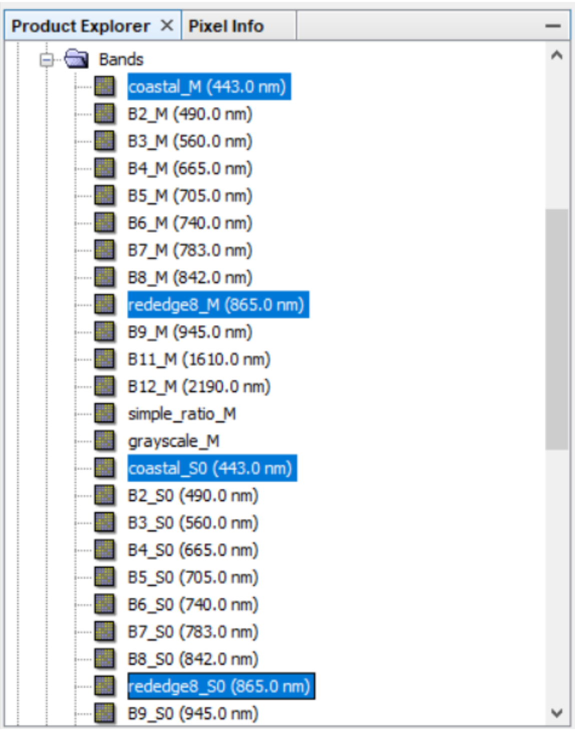
CALCULATE TEMPORAL METRICS (from a time series of images)

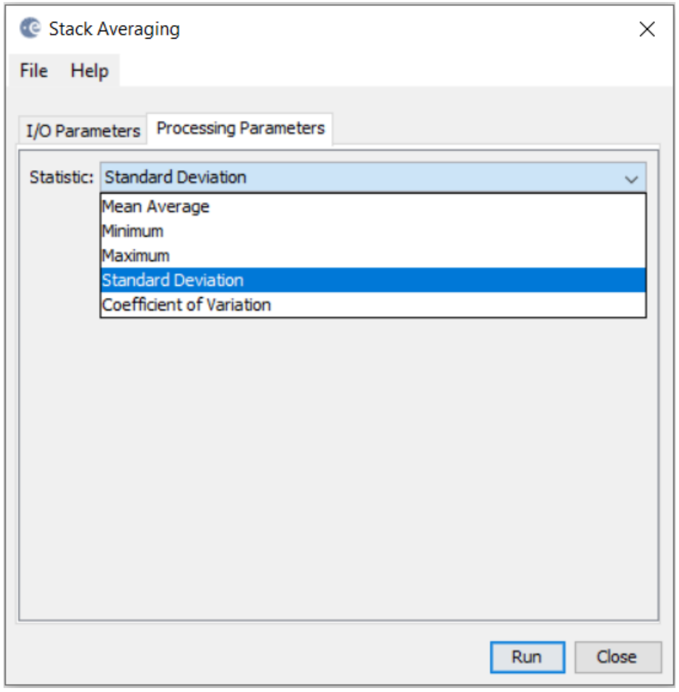
In SNAP, use **Radar > Coregistration > Stack tools > Stack Averaging** to compute statistics (including the mean, minimum, maximum, standard deviation, and coefficient of variation) for the corresponding bands in the given stack product. For example, the mean of all "B2_" bands.

However, for Sentinel-2 products, bands that share the same name (before the last underscore), such as the "B1_", "B11_" and "B12_" bands, and also "B8_" and "B8A_" bands, will be simultaneously (and incorrectly) considered for the mean (or other metric) calculation. To avoid this, rename the "B1_" band as "coastal_" band and "B8A_" band as "rededge8_" band.

NOTE: Before calculating the temporal metrics, delete all the colocation flags and other bands, besides the original bands and the subproducts, from the time series generated with **Raster > Geometric > Collocation**.







Statistic: Standard Deviation

PRACTICAL LESSON 3

EXERCISE 3.1

Perform change detection between two multispectral images using the MAD[7] (**Multivariate Alteration Detector**) algorithm implemented in the Orfeo Toolbox (OTB).

OTB performs change detection between two multispectral images using the MAD algorithm (https://www.orfeo-toolbox.org/CookBook/Applications/app_MultivariateAlterationDetector.html). The MAD algorithm produces a set of N change maps (where N is the maximum number of bands in first and second input images), with the following properties:

- Change maps are differences between a pair of linear combinations of bands from image 1 and bands from image 2 chosen to maximize the correlation,
- Each change map is orthogonal to the others.

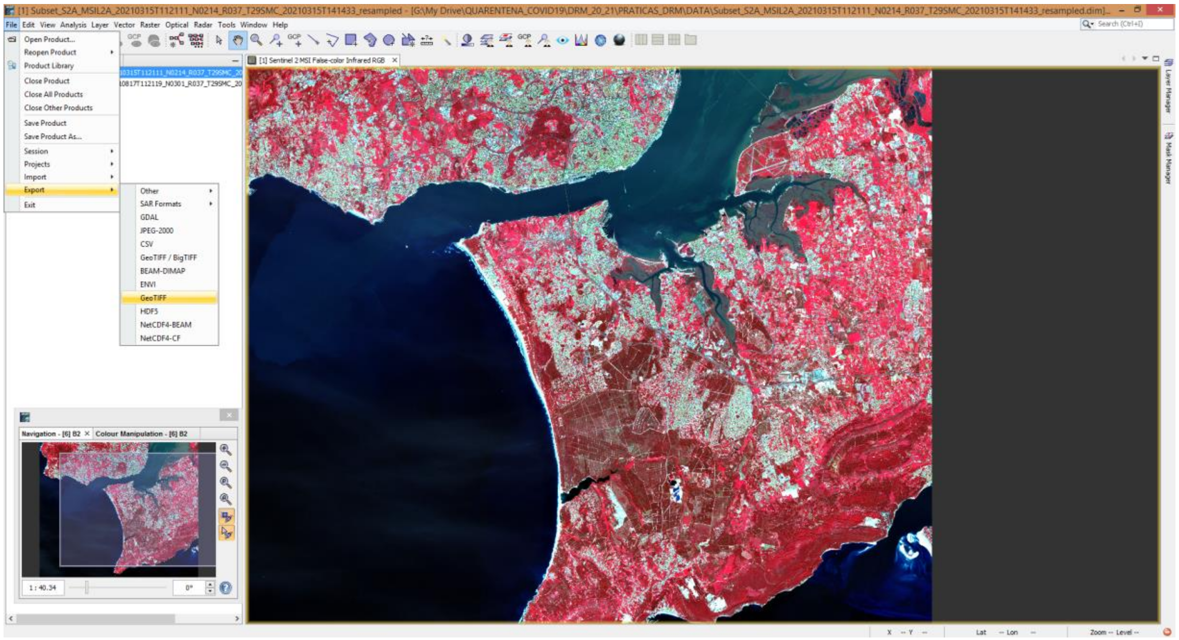
This is a statistical method that can handle different modalities and even different bands and number of bands between images. The application will output all change maps into a single multiband image. Change maps are sorted by increasing correlation.
The application will also print the following information:

- Mean1 and Mean2 which are the mean values of bands for both input images,
- V1 and V2 which are the two linear transforms that are applied to input image 1 and input image 2 to build the change map,
- Rho, the vector of correlation associated with each change map.

EXPORT MULTISPECTRAL IMAGES FROM BEAM-DIMAP FORMAT (SNAP) TO GeoTIFF FORMAT

In SNAP, use **File > Export** to export both images, generated with the graph in exercise 1.3, from BEAM_DIMAP format (*.dim) to GeoTIFF format (*.tif), which is recognizable by the QGIS and OTB.

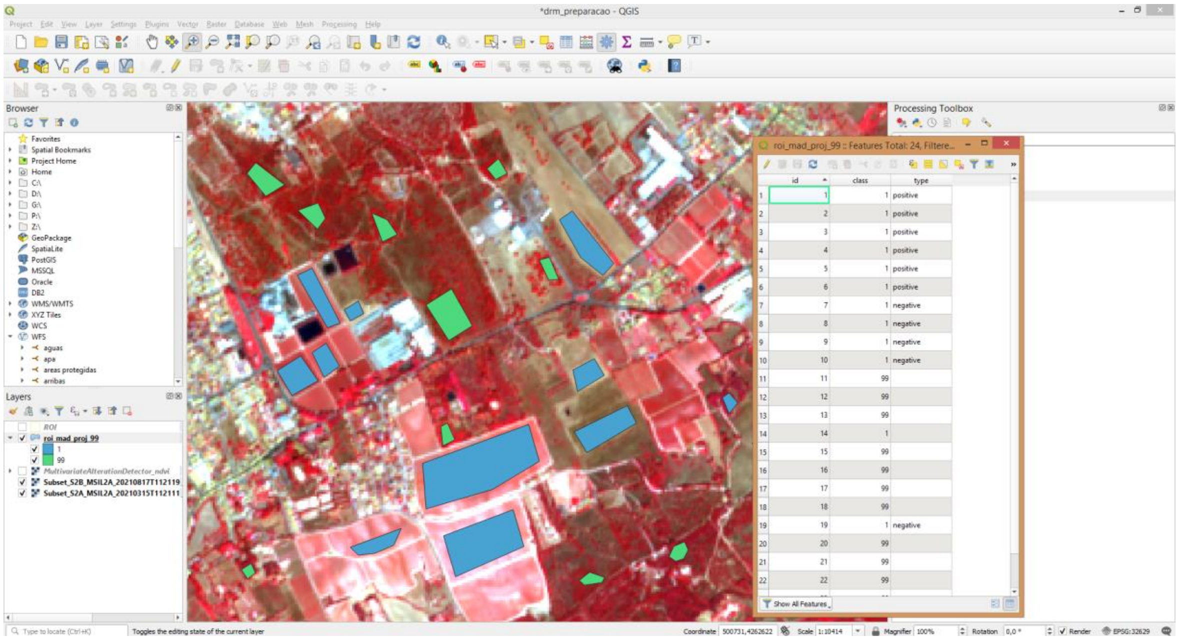
[7] Nielsen, A. A., & Conradsen, K. (1997). Multivariate alteration detection (MAD) in multispectral, bi-temporal image data: A new approach to change detection studies.



CREATE A POLYGON VECTOR FILE WITH SAMPLES FOR CHANGE DETECTION VALIDATION

In QGIS, use **Layer > Create Layer > New Shapefile Layer** to create a polygon shapefile with change/no change areas for change detection validation. Use the following coordinate reference system (EPSG: 32629-WGS84/UTM zone 29N) and add a new field ("class") to the attribute table, with the corresponding value for the two classes (change/no change): 1 for the "change" class and 99 for the "no change" class.

Instead of 1 for the "change" class and 0 for the "no change" class, use 1 for the "change" class and 99 for "no change" class due to some limitations of the ***OTB application "ComputeConfusionMatrix"*** *when dealing with the 0 value.*

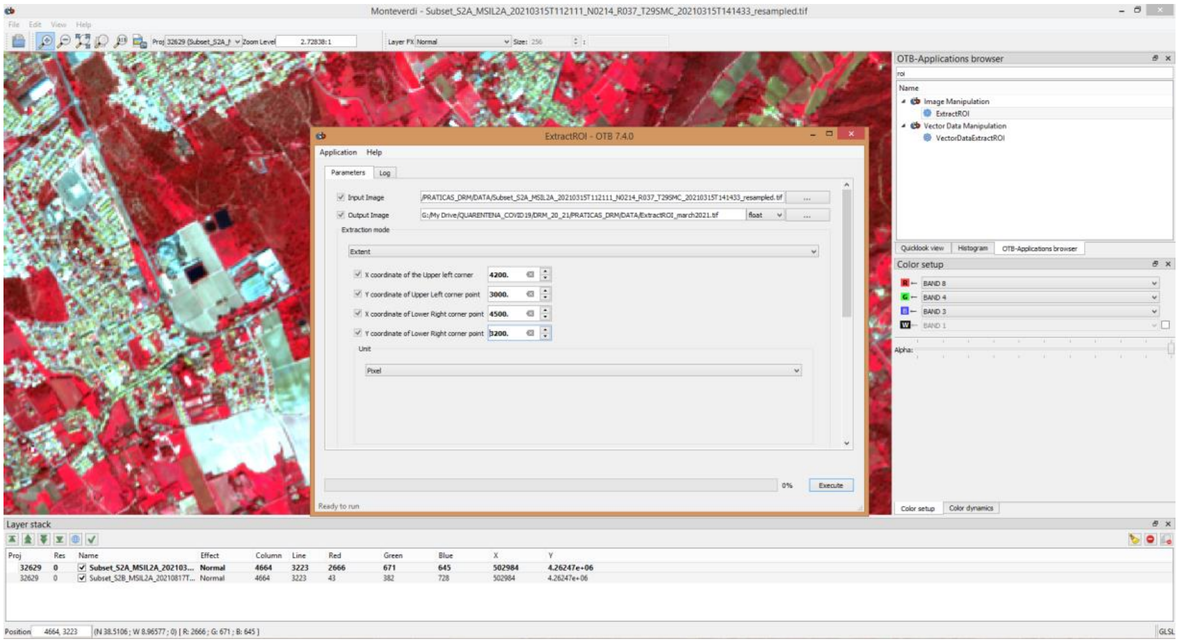


roi_mad_proj_99 :: Features Total: 24, Filtere...

	id	class	type
1	1	1	positive
2	2	1	positive
3	3	1	positive
4	4	1	positive
5	5	1	positive
6	6	1	positive
7	7	1	negative
8	8	1	negative
9	9	1	negative
10	10	1	negative
11	11	99	
12	12	99	
13	13	99	
14	14	1	
15	15	99	
16	16	99	
17	17	99	
18	18	99	
19	19	1	negative
20	20	99	
21	21	99	
22	22	99	

CREATE A SUBSET

In OTB, use **Image Manipulation > ExtractROI**[8] to extract an area of interest. First, use the Extraction mode "**Extent**" to subset, by pixel coordinates, one of the images. Then, use the Extraction mode "**Fit**" to subset the other image using a reference image (the one that was previously subset).


Monteverdi - Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resampled.tif

ExtractROI - OTB 7.4.0

☑ Input Image: /PRATICAS_DRM/DATA/Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resampled.tif

☑ Output Image: G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/ExtractROI_march2021.tif (float)

Extraction mode: Extent

☑ X coordinate of the Upper left corner: 4200.

☑ Y coordinate of Upper Left corner point: 3000.

☑ X coordinate of Lower Right corner point: 4500.

☑ Y coordinate of Lower Right corner point: 3200.

