UCLouvain

Land Monitoring by Advanced Earth Observation

LBRAT2104 – 2020-21

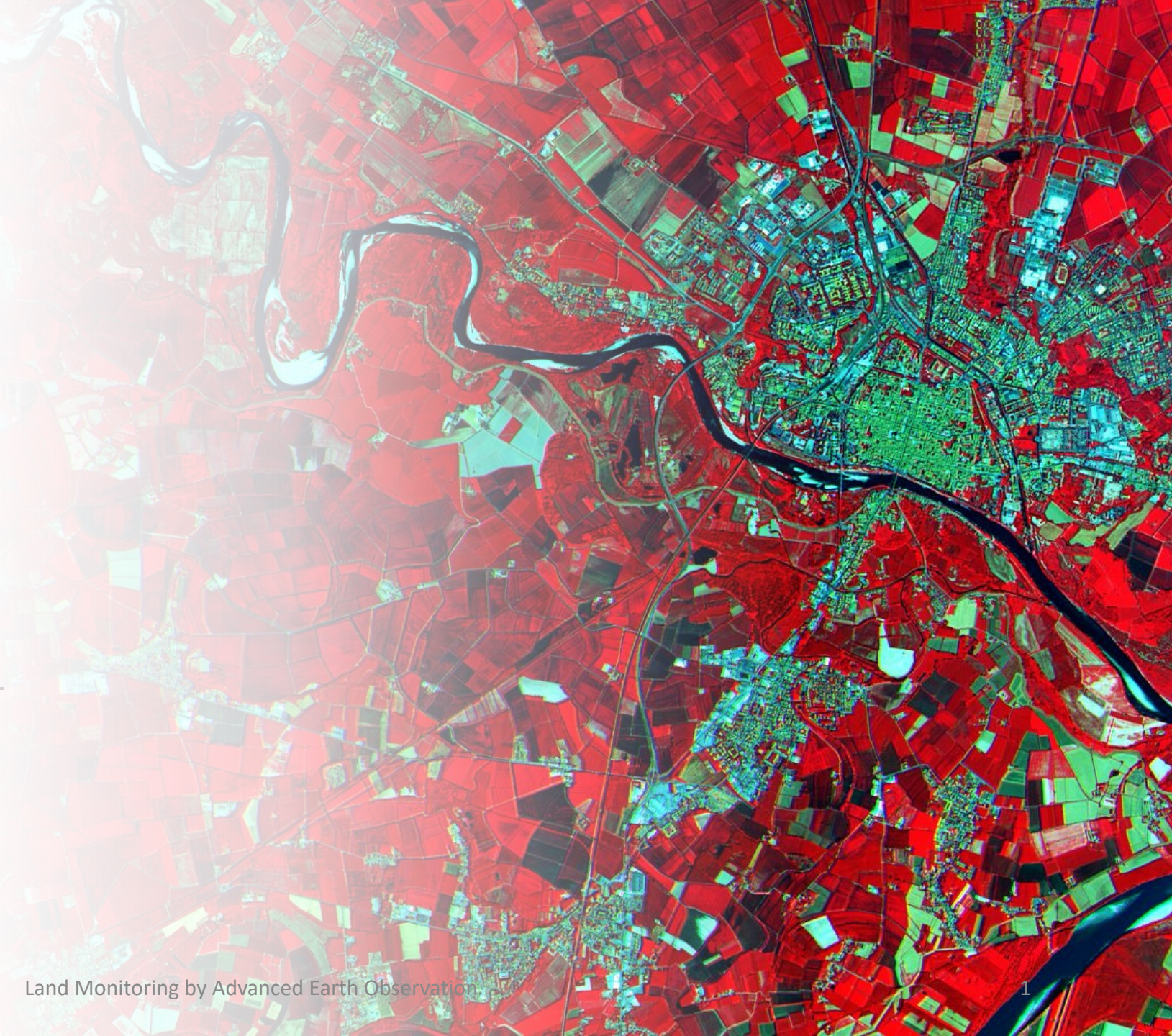
Professors

Pr. Pierre Defourny

Dr. Sophie Bontemps

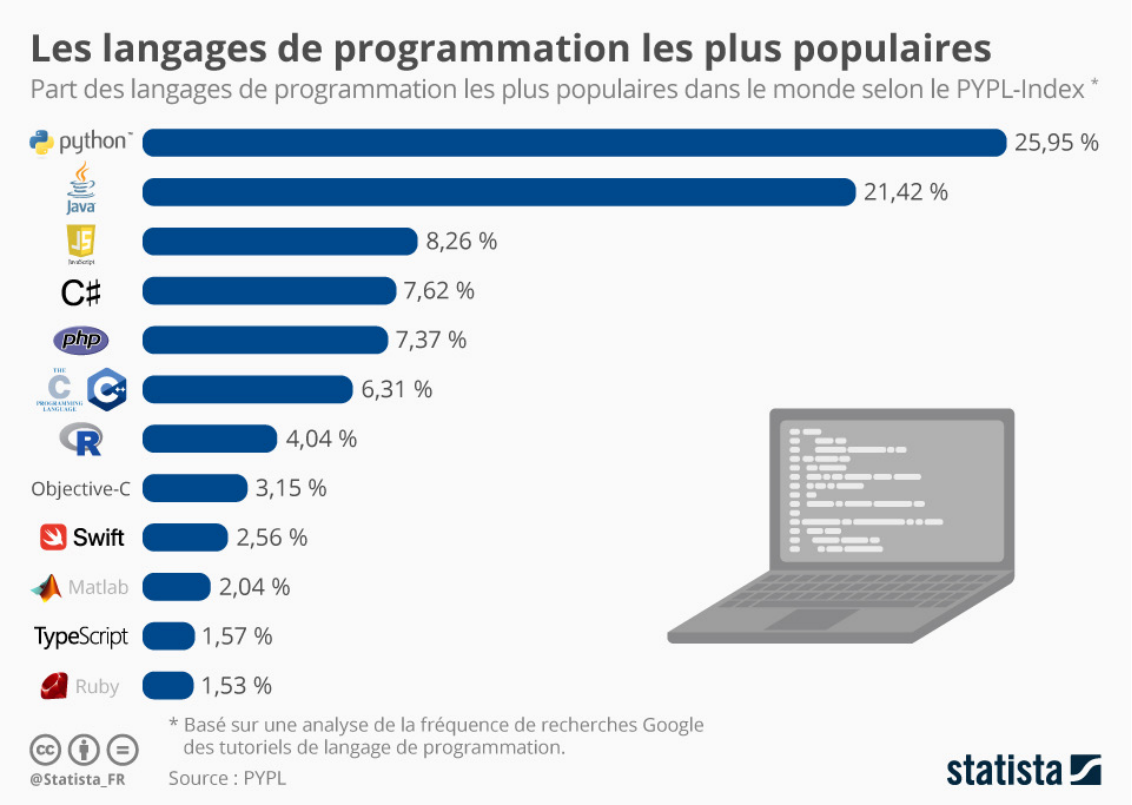
Teaching assistant

Nicolas Deffense



Objectives

- Achieve the entire processing chain from raw satellite images to meaningful results
- Learn new geoprocessing **Python** libraries
- Timeseries analyses
- Change detection
- Quality control
- Biophysical variable retrieval (NDVI, LAI, …)
- Classification with Machine Learning
- Validation & accuracy
- …



Disk management

Location	Capacity	Type	Read	Write
C:/	365GB	Local	YES	YES
Z:/LBRAT2104/	1GB / student	OASIS	YES	YES
Y:/AGRO/LBRAT2104/	256GB *(9GB free)*	OASIS	YES	NO
X:/LBRAT2104/	**512GB**	**Remote**	**YES**	**YES**

When you log out of your session, all data on disk "C:" is deleted !