Unit: Pixel

0% Execute

Ready to run

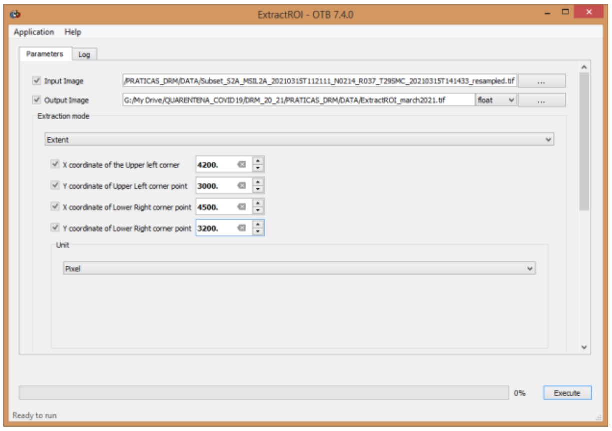
OTB-Applications browser: roi
- Image Manipulation
  - ExtractROI
- Vector Data Manipulation
  - VectorDataExtractROI

Color setup: R BAND 8; G BAND 4; B BAND 3; W BAND 1

Layer stack

Proj	Res	Name	Effect	Column	Line	Red	Green	Blue	X	Y
32629	0	☑ Subset_S2A_MSIL2A_202103...	Normal	4664	3223	2666	671	645	502984	4.26247e+06
32629	0	☑ Subset_S2B_MSIL2A_20210817T...	Normal	4664	3223	43	382	728	502984	4.26247e+06

Position 4664, 3223 (N 38.5106 ; W 8.96577 ; 0) [R: 2666 ; G: 671 ; B: 645]


ExtractROI - OTB 7.4.0

☑ Input Image: /PRATICAS_DRM/DATA/Subset_S2A_MSIL2A_20210315T112111_N0214_R037_T29SMC_20210315T141433_resampled.tif

☑ Output Image: G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/ExtractROI_march2021.tif (float)

Extraction mode: Extent

☑ X coordinate of the Upper left corner: 4200.

☑ Y coordinate of Upper Left corner point: 3000.

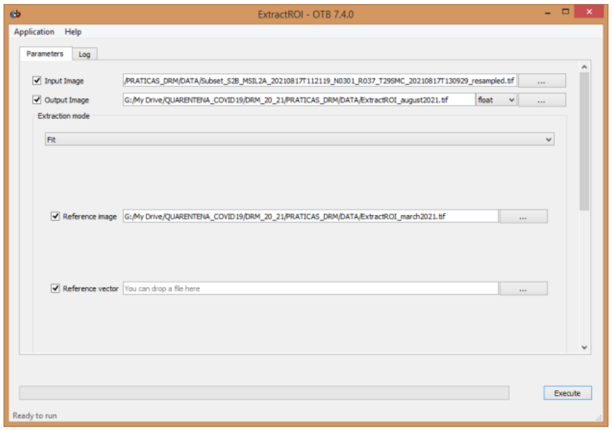
☑ X coordinate of Lower Right corner point: 4500.

☑ Y coordinate of Lower Right corner point: 3200.

Unit: Pixel

0% Execute

Ready to run


ExtractROI - OTB 7.4.0

☑ Input Image: /PRATICAS_DRM/DATA/Subset_S2B_MSIL2A_20210817T112119_N0301_R037_T29SMC_20210817T130929_resampled.tif

☑ Output Image: G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/ExtractROI_august2021.tif (float)

Extraction mode: Fit

☑ Reference image: G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/ExtractROI_march2021.tif

☑ Reference vector: You can drop a file here

Execute

Ready to run

[8] https://www.orfeo-toolbox.org/CookBook/Applications/app_ExtractROI.html

CALCULATE NDVI BANDS

In OTB, use **Feature Extraction > RadiometricIndices**[9] to compute radiometric indices (in this case, the **NDVI**) using the relevant channels of the input image.

As Sentinel-2 Level-2A products provide values coded in JPEG2000 with the same quantification value of 10,000 as for Level-1C products, a factor of 1/10,000 needs to be applied to Level-2A digital numbers (DN) to retrieve physical surface reflectance (SR) values. The physical values range from 1 (minimum reflectance 10^{-4}) to 10,000 (reflectance 1), but values higher than 1 can be observed in some cases due to specific angular reflectivity effects. The value 0 is reserved for "No Data"[10]*.*

This is particularly necessary for indices that use factors, such as the SAVI, since those factors were estimated for physical reflectance values. So, before calculating the indices, in OTB, use **Image Manipulation > BandMathX**[11] to perform a mathematical operation on several multi-band images and outputs the result into an image (multi- or mono-band, as opposed to the BandMath OTB-application). The mathematical formula is done by the muParserX library. Separating expressions by semicolons (;) will concatenate their results into a unique multiband output image.

In this case, the expression to be used to convert, at the same time, all the bands of an image from DN to SR values should be:

Since we intend to convert all the bands at the same time, the expression should be:

im1b1/10000; im1b2/10000; …; im1bn/10000

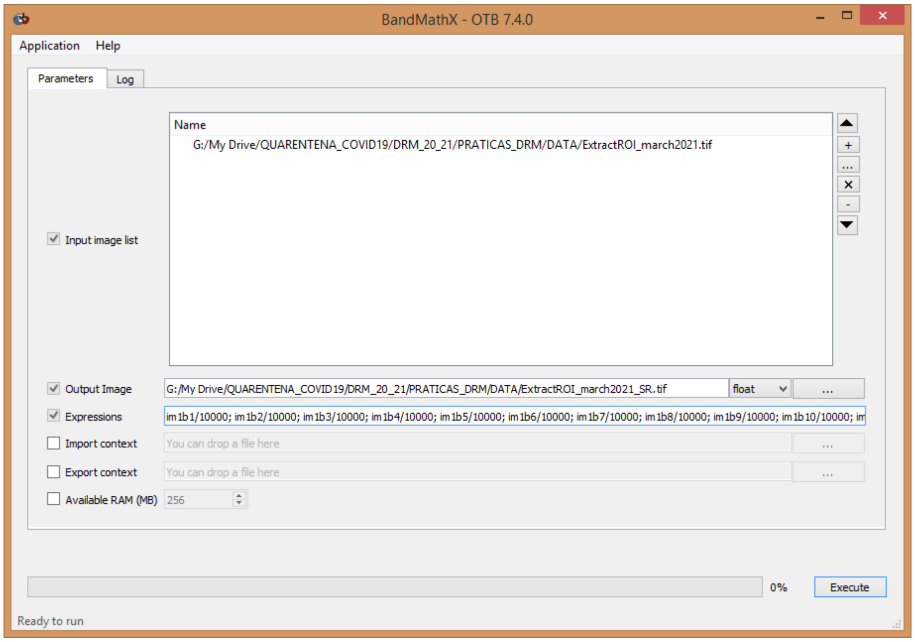
where **im** identifies the image, that in this case is unique, and **b** identifies the band, that in this case goes from 1 up to n.

DN (NIR band)= 4760 DN (Red band)= 235	**NDVI= 0.906**	$NDVI = \frac{(NIR - Red)}{(NIR + Red)}$	SR (NIR band)= 0.4760 SR (Red band)= 0.0235	**NDVI= 0.906** (range from -1 to 1)
	SAVI= 1.359	$SAVI = \frac{1.5 * (NIR - Red)}{(NIR + Red + 0.5)}$		**SAVI= 0.679** (range from -1 to 1)

[9] https://www.orfeo-toolbox.org/CookBook/Applications/app_RadiometricIndices.html

[10] https://www.mdpi.com/2072-4292/9/6/584 (in section 3.5, page 12/81)

[11] https://www.orfeo-toolbox.org/CookBook/Applications/app_BandMathX.html



G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/ExtractROI_march2021.tif

☑ Input image list

☑ Output Image G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/ExtractROI_march2021_SR.tif float

☑ Expressions im1b1/10000; im1b2/10000; im1b3/10000; im1b4/10000; im1b5/10000; im1b6/10000; im1b7/10000; im1b8/10000; im1b9/10000; im1b10/10000; im

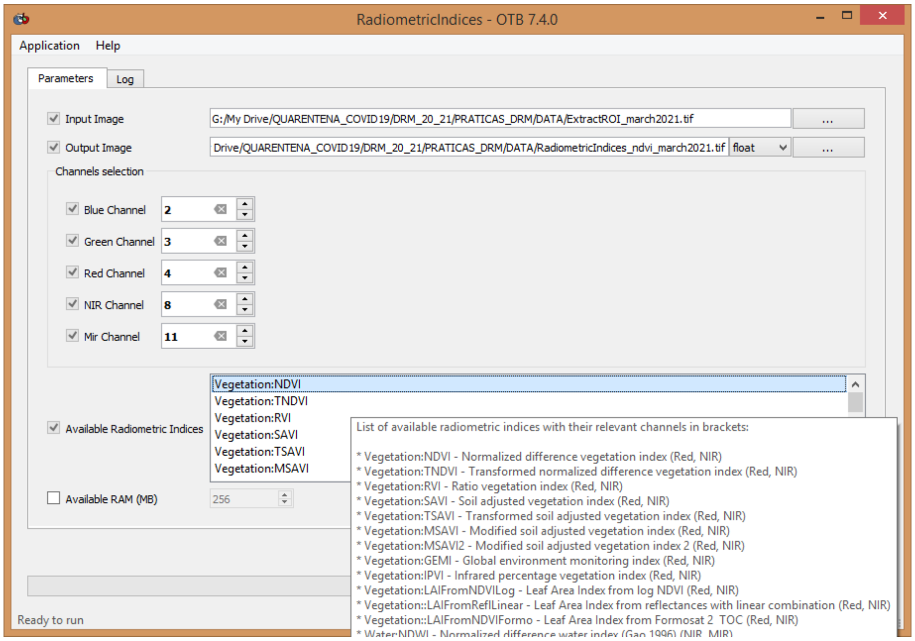
□ Import context You can drop a file here

□ Export context You can drop a file here

□ Available RAM (MB) 256

0%

Ready to run



☑ Input Image G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/ExtractROI_march2021.tif

☑ Output Image Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/RadiometricIndices_ndvi_march2021.tif float

Channels selection

☑ Blue Channel 2

☑ Green Channel 3

☑ Red Channel 4

☑ NIR Channel 8

☑ Mir Channel 11

☑ Available Radiometric Indices

Vegetation:NDVI
Vegetation:TNDVI
Vegetation:RVI
Vegetation:SAVI
Vegetation:TSAVI
Vegetation:MSAVI

□ Available RAM (MB) 256

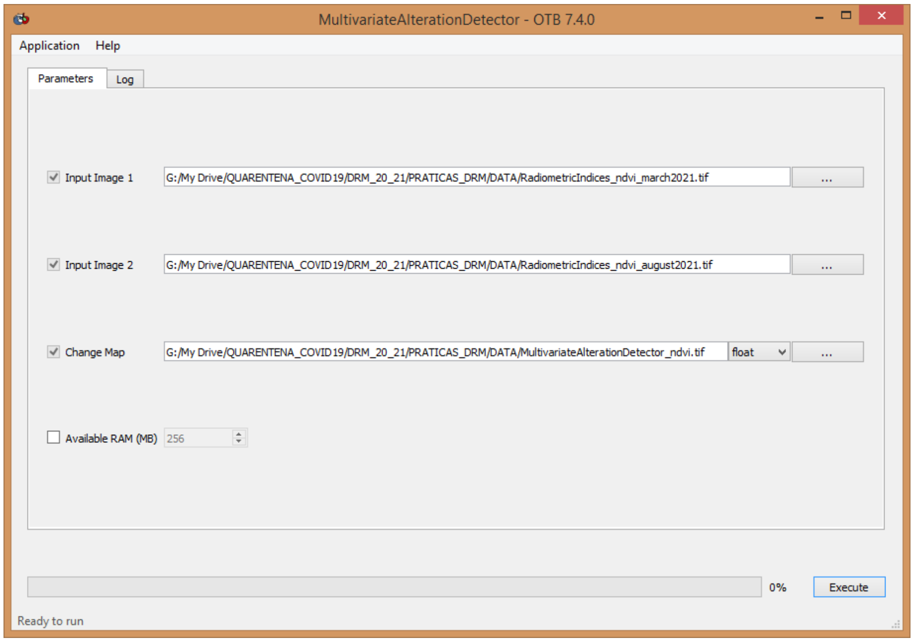
List of available radiometric indices with their relevant channels in brackets:

* Vegetation:NDVI - Normalized difference vegetation index (Red, NIR)
* Vegetation:TNDVI - Transformed normalized difference vegetation index (Red, NIR)
* Vegetation:RVI - Ratio vegetation index (Red, NIR)
* Vegetation:SAVI - Soil adjusted vegetation index (Red, NIR)
* Vegetation:TSAVI - Transformed soil adjusted vegetation index (Red, NIR)
* Vegetation:MSAVI - Modified soil adjusted vegetation index (Red, NIR)
* Vegetation:MSAVI2 - Modified soil adjusted vegetation index 2 (Red, NIR)
* Vegetation:GEMI - Global environment monitoring index (Red, NIR)
* Vegetation:IPVI - Infrared percentage vegetation index (Red, NIR)
* Vegetation:LAIFromNDVILog - Leaf Area Index from log NDVI (Red, NIR)
* Vegetation::LAIFromReflLinear - Leaf Area Index from reflectances with linear combination (Red, NIR)
* Vegetation::LAIFromNDVIFormo - Leaf Area Index from Formosat 2 TOC (Red, NIR)
* Water:NDWI - Normalized difference water index (Gao 1996) (NIR, MIR)

Ready to run

APPLY THE MULTIVARIATE ALTERATION DETECTOR (to a pair of images)

In OTB, use **Change Detection > MultivariateAlterationDetector**[12] to perform change detection between two NDVI images, created in the previous step, using the MAD algorithm.



☑ Input Image 1 G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/RadiometricIndices_ndvi_march2021.tif

☑ Input Image 2 G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/RadiometricIndices_ndvi_august2021.tif

☑ Change Map G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/MultivariateAlterationDetector_ndvi.tif float

□ Available RAM (MB) 256

Ready to run

CREATE A BINARY MASK (CHANGE/NO CHANGE AREAS)

In OTB, use **Image Manipulation > BandMath**[13] to define a threshold condition to generate a binary mask with: 1 for the class "change" and 99 for the class "no change", as defined previously for the attribute "class " of the validation shapefile.

After the definition of a threshold value by visual interpretation (as an example let´s consider that the threshold is the value 1.2), the expression (if-then-else operator: (condition ? value_true : value_false)) should be:

(im1b1>=1.2 or im1b1<=-1.2 ? 1 : 99)

where **im** identifies the image, that in this case is unique, and **b** identifies the band (also unique in this case), the symbol (?) separates the condition from the true value, and the symbol (:) separates the else value. In this case, we have an operator (or), since the change might be positive or negative.

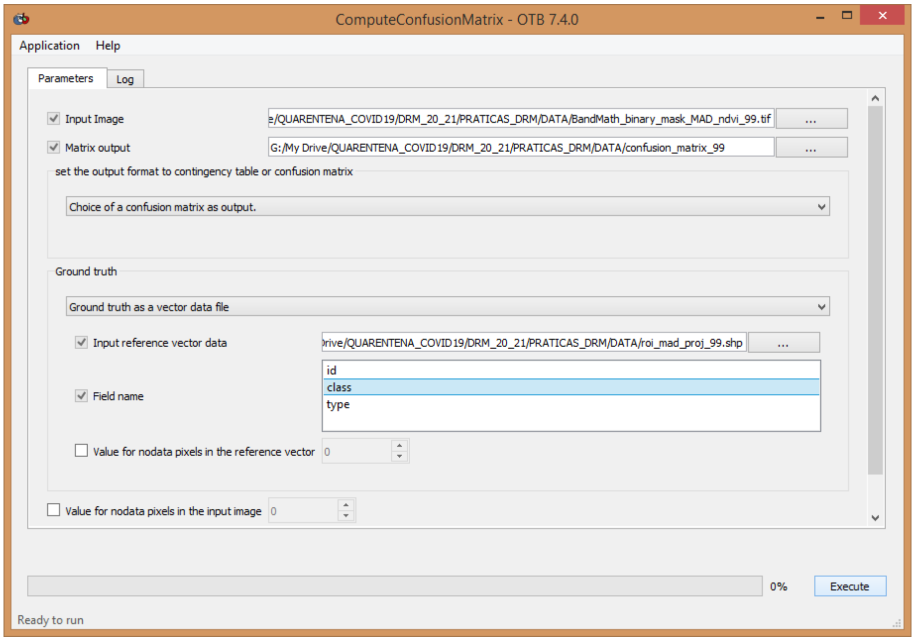
[12] https://www.orfeo-toolbox.org/CookBook/Applications/app_MultivariateAlterationDetector.html

[13] https://www.orfeo-toolbox.org/CookBook/Applications/app_BandMath.html

COMPUTE A CONFUSION MATRIX

In OTB, use **Learning > ComputeConfusionMatrix**[14] to compute the confusion matrix of a classification.

This application computes the confusion matrix of a classification map relative to a ground truth dataset. The ground truth can be provided as either a raster or a vector data. Only reference and produced pixels with values different from NoData are handled in the calculation of the confusion matrix. *In OTB, the confusion matrix is organized the following way: rows = reference labels, columns = produced labels*. In the header of the output file, the reference and produced class labels are ordered according to the rows/columns of the confusion matrix.



☑ Input Image: e/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/BandMath_binary_mask_MAD_ndvi_99.tif

☑ Matrix output: G:/My Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/confusion_matrix_99

set the output format to contingency table or confusion matrix: Choice of a confusion matrix as output.

Ground truth: Ground truth as a vector data file

☑ Input reference vector data: Drive/QUARENTENA_COVID19/DRM_20_21/PRATICAS_DRM/DATA/roi_mad_proj_99.shp

☑ Field name: id / class / type

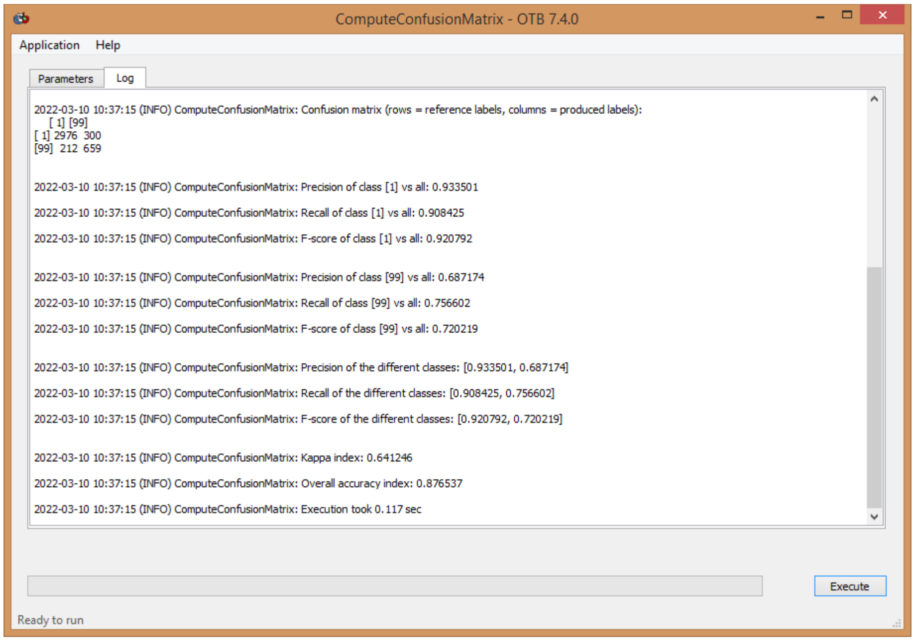
☐ Value for nodata pixels in the reference vector: 0

☐ Value for nodata pixels in the input image: 0

0%

Ready to run

[14] https://www.orfeo-toolbox.org/CookBook/Applications/app_ComputeConfusionMatrix.html



```
2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Confusion matrix (rows = reference labels, columns = produced labels):
    [ 1] [99]
[ 1] 2976  300
[99]  212  659


2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Precision of class [1] vs all: 0.933501

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Recall of class [1] vs all: 0.908425

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: F-score of class [1] vs all: 0.920792


2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Precision of class [99] vs all: 0.687174

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Recall of class [99] vs all: 0.756602

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: F-score of class [99] vs all: 0.720219


2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Precision of the different classes: [0.933501, 0.687174]

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Recall of the different classes: [0.908425, 0.756602]

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: F-score of the different classes: [0.920792, 0.720219]


2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Kappa index: 0.641246

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Overall accuracy index: 0.876537

2022-03-10 10:37:15 (INFO) ComputeConfusionMatrix: Execution took 0.117 sec
```

A confusion (or error) matrix compares information from reference samples with the corresponding information on the produced map the same samples. The matrix is a square array of numbers set out in rows and columns which express the labels of samples assigned to a particular category in one classification relative to the labels of samples assigned to a particular category in another classification.

One of the classifications, is assumed to be correct and is termed the reference data and is displayed in rows in OTB. Thus, OTB displays the map labels or classified data generated from the remotely sensed image in columns. Thus, two labels from each sample are compared to one another:

Reference data labels: The class label or value of the accuracy assessment site, which is derived from data collected that is assumed to be correct; and

Classified data or map labels: The class label or value of the accuracy assessment site derived from the map.

Error matrices are very effective representations of map accuracy because the individual accuracies of each map category are plainly described along with both the errors of inclusion (commission errors) and errors of exclusion (omission errors) present in the map. A commission error occurs when an area is included in an incorrect category. An omission error occurs when an area is excluded from the category to which it belongs. Every error on the map is an omission from the correct category and a commission to an incorrect category.

In addition to clearly showing errors of omission and commission, the error matrix can be used to compute not only the **overall accuracy**, but also the producer's accuracy or **recall** (the complement of the omission error) and the user's accuracy or **precision** (complement of the commission error). Overall accuracy is simply the sum of the major diagonal (i.e., the correctly classified sample units) divided by the total number of sample units in the error matrix. Recall and precision are ways of representing individual category accuracies instead of just the overall classification accuracy.

		PRODUCED LABELS				
	CLASS	**Change**	**No Change**	Sum rows	**Recall**	**F1-score**
REFERENCE LABELS	**Change**	**2976**	300	3276	2979/3276= **0.91**	2x(0.93x0.91)/(0.93+0.91)= **0.92**
	No Change	212	**659**	871	659/871= **0.76**	2x(0.69x0.76)/(0.69+0.76)= **0.72**
	Sum columns	3188	959	4147 (*)		
	Precision	2976/3188= **0.93**	659/959= **0.69**			
	Overall Accuracy= (2976+659)/4147= **0.88** **Kappa Coefficient**= (4147*(2976+659)-(3276x3188+871x959))/ (4147^2-(3276x3188+871x959))= **0.64**					

(*) the sum of all column's totals must be equal to the sum of all rows totals.

The **Kappa** coefficient is a measure of overall agreement of a matrix. In contrast to the overall accuracy — the ratio of the sum of diagonal values to total number of cells counts in the matrix — the Kappa coefficient takes also non-diagonal elements into account. Therefore, the Kappa coefficient measures the proportion of agreement after chance agreements have been removed from considerations. A Kappa value of 1 represents perfect agreement while a value of 0 represents no agreement. Kappa has the following formulation:

$$K = \frac{N \sum_{i=1}^{n} m_{i,i} - \sum_{i=1}^{n} G_i C_i}{N^2 - \sum_{i=1}^{n} G_i C_i}$$

where *i* is the class number; *N* is the total number of classified pixels that are being compared to ground truth; $m_{i,i}$ is the number of pixels belonging to the ground truth class *i*, that have also been classified with a class *i* (i.e., values found along the diagonal of the confusion matrix); C_i is the total number of classified pixels belonging to class *i*; and G_i is the total number of ground truth pixels belonging to class *i*.

Besides, F1-score is also used in machine-learning. F1-score is the weighted average of precision and recall. Therefore, this score takes both commission and omission errors into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if commission and omission errors have similar cost. If the cost of commission and omission errors are very different, it's better to look at both precision and recall.

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall}$$

PRACTICAL LESSON 4

EXERCISE 4.1

Classify a composite with two Sentinel-2 images acquired in March and August 2021 (*subset_0_collocate.tif*) using the Random Forest classifier. The image composite has 26 bands, 13 for each acquisition date (B1-B8, B8A, B9, B11-B12, simple_ratio).

The training and validation data are in a polygon shapefile created in QGIS (*reference_samples.shp*), whose polygons correspond to six different classes adopted from the Level 1 of the COSsim (a Simplified Land Cover and Land Use Map for Portugal produced by DGT) nomenclature[15] adopted (1: Built-up Areas, 2: Agriculture; 3: Forest; 4: Bushes and spontaneous herbaceous vegetation; 5: Surfaces without vegetation; 6: Water and wetlands).

Both types of data can be downloaded from the OneDrive platform (**folder DRM2024**).

Orfeo ToolBox offers a set of applications to perform supervised or unsupervised pixel-based image classification[16]. This framework allows to learn from multiple images and using several-machine learning methods such as SVM, Bayes, *k*-NN, Random Forests, Artificial Neural Network, and others (see application help of TrainImagesClassifier and TrainVectorClassifier for further details about all the available classifiers). Here is an overview of the complete workflow:

1. Compute samples statistics for each image;
2. Compute sampling rates for each image (only if more than one input image);
3. Select samples positions for each image;
4. Extract samples measurements for each image;
5. Compute images statistics;
6. Train the machine learning model from samples;
7. Perform the classification by applying the model.

STEP 1: COMPUTE SAMPLES STATISTICS FOR EACH IMAGE (PolygonClassStatistics)

In OTB, use **Learning > PolygonClassStatistics**[17] to compute statistics on a training polygon set. Process a set of geometries intended for training (they should have a field giving the associated class). The geometries are analyzed against a support image to compute statistics.

[15] https://geo2.dgterritorio.gov.pt/atom-dgt/COSsim/Nomenclatura_COSsim.pdf
[16] https://www.orfeo-toolbox.org/CookBook/recipes/pbclassif.html#pixel-based-classification
[17] https://www.orfeo-toolbox.org/CookBook/Applications/app_PolygonClassStatistics.html

The first step of the framework is to know how many samples are available for each class in your image. The PolygonClassStatistics will do this job for you. This application processes a set of training geometries and an image and outputs statistics about available samples (i.e. pixel covered by the image and out of a no-data mask if provided), in the form of an XML file:

- number of samples per class;
- number of samples per geometry.

Supported geometries are polygons, lines and points. Depending on the geometry type, this application behaves differently:

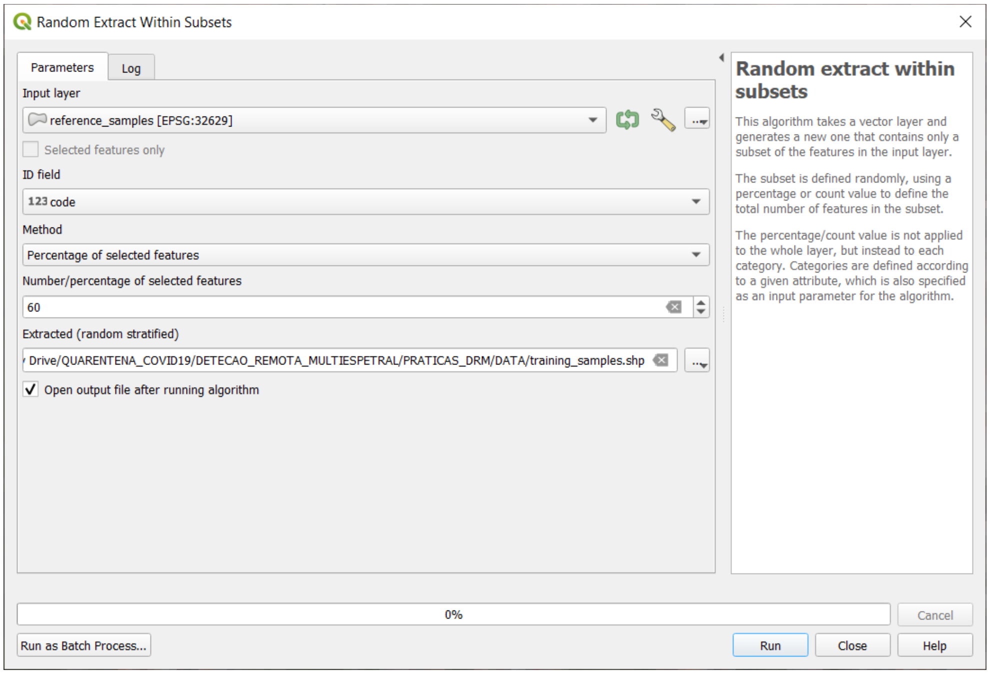
- polygons: select pixels whose center falls inside the polygon;
- lines: select pixels intersecting the line;
- points: select closest pixel to the provided point.

These training geometries must be, previously, created by the user in ArcMap or QGIS, adopting the reference system of the image to be classified. For this exercise, use the polygon vector file already created (named *reference_samples.shp*) which contains 36 samples: 6 for Built-up Areas (Class 1); 7 for Agriculture (Class 2); 6 for Forest (Class 3); 7 for Bushes and spontaneous herbaceous vegetation (Class 4); 6 for Surfaces without vegetation (Class 5); and 4 for Water and wetlands (Class 6).

These samples are collected by digitizing, on screen, polygons corresponding to those land use classes, having the images in the background. Each polygon should contain the same spectral characteristics, and the polygons defined for a certain class must represent all the spectral variability of that class! *Add a field, named "code" for example, to the attribute table to add an integer value corresponding to each class. The values in this field shall be cast into integers. Only geometries with an integer field will be considered by OTB.* Note that, this is a very simplistic example just for you to know how this works! In a real application, the number of reference samples should be much higher, especially when working with machine-learning classification algorithms.

Using the QGIS tool **Vector Selection > Random extract within subsets**, generate a new vector layer (name this file as *training_samples.shp*) containing only a random subset of the features in the original layer, using a percentage (in this case, 60%), for each class (in this case, using the attribute "code" with the number corresponding to each class).

Using the QGIS tool **Vector overlay > Difference**, extract features from the input layer (reference_samples.shp) that fall outside features in the overlay layer (trainining_samples.shp), to generate a new shapefile with the remaining 40% of the polygons in the original shapefile (name it *validation_samples.shp*).