We will work exclusively on disk "X:"

Create an Anaconda Environment with Python 3.7

Numpy	1.19.2
Pandas	1.2.2
Geopandas	0.8.1
Rasterio	1.1.0
Matplotlib	3.3.4

Jupyter Notebooks accessible in GitHub

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

If you want to be sure to get the last version of the Jupyter notebooks, download them from my GitHub repository! All you need is an internet connection.

https://github.com/nicolasdeffense/eo-toolbox/tree/main/notebooks

You can also open HTML version of the notebooks to have a better view (but you cannot launch code in that way!)

https://nicolasdeffense.github.io/eo-toolbox/notebooks/geopandas_basics.html
https://nicolasdeffense.github.io/eo-toolbox/notebooks/sentinel_2_prepro.html
https://nicolasdeffense.github.io/eo-toolbox/notebooks/loops_multiple_images.html
https://nicolasdeffense.github.io/eo-toolbox/notebooks/in_situ_sampling_design.html
https://nicolasdeffense.github.io/eo-toolbox/notebooks/random_forest_classification.html

Activate an environment with command line

0. Open “DS-5 v5.29.1 Command Prompt”

1. Create Anaconda environment on disk X **/!\ already done /!**

```
conda create -p X:/anaconda_env/LBRAT2104_env_ndeffense
python=3.7 numpy=1.19.2 pandas=1.1.5 geopandas=0.8.1
rasterio=1.1.0 rasterstats=0.14.0 matplotlib=3.3.2 scipy=1.5.2
```

2. Go to disk X

```
X:
```

3. Activate environment

```
conda activate /anaconda_env/LBRAT2104_env_ndeffense
```

4. Launch Jupyter Notebook

```
jupyter notebook
```

5. Close Jupyter Notebook

```
CTRL-C
CTRL-C
```

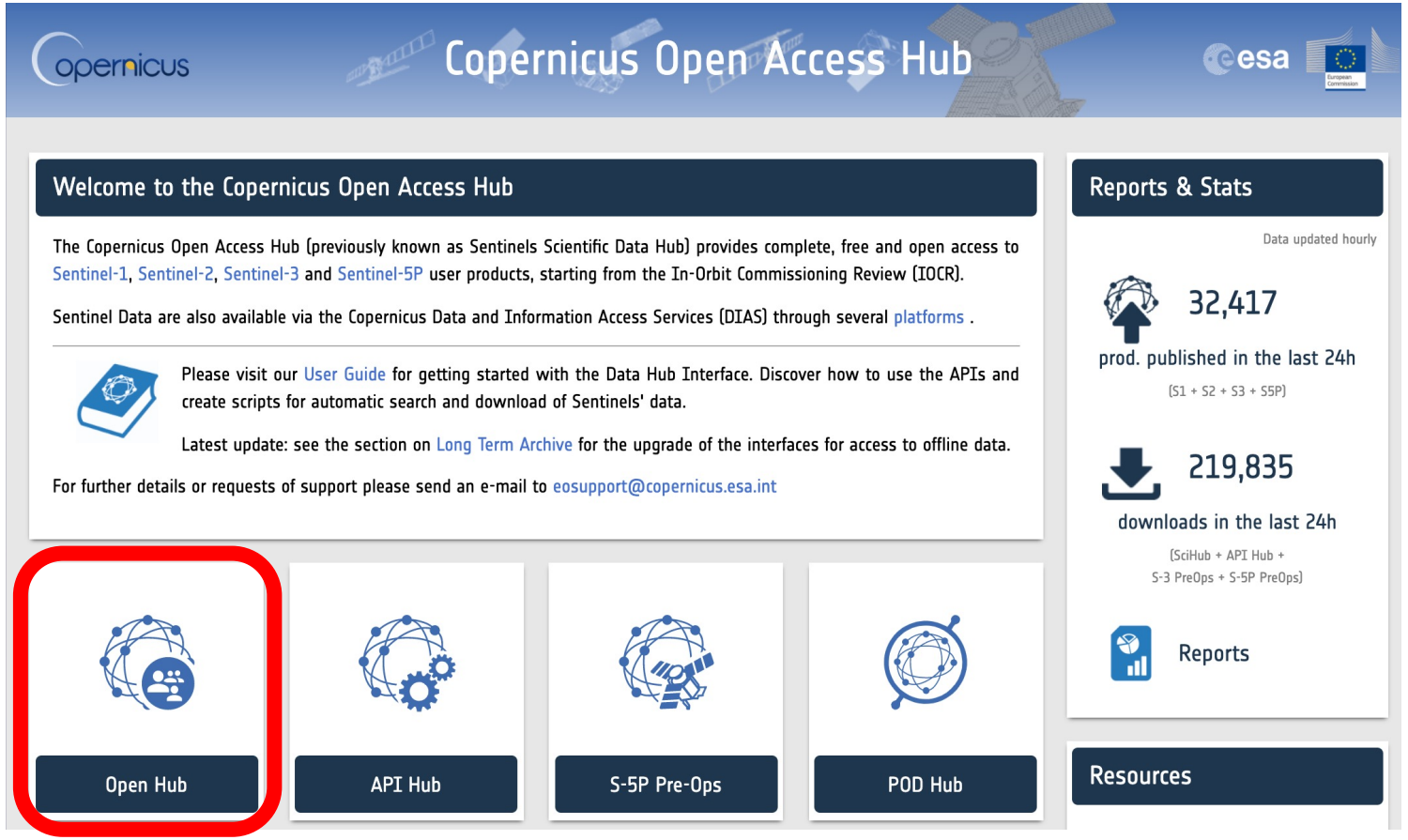
Louvain-la-Neuve, 31/07/2020
Sentinel2B – MSI – L2A