Random Extract Within Subsets

Input layer: reference_samples [EPSG:32629]

□ Selected features only

ID field: 123 code

Method: Percentage of selected features

Number/percentage of selected features: 60

Extracted (random stratified): Drive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/training_samples.shp

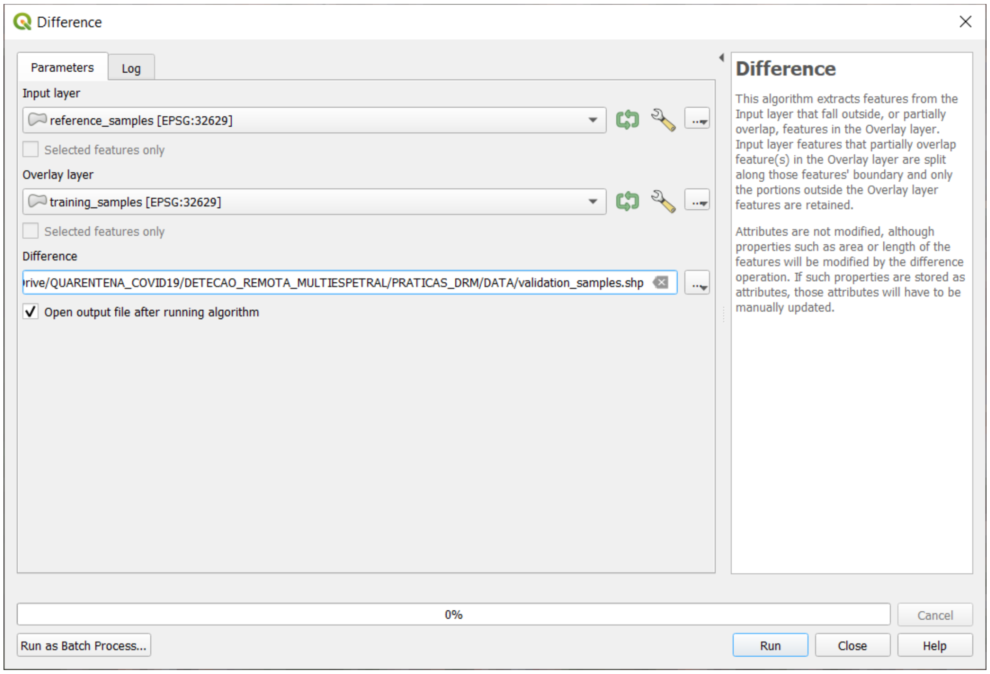
☑ Open output file after running algorithm

Random extract within subsets

This algorithm takes a vector layer and generates a new one that contains only a subset of the features in the input layer.

The subset is defined randomly, using a percentage or count value to define the total number of features in the subset.

The percentage/count value is not applied to the whole layer, but instead to each category. Categories are defined according to a given attribute, which is also specified as an input parameter for the algorithm.


Difference

Input layer: reference_samples [EPSG:32629]

□ Selected features only

Overlay layer: training_samples [EPSG:32629]

□ Selected features only

Difference: rive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/validation_samples.shp

☑ Open output file after running algorithm

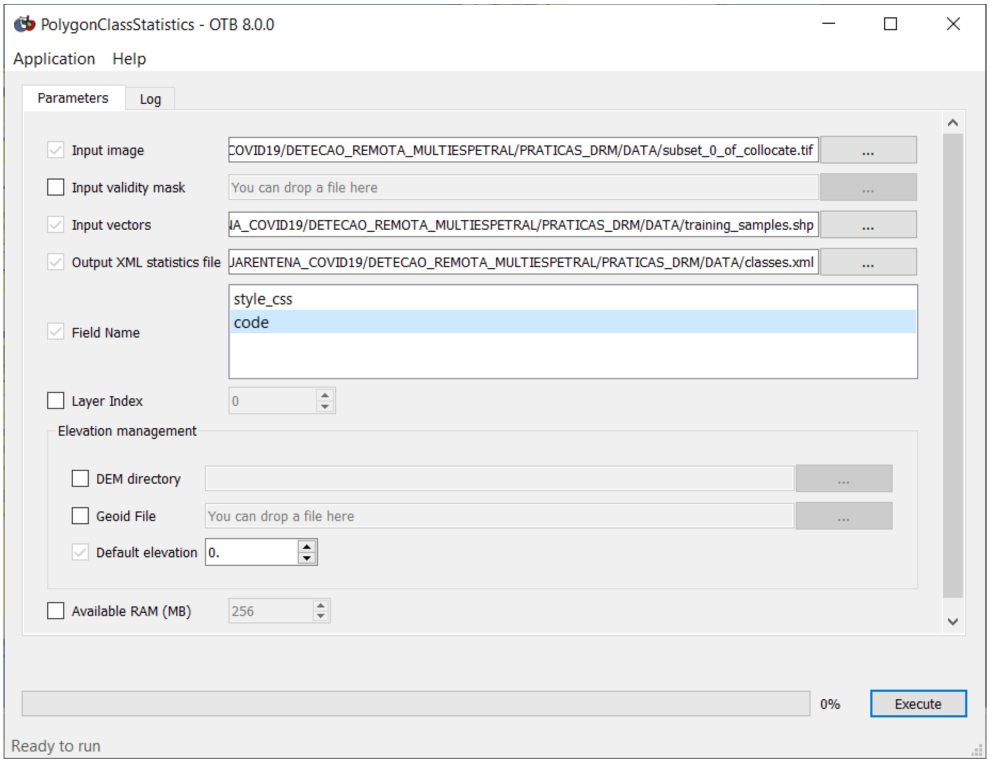
Difference

This algorithm extracts features from the Input layer that fall outside, or partially overlap, features in the Overlay layer. Input layer features that partially overlap feature(s) in the Overlay layer are split along those features' boundary and only the portions outside the Overlay layer features are retained.

Attributes are not modified, although properties such as area or length of the features will be modified by the difference operation. If such properties are stored as attributes, those attributes will have to be manually updated.

The application will require the input image, but it is only used to define the footprint in which samples will be selected. The user can also provide a raster mask, that will be used to discard pixel positions, using parameter `-mask`.

The `-field` parameter is the name of the field that corresponds to class labels in the input geometries (in this case, the attribute "code"). *As mentioned before, the values in this field shall be cast into integers.*



☑ Input image COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/subset_0_of_collocate.tif

□ Input validity mask You can drop a file here

☑ Input vectors IA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/training_samples.shp

☑ Output XML statistics file JARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/classes.xml

☑ Field Name style_css code

□ Layer Index 0

Elevation management

□ DEM directory

□ Geoid File You can drop a file here

☑ Default elevation 0.

□ Available RAM (MB) 256

0%

Ready to run

The Output XML statistics file (here named *class.xml*) looks like this (use WordPad to open the file):

```
<?xml version="1.0" ?>
<GeneralStatistics>
    <Statistic name="samplesPerClass">
        <StatisticMap key="1" value="579" />
        <StatisticMap key="2" value="378" />
        <StatisticMap key="3" value="2339" />
        <StatisticMap key="4" value="898" />
        <StatisticMap key="5" value="484" />
        <StatisticMap key="6" value="1897" />
    </Statistic>
    <Statistic name="samplesPerVector">
        <StatisticMap key="0" value="71" />
        <StatisticMap key="1" value="42" />
        <StatisticMap key="10" value="581" />
        <StatisticMap key="11" value="302" />
        <StatisticMap key="12" value="208" />
        <StatisticMap key="14" value="628" />
        <StatisticMap key="15" value="62" />
        <StatisticMap key="16" value="32" />
        <StatisticMap key="17" value="103" />
        <StatisticMap key="18" value="28" />
        <StatisticMap key="19" value="321" />
        <StatisticMap key="2" value="47" />
        <StatisticMap key="22" value="1012" />
        <StatisticMap key="23" value="885" />
        <StatisticMap key="3" value="218" />
        <StatisticMap key="4" value="88" />
        <StatisticMap key="5" value="132" />
        <StatisticMap key="6" value="168" />
        <StatisticMap key="7" value="191" />
        <StatisticMap key="8" value="641" />
        <StatisticMap key="9" value="815" />
    </Statistic>
</GeneralStatistics>
```

STEP 2: SELECT SAMPLES POSITIONS FOR EACH IMAGE (SampleSelection)

In OTB, use **Learning > SampleSelection**[18] to select samples from a training vector data set. The application selects a set of samples from geometries intended for training (they should have a field giving the associated class).

First of all, the geometries must have been previously analyzed by the PolygonClassStatistics application to compute statistics about the geometries, which are summarized in an XML file. Then, this XML file must be given as an input to this application (Input Statistics).

Now, we know exactly how many samples are available in the image for each class and each geometry in the training set. From these statistics, we can now compute the sampling rates to apply for each class and perform the sample selection. This will be done by the SampleSelection application.

There are several strategies to compute those sampling rates:

- **Constant strategy:** All classes will be sampled with the same number of samples, which is user-defined.
- **Smallest class strategy:** The class with the least number of samples will be fully sampled. All other classes will be sampled with the same number of samples.
- **Percent strategy:** Each class will be sampled with a user-defined percentage (same value for all classes) of samples available in this class.
- **Total strategy:** A global number of samples to select is divided proportionally among each class (class proportions are enforced).
- **Take all strategy:** Take all the available samples.
- **By class strategy:** Set a target number of samples for each class. The number of samples for each class is read from a CSV file.

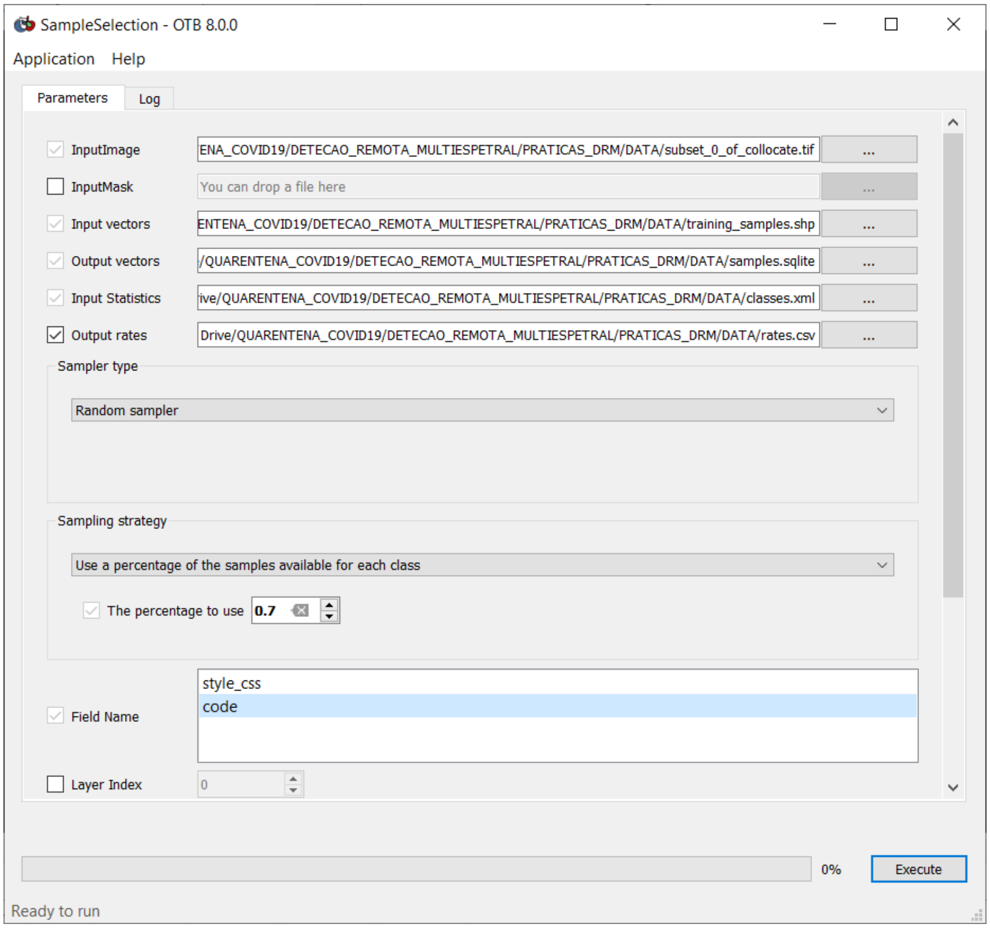
To select the sample positions, there are two available sampling techniques:

- **Random:** Randomly select samples while respecting the sampling rate.
- **Periodic:** Sample periodically using the sampling rate.

The application will make sure that samples span the whole training set extent by adjusting the sampling rate. Depending on the strategy to determine the sampling rate, some geometries of the training set may not be sampled.

[18] https://www.orfeo-toolbox.org/CookBook/Applications/app_SampleSelection.html

The application will accept as input the input image and training geometries, as well as the class statistics XML file computed during the previous step. It will output a vector file containing point geometries that indicate the location of the samples.



☑ InputImage: ENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/subset_0_of_collocate.tif

□ InputMask: You can drop a file here

☑ Input vectors: ENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/training_samples.shp

☑ Output vectors: /QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/samples.sqlite

☑ Input Statistics: ive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/classes.xml

☑ Output rates: Drive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/rates.csv

Sampler type: Random sampler

Sampling strategy: Use a percentage of the samples available for each class

☑ The percentage to use: 0.7

☑ Field Name: style_css, code

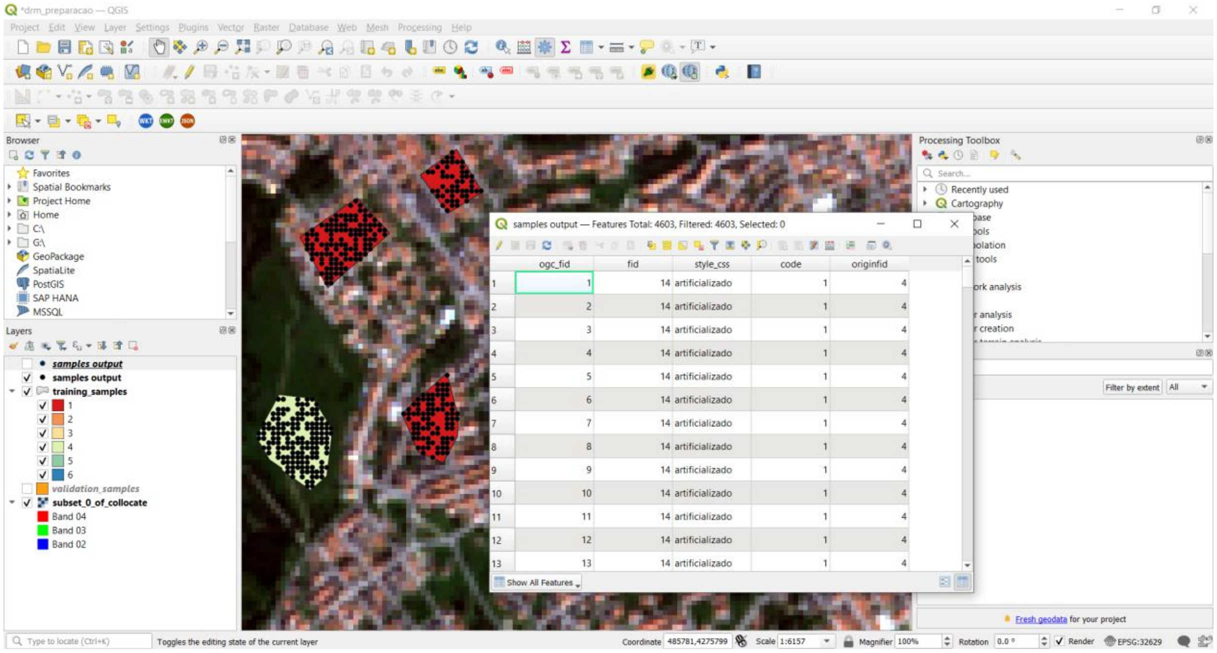
□ Layer Index: 0

Ready to run

The Output rates file (here named *rates.csv*) written by the optional -outrates parameter sums-up what has been done during sample selection (use WordPad to open the file).

```
#className requiredSamples totalSamples rate
1       405     579     0.7
2       265     378     0.7
3       1637    2339    0.7
4       629     898     0.7
5       339     484     0.7
6       1328    1897    0.7
```

The Output vectors file (here named *samples.sqlite*) might be viewed using QGIS (open it as a vector file). The black dots show the samples that have been selected.



samples output — Features Total: 4603, Filtered: 4603, Selected: 0

	ogc_fid	fid	style_css	code	originfid
1	1	14	artificializado	1	4
2	2	14	artificializado	1	4
3	3	14	artificializado	1	4
4	4	14	artificializado	1	4
5	5	14	artificializado	1	4
6	6	14	artificializado	1	4
7	7	14	artificializado	1	4
8	8	14	artificializado	1	4
9	9	14	artificializado	1	4
10	10	14	artificializado	1	4
11	11	14	artificializado	1	4
12	12	14	artificializado	1	4
13	13	14	artificializado	1	4

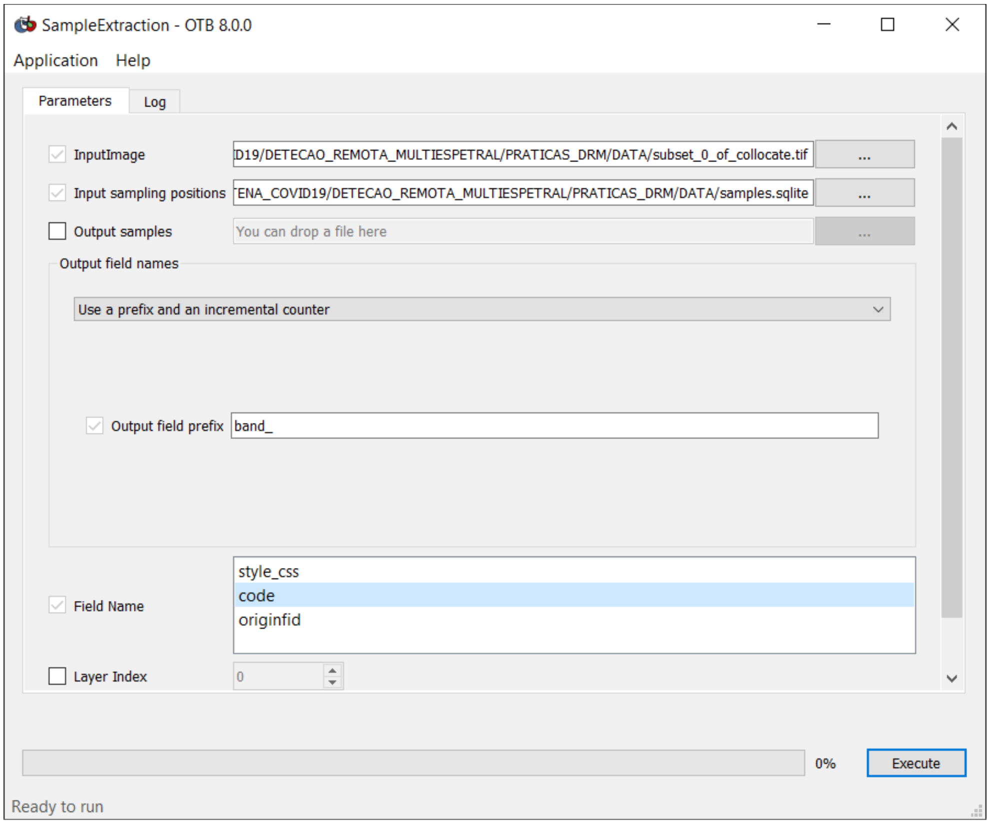
STEP 3: EXTRACT SAMPLES MEASUREMENTS FOR EACH IMAGE (SampleExtraction)

In OTB, use **Learning > SampleExtration**[19] to extract samples values from an image. The application extracts samples values from an image using positions contained in a vector data file.

Now that the locations of the samples are selected, we will attach measurements to them. This is the purpose of the SampleExtraction application. It will walk through the list of samples and extract the underlying pixel values. If no -out parameter is given, the SampleExtraction application can work in update mode (updates the samples file – samples.sqlite – created in the previous step), thus allowing to extract features from multiple images of the same location.

Features will be stored in fields attached to each sample. Field name can be generated from a prefix a sequence of numbers (i.e. if prefix is feature_ then features will be named feature_0, feature_1, …). This can be achieved with the -outfield prefix option. Alternatively, one can set explicit names for all features using the -outfield list option.

[19] https://www.orfeo-toolbox.org/CookBook/Applications/app_SampleExtraction.html



SampleExtraction - OTB 8.0.0

☑ InputImage: D19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/subset_0_of_collocate.tif

☑ Input sampling positions: ENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/samples.sqlite

□ Output samples: You can drop a file here

Output field names: Use a prefix and an incremental counter

☑ Output field prefix: band_

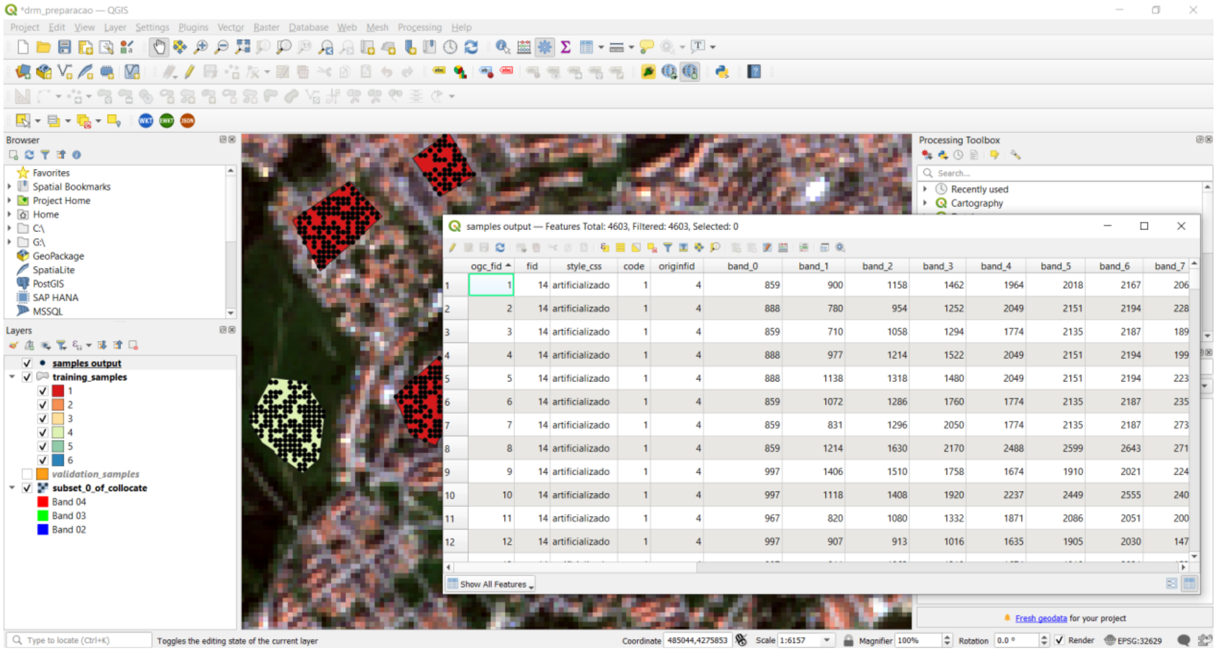
☑ Field Name: style_css / code (selected) / originfid

□ Layer Index: 0

0%

Ready to run

The updated samples file attributes table has as many more columns as the number of bands in the image (in this case, 26 bands):



*drm_preparacao — QGIS

samples output — Features Total: 4603, Filtered: 4603, Selected: 0

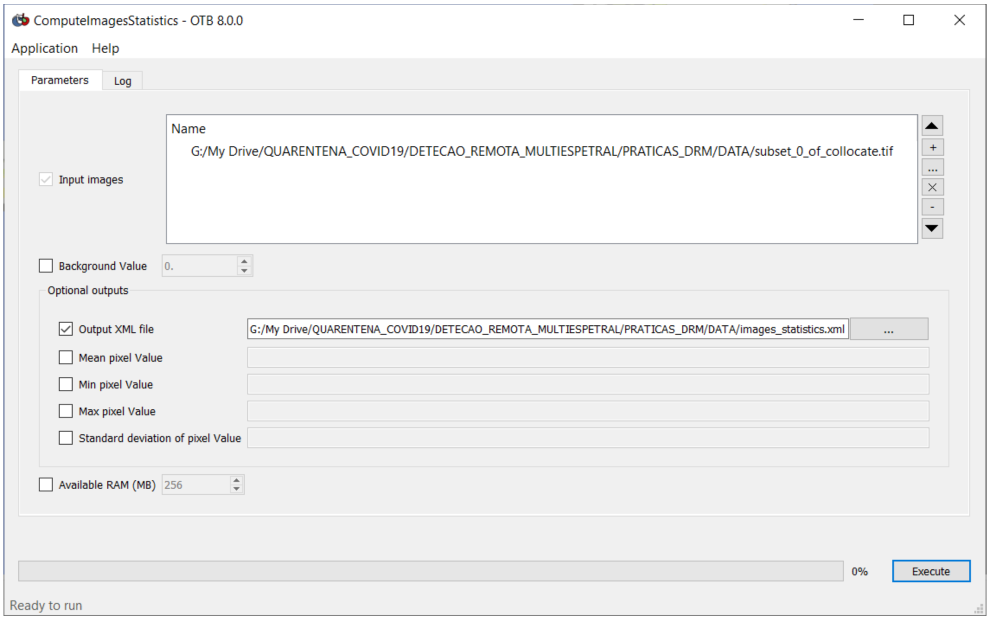
	ogc_fid	fid	style_css	code	originfid	band_0	band_1	band_2	band_3	band_4	band_5	band_6	band_7
1	1	14	artificializado	1	4	859	900	1158	1462	1964	2018	2167	206
2	2	14	artificializado	1	4	888	780	954	1252	2049	2151	2194	228
3	3	14	artificializado	1	4	859	710	1058	1294	1774	2135	2187	189
4	4	14	artificializado	1	4	888	977	1214	1522	2049	2151	2194	199
5	5	14	artificializado	1	4	888	1138	1318	1480	2049	2151	2194	223
6	6	14	artificializado	1	4	859	1072	1286	1760	1774	2135	2187	235
7	7	14	artificializado	1	4	859	831	1296	2050	1774	2135	2187	273
8	8	14	artificializado	1	4	859	1214	1630	2170	2488	2599	2643	271
9	9	14	artificializado	1	4	997	1406	1510	1758	1674	1910	2021	224
10	10	14	artificializado	1	4	997	1118	1408	1920	2237	2449	2555	240
11	11	14	artificializado	1	4	967	820	1080	1332	1871	2086	2051	200
12	12	14	artificializado	1	4	997	907	913	1016	1635	1905	2030	147

STEP 4: COMPUTE IMAGES STATISTICS (ComputeImagesStatistics)

In OTB, use **Learning > ComputeImageStatistics**[20] to compute the global mean and standard deviation for each band from a set of images and optionally saves the results in an XML file. Each image of the set must contain the same bands as the others (i.e. same types, in the same order).

The output XML is intended to be used as an input for the TrainImagesClassifier application to normalize samples before learning. You can also normalize the image with the XML file in the ImageClassifier application.

Some machine learning algorithms converge faster if the range of features is [-1, 1] or [0, 1]. Others will be sensitive to relative ranges between features, e.g. a feature with a larger range might have more weight in the final decision. This is for instance the case for machine learning algorithms using euclidean distance at some point to compare features (such as the Support Vector Machines, SVM[21]). In those cases, it is advised to normalize all features to the range [-1, 1] before performing the learning. For this purpose, the ComputeImageStatistics application allows to compute and output to an XML file the mean and standard deviation based on the pooled variance of each band for one or several images.



☑ Input images

Name
G:/My Drive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/subset_0_of_collocate.tif

□ Background Value 0.

Optional outputs

☑ Output XML file G:/My Drive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/images_statistics.xml

□ Mean pixel Value

□ Min pixel Value

□ Max pixel Value

□ Standard deviation of pixel Value

□ Available RAM (MB) 256

0%

Ready to run

The Output statistics file (here named *images_statistics.xml*) can then be fed to the training and classification applications (use WordPad to open the file).

[20] https://www.orfeo-toolbox.org/CookBook/Applications/app_ComputeImagesStatistics.html

[21] https://medium.com/@meritshot/standardization-v-s-normalization-6f93225fbd84