Download a satellite image

Copernicus Open Access Hub

https://scihub.copernicus.eu/

Access to all Sentinel missions data through the interactive Graphical User Interface

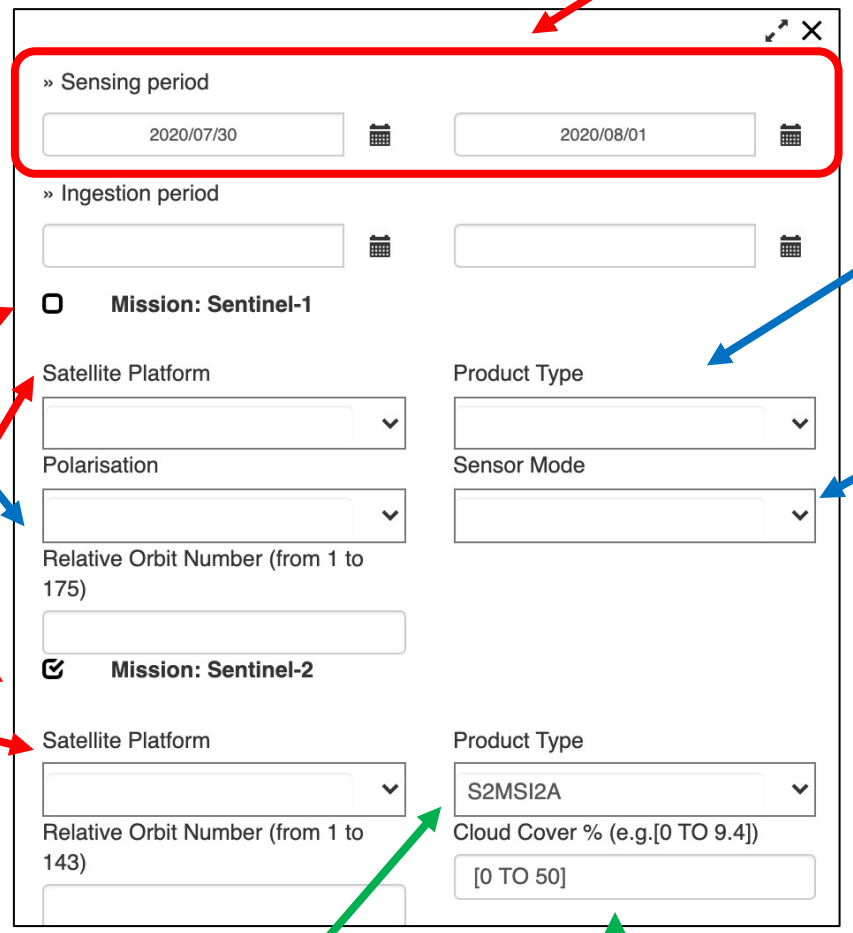


Create an account

Advanced Search

Sensing period

Polarisation
- HH
- HV
- VV