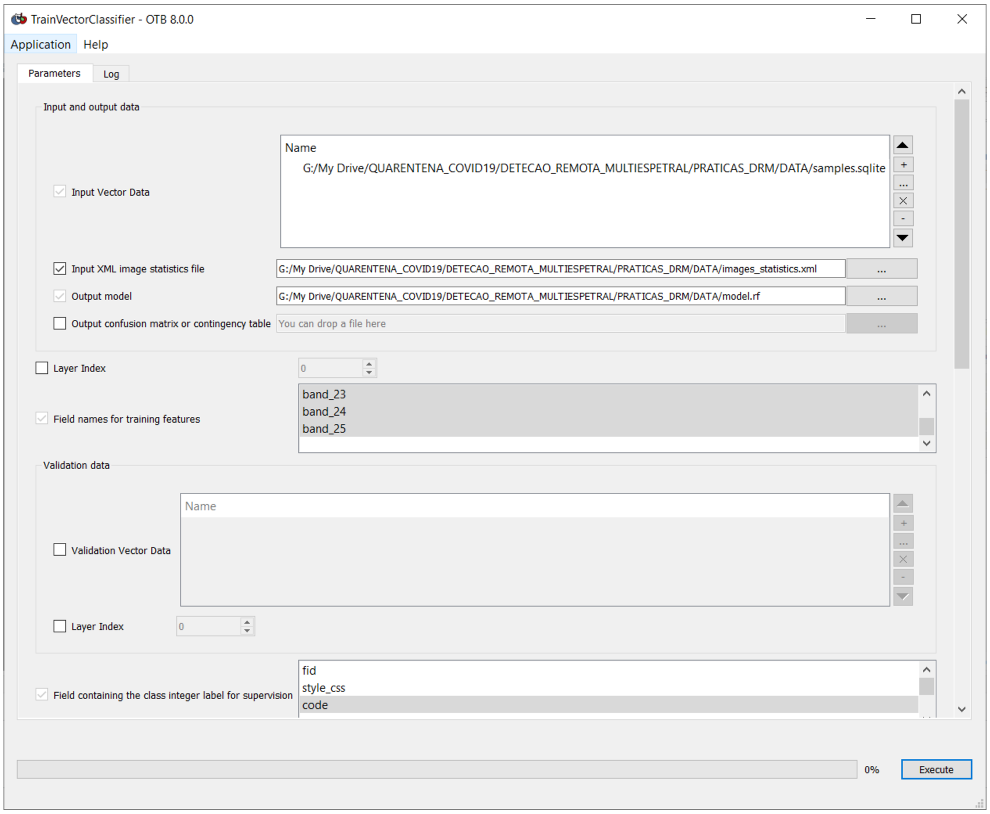
```
<?xml version="1.0" ?>
<FeatureStatistics>
    <Statistic name="mean">
        <StatisticVector value="641.594" />
        <StatisticVector value="688.629" />
        <StatisticVector value="914.817" />
        <StatisticVector value="889.033" />
        <StatisticVector value="1171.85" />
        <StatisticVector value="1708.58" />
        <StatisticVector value="1863.41" />
        <StatisticVector value="1963.26" />
        <StatisticVector value="1977.88" />
        <StatisticVector value="2023.28" />
        <StatisticVector value="1594.05" />
        <StatisticVector value="1203.18" />
        <StatisticVector value="2.62903" />
        <StatisticVector value="564.895" />
        <StatisticVector value="722.768" />
        <StatisticVector value="946.704" />
        <StatisticVector value="1084.5" />
        <StatisticVector value="1331.47" />
        <StatisticVector value="1660.54" />
        <StatisticVector value="1815.59" />
        <StatisticVector value="1919.3" />
        <StatisticVector value="1957.28" />
        <StatisticVector value="1955.03" />
        <StatisticVector value="2084.22" />
        <StatisticVector value="1632.36" />
        <StatisticVector value="1.95031" />
    </Statistic>
    <Statistic name="min">
        <StatisticVector value="105" />
        <StatisticVector value="19" />
        <StatisticVector value="102" />
        <StatisticVector value="52" />
        <StatisticVector value="1" />
        <StatisticVector value="1" />
        <StatisticVector value="1" />
        <StatisticVector value="1" />
         ...

    <Statistic name="stddev">
        <StatisticVector value="368.275" />
        <StatisticVector value="527.229" />
        <StatisticVector value="618.283" />
        <StatisticVector value="809.745" />
        <StatisticVector value="891.29" />
        <StatisticVector value="1228.62" />
        <StatisticVector value="1345.35" />
        <StatisticVector value="1447.53" />
        <StatisticVector value="1435.57" />
        <StatisticVector value="1396.4" />
        <StatisticVector value="1246.89" />
        <StatisticVector value="1040.09" />
        <StatisticVector value="2.57724" />
        <StatisticVector value="450.878" />
        <StatisticVector value="640.937" />
        <StatisticVector value="787.111" />
        <StatisticVector value="1011.39" />
        <StatisticVector value="1092.76" />
        <StatisticVector value="1263.84" />
        <StatisticVector value="1361.8" />
        <StatisticVector value="1456.61" />
        <StatisticVector value="1462.56" />
        <StatisticVector value="1440.06" />
        <StatisticVector value="1622.09" />
        <StatisticVector value="1349.36" />
        <StatisticVector value="2.75754" />
    </Statistic>
</FeatureStatistics>
```

STEP 5: TRAIN THE MACHINE LEARNING MODEL (TrainVectorClassifier)

In OTB, use **Learning > TrainingVectorClassifier**[22] to train a classifier based on labeled geometries and a list of features to consider for classification. This application is based on LibSVM, OpenCV Machine Learning (2.3.1 and later), and Shark ML The output of this application is a text model file, whose format corresponds to the ML model type chosen. There are no image or vector data outputs created.

Now that the training samples are ready, we can perform the learning using the TrainVectorClassifier application. In this case, use the Random Forest algorithm and change the number of trees from 100 to 50 (see input parameters and corresponding default values below).



☑ Input Vector Data
Name
G:/My Drive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/samples.sqlite

☑ Input XML image statistics file: G:/My Drive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/images_statistics.xml

☑ Output model: G:/My Drive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/model.rf

☐ Output confusion matrix or contingency table: You can drop a file here

☐ Layer Index: 0

☑ Field names for training features: band_23, band_24, band_25

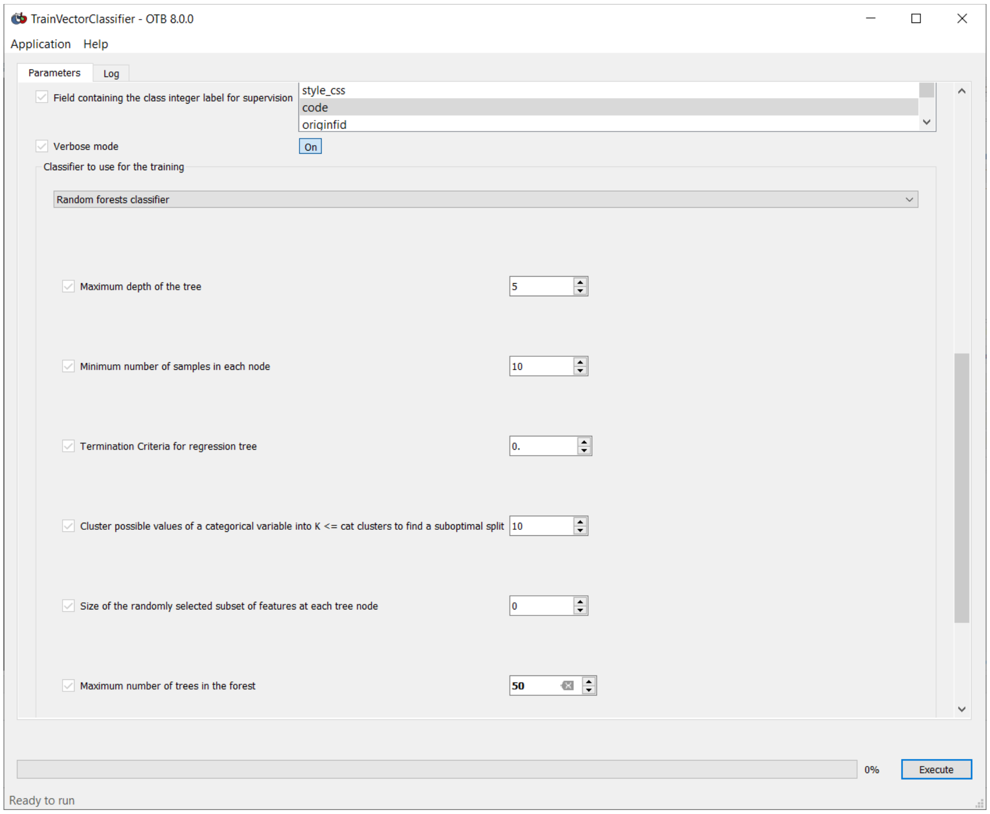
☐ Validation Vector Data
Name

☐ Layer Index: 0

☑ Field containing the class integer label for supervision: fid, style_css, code

…

[22] https://www.orfeo-toolbox.org/CookBook/Applications/app_TrainVectorClassifier.html



☑ Field containing the class integer label for supervision: style_css / **code** / originfid

☑ Verbose mode: On

Classifier to use for the training: Random forests classifier

☑ Maximum depth of the tree: 5

☑ Minimum number of samples in each node: 10

☑ Termination Criteria for regression tree: 0.

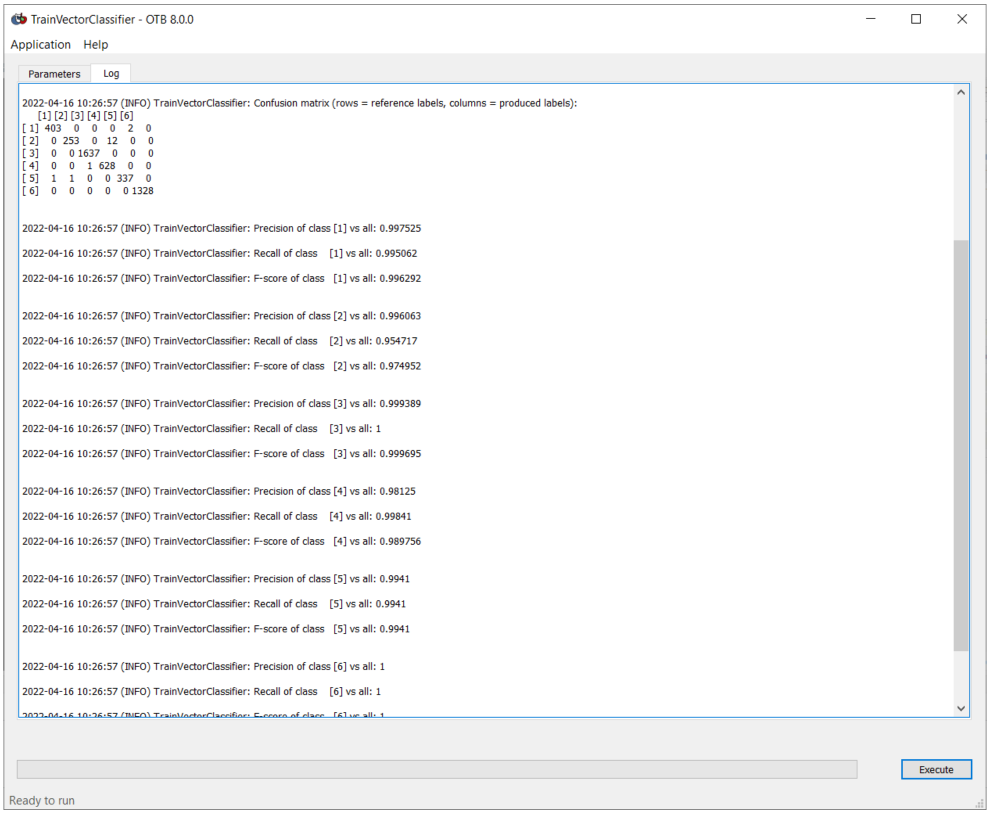
☑ Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal split: 10

☑ Size of the randomly selected subset of features at each tree node: 0

☑ Maximum number of trees in the forest: 50

0%

Ready to run



```
2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Confusion matrix (rows = reference labels, columns = produced labels):
   [1] [2] [3] [4] [5] [6]
[ 1]  403    0    0    0    2    0
[ 2]    0  253    0   12    0    0
[ 3]    0    0 1637    0    0    0
[ 4]    0    0    1  628    0    0
[ 5]    1    1    0    0  337    0
[ 6]    0    0    0    0    0 1328


2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Precision of class [1] vs all: 0.997525

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Recall of class    [1] vs all: 0.995062

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: F-score of class   [1] vs all: 0.996292


2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Precision of class [2] vs all: 0.996063

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Recall of class    [2] vs all: 0.954717

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: F-score of class   [2] vs all: 0.974952


2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Precision of class [3] vs all: 0.999389

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Recall of class    [3] vs all: 1

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: F-score of class   [3] vs all: 0.999695


2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Precision of class [4] vs all: 0.98125

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Recall of class    [4] vs all: 0.99841

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: F-score of class   [4] vs all: 0.989756


2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Precision of class [5] vs all: 0.9941

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Recall of class    [5] vs all: 0.9941

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: F-score of class   [5] vs all: 0.9941


2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Precision of class [6] vs all: 1

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: Recall of class    [6] vs all: 1

2022-04-16 10:26:57 (INFO) TrainVectorClassifier: F-score of class   [6] vs all: 1
```

Ready to run

RANDOM FOREST CLASSIFIER OPTIONS (default values)

(more details in https://docs.opencv.org/2.4/modules/ml/doc/random_trees.html)

Maximum depth of the tree -classifier.rf.max int DEFAULT VALUE: 5

The depth of the tree. A low value will likely underfit and conversely a high value will likely overfit. The optimal value can be obtained using cross-validation or other suitable methods.

Minimum number of samples in each node -classifier.rf.min int DEFAULT VALUE: 10

If the number of samples in a node is smaller than this parameter, then the node will not be split. A reasonable value is a small percentage of the total data e.g. 1 percent.

Termination Criteria for regression tree -classifier.rf.ra float DEFAULT VALUE: 0

If all absolute differences between an estimated value in a node and the values of the train samples in this node are smaller than this regression accuracy parameter, then the node will not be split.

Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal split -classifier.rf.cat int DEFAULT VALUE: 10
Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal split.

Size of the randomly selected subset of features at each tree node -classifier.rf.var int DEFAULT VALUE: 0

The size of the subset of features, randomly selected at each tree node, that are used to find the best split(s). If you set it to 0, then the size will be set to the square root of the total number of features.

Maximum number of trees in the forest -classifier.rf.nbtrees int DEFAULT VALUE: 100

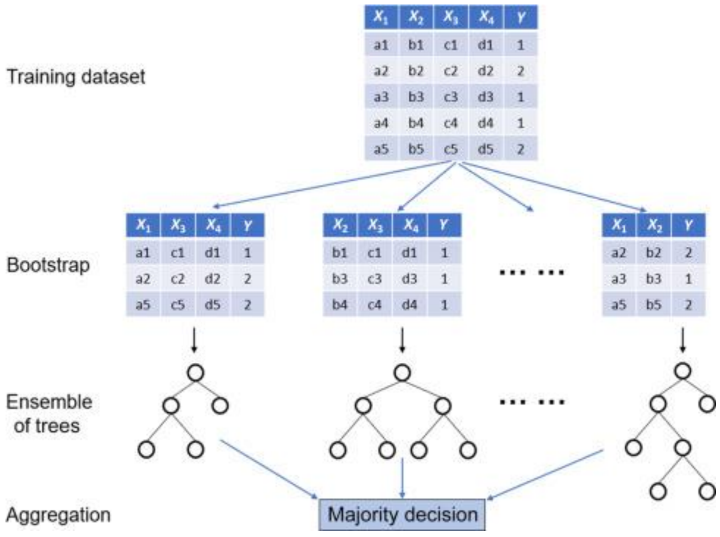
The maximum number of trees in the forest. Typically, the more trees you have, the better the accuracy. However, the improvement in accuracy generally diminishes and reaches an asymptote for a certain number of trees. Also keep in mind, that increasing the number of trees increases the prediction time linearly.

Sufficient accuracy (OOB error) -classifier.rf.acc float DEFAULT VALUE: 0.01

Sufficient accuracy (OOB error).

https://towardsdatascience.com/what-is-out-of-bag-oob-score-in-random-forest-a7fa23d710

https://www.sciencedirect.com/topics/engineering/random-forest



X_1	X_2	X_3	X_4	Y
a1	b1	c1	d1	1
a2	b2	c2	d2	2
a3	b3	c3	d3	1
a4	b4	c4	d4	1
a5	b5	c5	d5	2

X_1	X_3	X_4	Y
a1	c1	d1	1
a2	c2	d2	2
a5	c5	d5	2

X_2	X_3	X_4	Y
b1	c1	d1	1
b3	c3	d3	1
b4	c4	d4	1

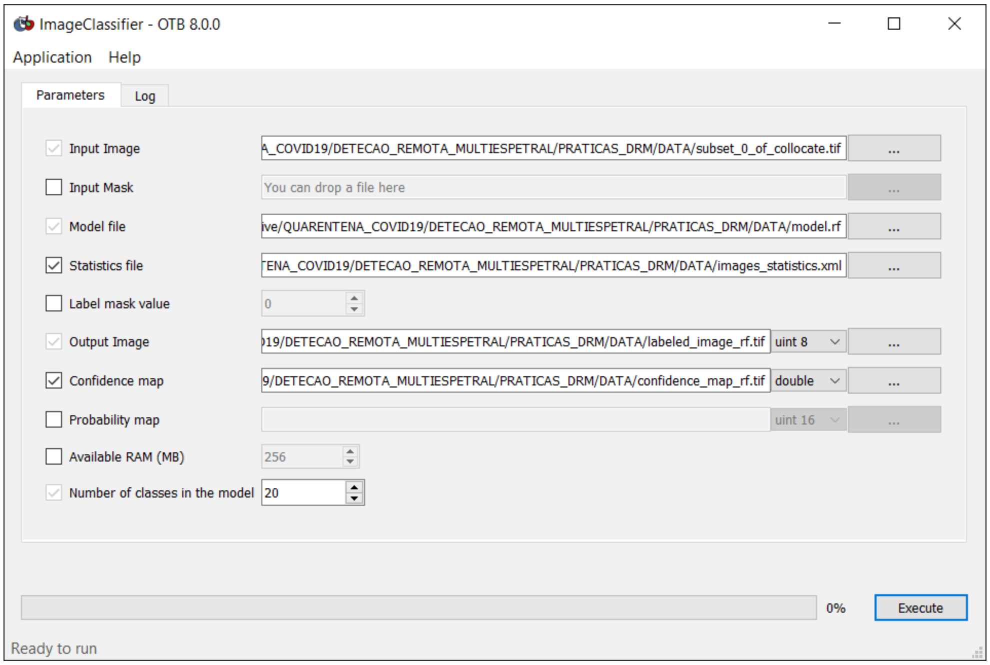
X_1	X_2	Y
a2	b2	2
a3	b3	1
a5	b5	2

STEP 6: PERFORM THE CLASSIFICATION BY APPLYING THE MODEL (ImageClassifier)

In OTB, use **Learning > ImageClassifier**[23] to perform a classification of the input image according to a model file. This application performs an image classification based on a model file produced by the TrainImagesClassifier or the TrainVectorClassifier application. Pixels of the output image will contain the class labels decided by the classifier (maximal class label = 65535). The input pixels can be optionally centered and reduced according to the statistics file produced by the ComputeImagesStatistics application.

Once the classifier has been trained, one can apply the model to classify pixels inside defined classes on a new image using the ImageClassifier application. A confidence map of the produced classification might be produced. The confidence index depends on the model: in the case of Random Forests, the confidence is the proportion of votes for the majority class.

Note that, the probability map is only implemented for the Shark Random Forest classifier at this point.



☑ Input Image: A_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/subset_0_of_collocate.tif

☐ Input Mask: You can drop a file here

☑ Model file: ive/QUARENTENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/model.rf

☑ Statistics file: TENA_COVID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/images_statistics.xml

☐ Label mask value: 0

☑ Output Image:)19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/labeled_image_rf.tif — uint 8

☑ Confidence map: 9/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/confidence_map_rf.tif — double

☐ Probability map: uint 16

☐ Available RAM (MB): 256

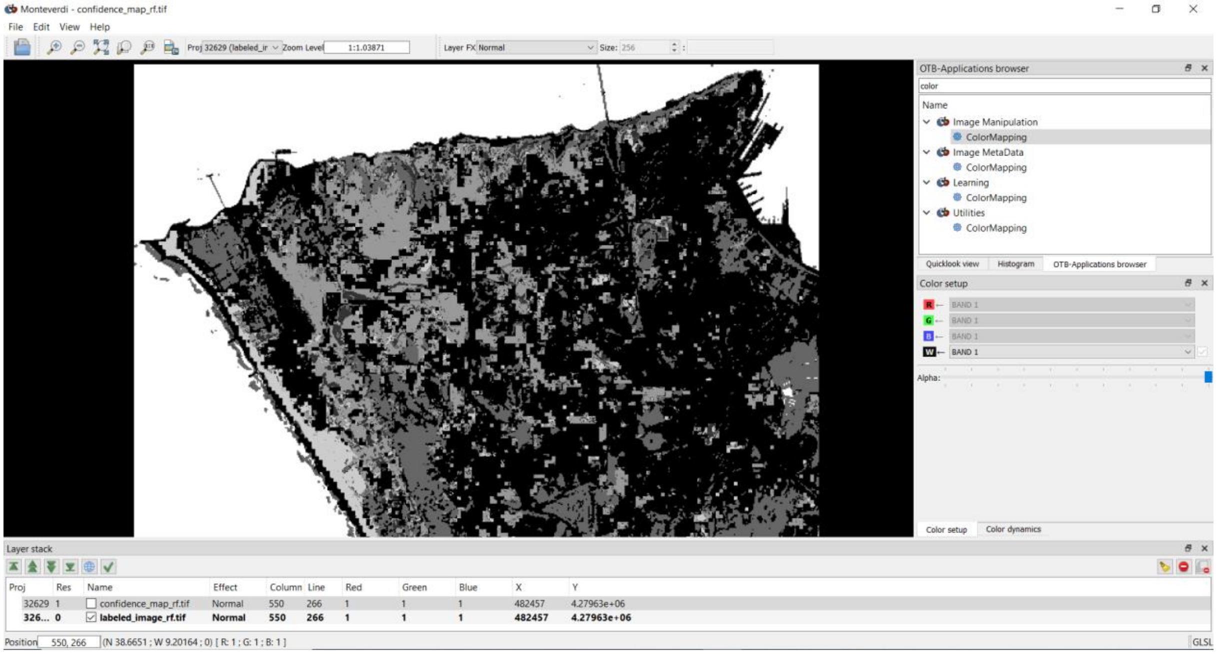
☑ Number of classes in the model: 20

0%

Ready to run

[23] https://www.orfeo-toolbox.org/CookBook/Applications/app_ImageClassifier.html

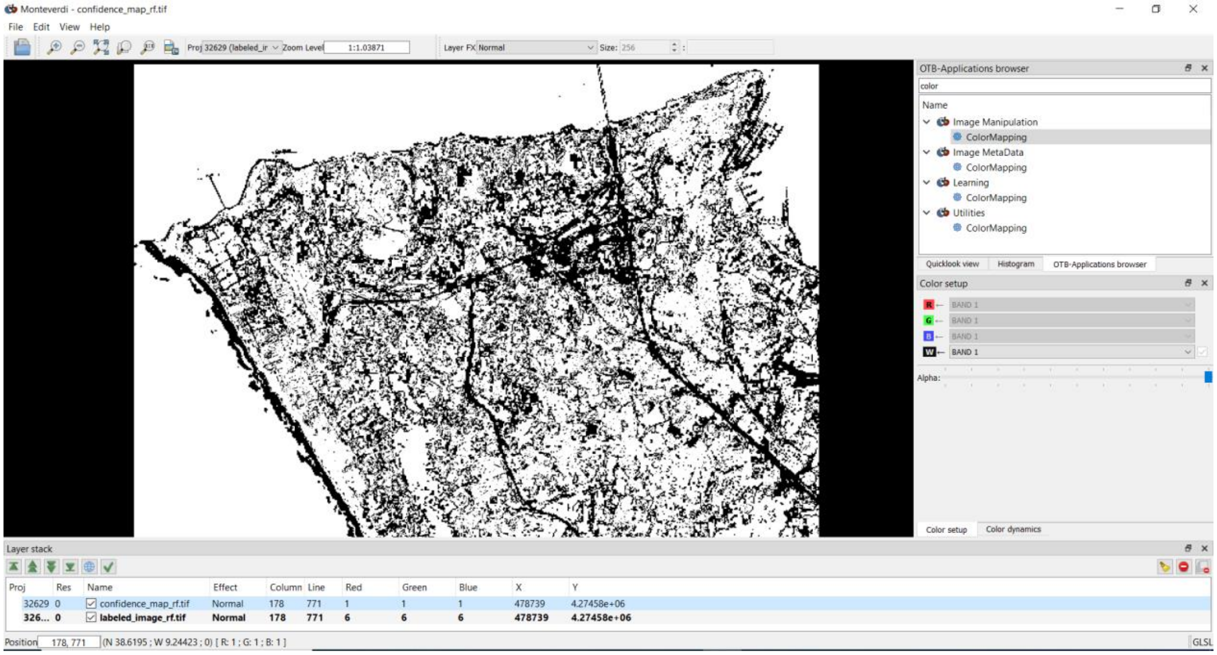
Labeled Image



Proj	Res	Name	Effect	Column	Line	Red	Green	Blue	X	Y
32629	1	□ confidence_map_rf.tif	Normal	550	266	1	1	1	482457	4.27963e+06
326...	**0**	☑ **labeled_image_rf.tif**	**Normal**	**550**	**266**	**1**	**1**	**1**	**482457**	**4.27963e+06**

Position 550, 266 (N 38.6651 ; W 9.20164 ; 0) [R: 1 ; G: 1 ; B: 1]

Confidence map

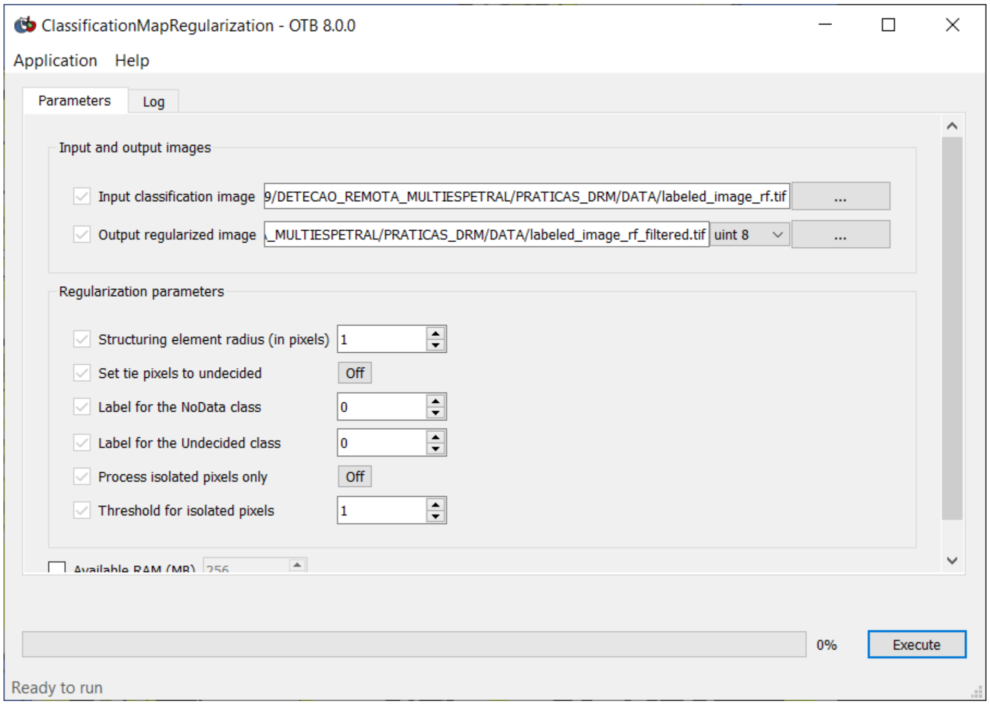


Proj	Res	Name	Effect	Column	Line	Red	Green	Blue	X	Y
32629	0	☑ confidence_map_rf.tif	Normal	178	771	1	1	1	478739	4.27458e+06
326...	**0**	☑ **labeled_image_rf.tif**	**Normal**	**178**	**771**	**6**	**6**	**6**	**478739**	**4.27458e+06**

Position 178, 771 (N 38.6195 ; W 9.24423 ; 0) [R: 1 ; G: 1 ; B: 1]

STEP 7: FILTER THE LABELED IMAGES (ClassificationMapRegularization)

In OTB, use **Learning > ClassificationMapRegularization**[24] to filter the input labeled image (with a maximal class label = 65535) using Majority Voting in a ball-shaped neighborhood. Majority Voting takes the more representative value of all the pixels identified by the ball-shaped structuring element and then sets the center pixel to this majority label value.



ClassificationMapRegularization - OTB 8.0.0

Application Help

Parameters | Log

Input and output images

- ☑ Input classification image: 9/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/labeled_image_rf.tif
- ☑ Output regularized image: _MULTIESPETRAL/PRATICAS_DRM/DATA/labeled_image_rf_filtered.tif, uint 8

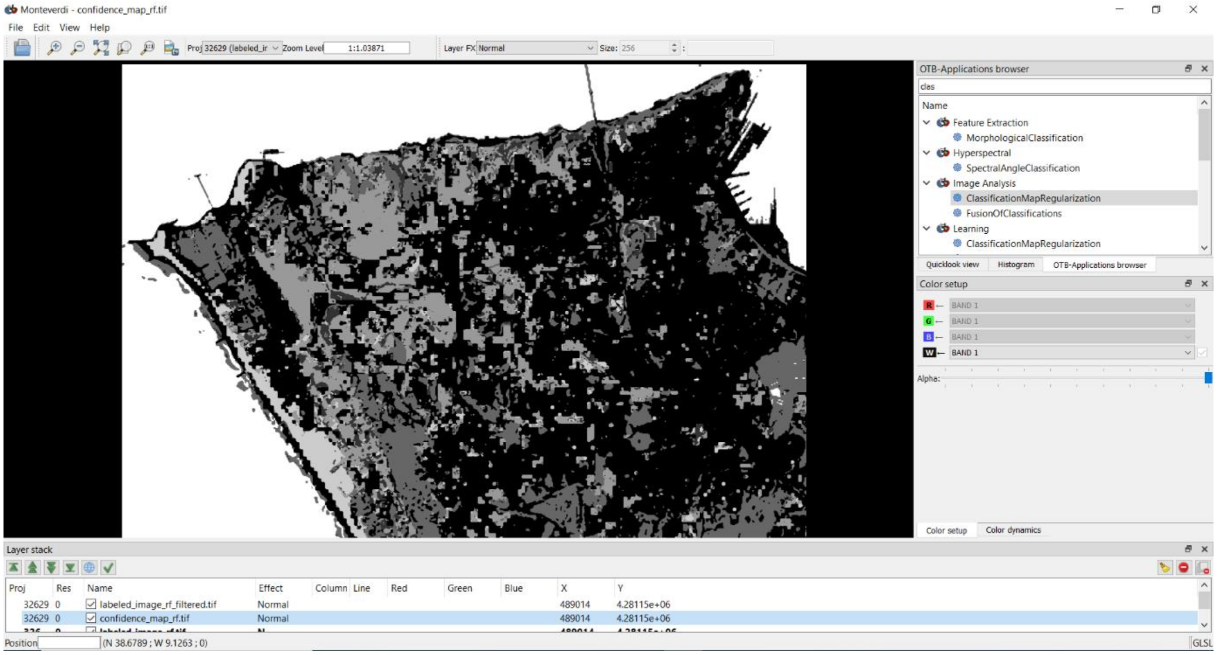
Regularization parameters

- ☑ Structuring element radius (in pixels): 1
- ☑ Set tie pixels to undecided: Off
- ☑ Label for the NoData class: 0
- ☑ Label for the Undecided class: 0
- ☑ Process isolated pixels only: Off
- ☑ Threshold for isolated pixels: 1