- VH



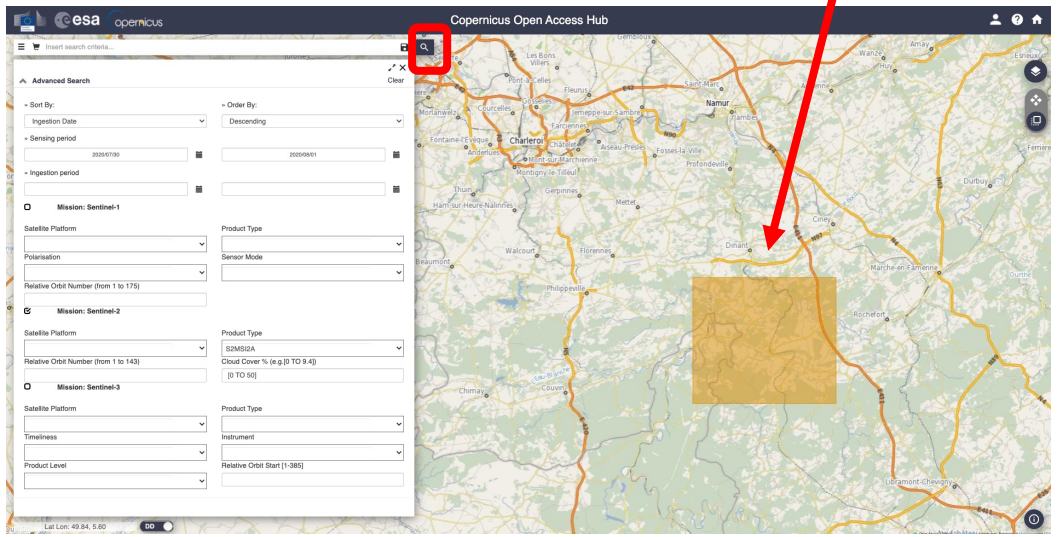
Product Type
- SLC
- GRD

Sensor Mode
- SM
- IW
- EW
- WV

Mission

Satellite Platform
- S2A / S1A
- S2B / S1B
- Both (leave empty)

Region of Interest (ROI)

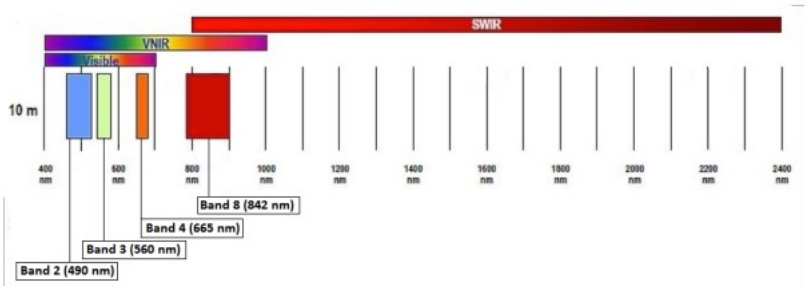


Product Type
- S2 MSI 1c
- S2 MSI 2a
- (S2 MSI 2ap)