□ Available RAM (MB): 256

0% Execute

Ready to run



Monteverdi - confidence_map_rf.tif

File Edit View Help

Proj 32629 (labeled_ir · Zoom Level 1:1.03871 · Layer FX Normal · Size: 256

OTB-Applications browser

clas

Name
- Feature Extraction
  - MorphologicalClassification
- Hyperspectral
  - SpectralAngleClassification
- Image Analysis
  - ClassificationMapRegularization
  - FusionOfClassifications
- Learning
  - ClassificationMapRegularization

Quicklook view | Histogram | OTB-Applications browser

Color setup

- R ← BAND 1
- G ← BAND 1
- B ← BAND 1
- W ← BAND 1

Alpha:

Color setup | Color dynamics

Layer stack

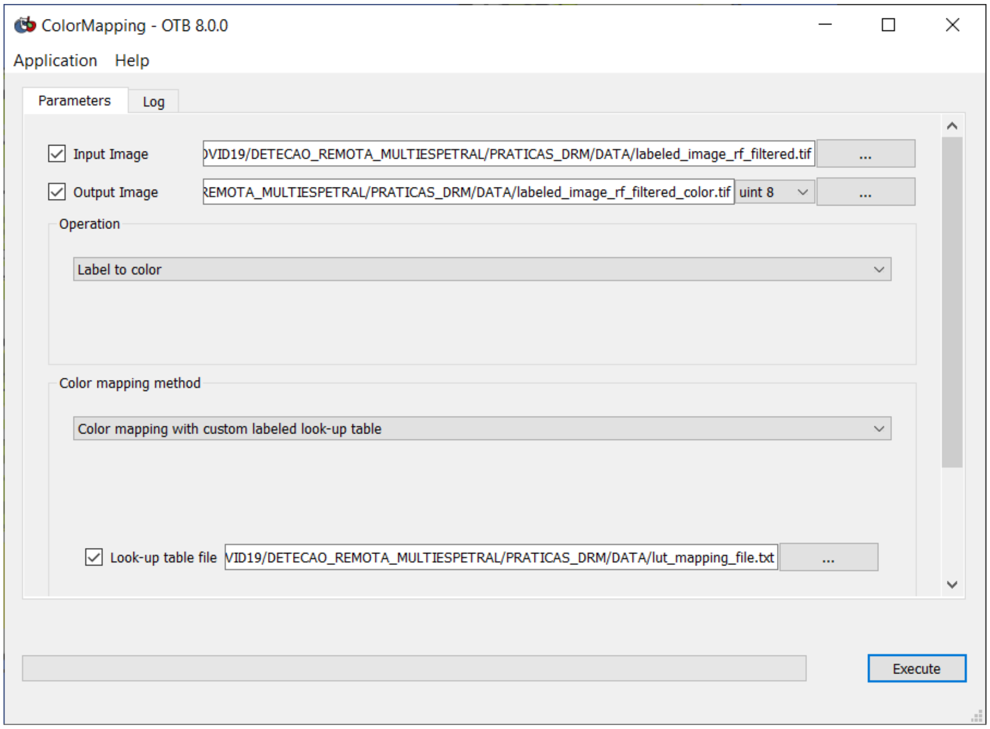
Proj	Res	Name	Effect	Column	Line	Red	Green	Blue	X	Y
32629	0	☑ labeled_image_rf_filtered.tif	Normal						489014	4.28115e+06
32629	0	☑ confidence_map_rf.tif	Normal						489014	4.28115e+06
326	0	☑ labeled_image_rf.tif	N						489014	4.28115e+06

Position (N 38.6789 ; W 9.1263 ; 0) GLSL

[24] https://www.orfeo-toolbox.org/CookBook/Applications/app_ClassificationMapRegularization.html

STEP 8: MAP THE LABELED IMAGE TO AN 8-BITS IMAGE (ColorMapping)

In OTB, use **Learning > ColorMapping**[25] to map a label image to an 8-bits RGB image (both ways) using different methods.



☑ Input Image:)VID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/labeled_image_rf_filtered.tif

☑ Output Image: ΚEMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/labeled_image_rf_filtered_color.tif — uint 8

Operation: Label to color

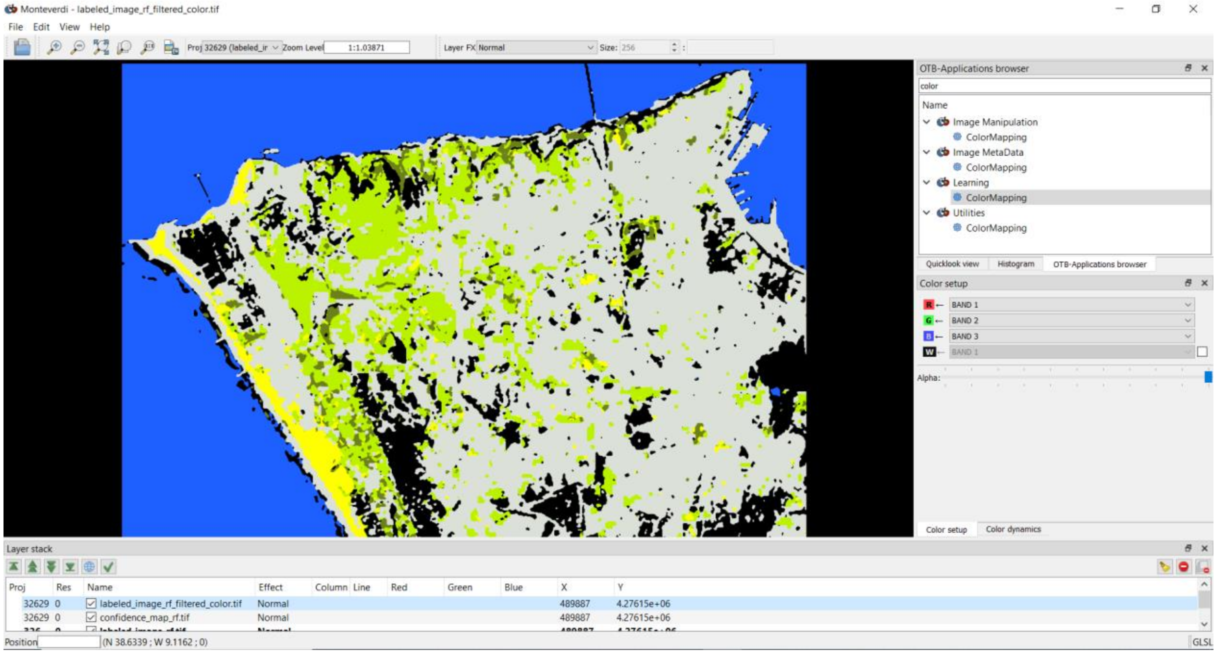
Color mapping method: Color mapping with custom labeled look-up table

☑ Look-up table file: VID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/lut_mapping_file.txt

When using ColorMapping with custom labeled look-up table options, a Look-up table file -method.custom.lut filename [dtype] is required. An ASCII file containing the look-up table with one color per line (for instance the line ‘1 255 0 0’ means that all pixels with label 1 will be replaced by RGB color 255 0 0). Lines beginning with a # are ignored.

```
# Lines beginning with a # are ignored
1 192 192 192
2 96 169 23
3 0 138 0
4 164 196 0
5 227 200 0
6 27 161 226
```

[25] https://www.orfeo-toolbox.org/CookBook/Applications/app_ColorMapping.html



Monteverdi - labeled_image_rf_filtered_color.tif

Proj 32629 (labeled_ir — Zoom Level 1:1.03871 — Layer FX Normal — Size: 256

OTB-Applications browser: color

- Image Manipulation
  - ColorMapping
- Image MetaData
  - ColorMapping
- Learning
  - ColorMapping
- Utilities
  - ColorMapping

Color setup: R ← BAND 1; G ← BAND 2; B ← BAND 3; W ← BAND 1

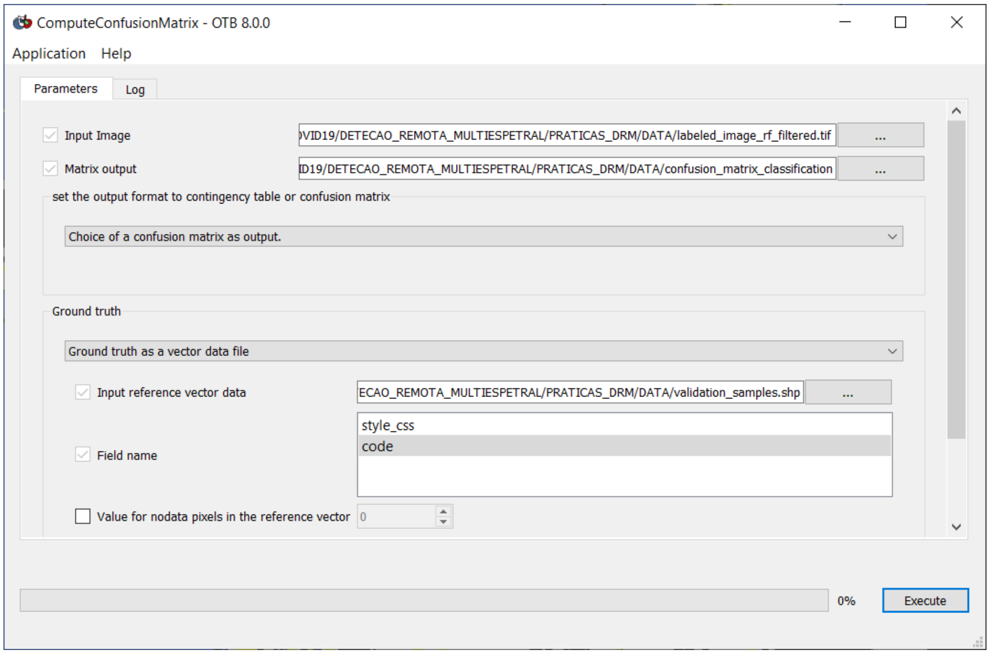
Layer stack

Proj	Res	Name	Effect	Column	Line	Red	Green	Blue	X	Y
32629	0	☑ labeled_image_rf_filtered_color.tif	Normal						489887	4.27615e+06
32629	0	☑ confidence_map_rf.tif	Normal						489887	4.27615e+06

Position (N 38.6339 ; W 9.1162 ; 0)

STEP 9: COMPUTE A CONFUSION MATRIX (ComputeConfusionMatrix)

In OTB, use **Learning > ComputeConfusionMatrix**[26] to compute the confusion matrix for a classification (in this case, a Random Forest classification.



ComputeConfusionMatrix - OTB 8.0.0

☑ Input Image:)VID19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/labeled_image_rf_filtered.tif

☑ Matrix output: D19/DETECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/confusion_matrix_classification

set the output format to contingency table or confusion matrix: Choice of a confusion matrix as output.

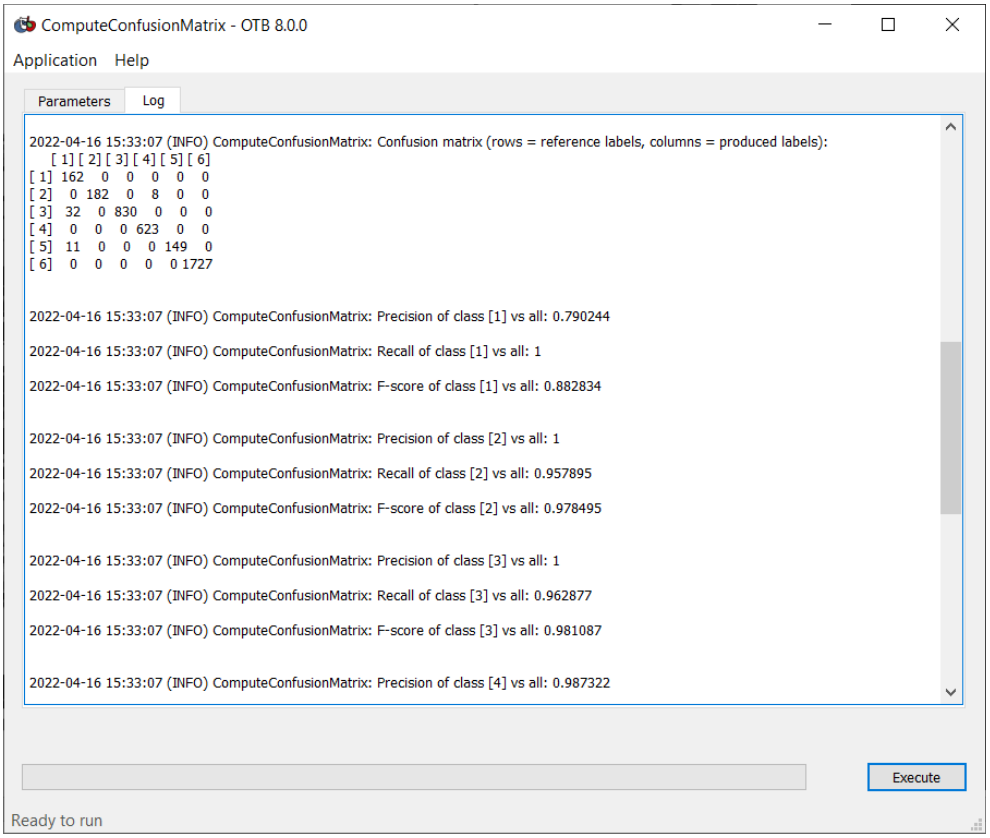
Ground truth: Ground truth as a vector data file

☑ Input reference vector data: ECAO_REMOTA_MULTIESPETRAL/PRATICAS_DRM/DATA/validation_samples.shp

☑ Field name: style_css / code (selected: code)

□ Value for nodata pixels in the reference vector: 0

[26] https://www.orfeo-toolbox.org/CookBook/Applications/app_ComputeConfusionMatrix.html



```
2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Confusion matrix (rows = reference labels, columns = produced labels):
    [ 1] [ 2] [ 3] [ 4] [ 5] [ 6]
[ 1]  162    0    0    0    0    0
[ 2]    0  182    0    8    0    0
[ 3]   32    0  830    0    0    0
[ 4]    0    0    0  623    0    0
[ 5]   11    0    0    0  149    0
[ 6]    0    0    0    0    0 1727


2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Precision of class [1] vs all: 0.790244

2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Recall of class [1] vs all: 1

2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: F-score of class [1] vs all: 0.882834


2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Precision of class [2] vs all: 1

2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Recall of class [2] vs all: 0.957895

2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: F-score of class [2] vs all: 0.978495


2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Precision of class [3] vs all: 1

2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Recall of class [3] vs all: 0.962877

2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: F-score of class [3] vs all: 0.981087


2022-04-16 15:33:07 (INFO) ComputeConfusionMatrix: Precision of class [4] vs all: 0.987322
```

Produced Labels

Reference Data

Classes	**1**	**2**	**3**	**4**	**5**	**6**	**Row Total**	**Recall**	**F1-score**
1	**162**						162	1	0.88
2		**182**		8			190	0.96	0.98
3	32		**830**				862	0.96	0.98
4				**623**			623	1	0.99
5	11				**149**		160	0.93	0.96
6						**1727**	1727	1	1
Column Total	205	182	830	631	149	1727	**3724**		
Precision	0.79	1	1	0.99	1	1			

Overall Accuracy= 0.99

Kappa Coefficient= 0.98