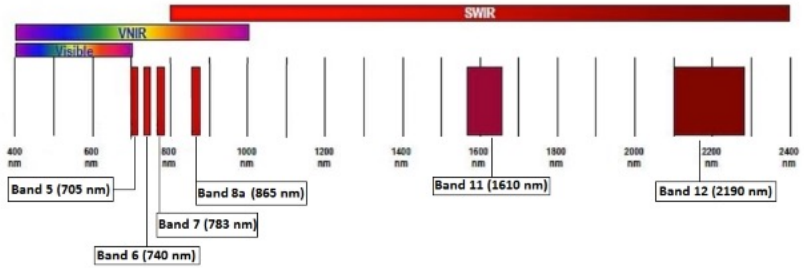
Cloud Cover

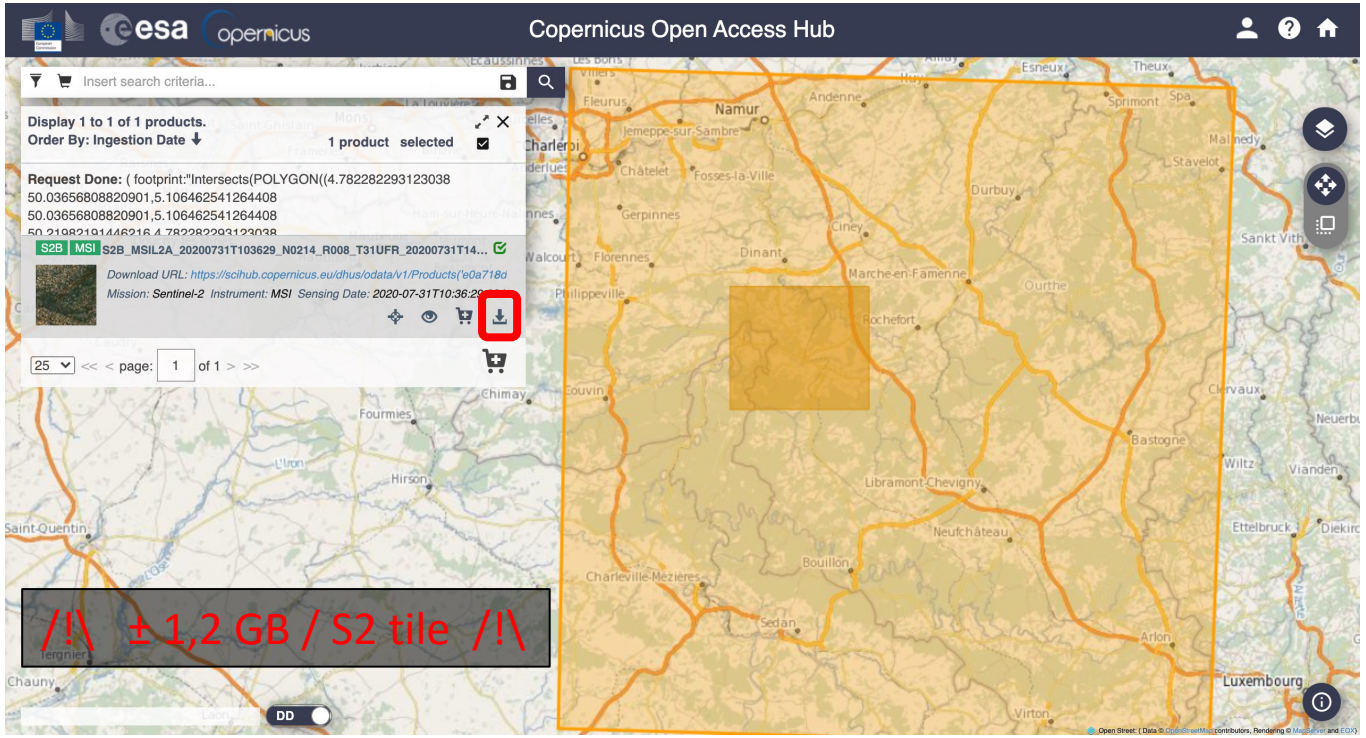
Download image

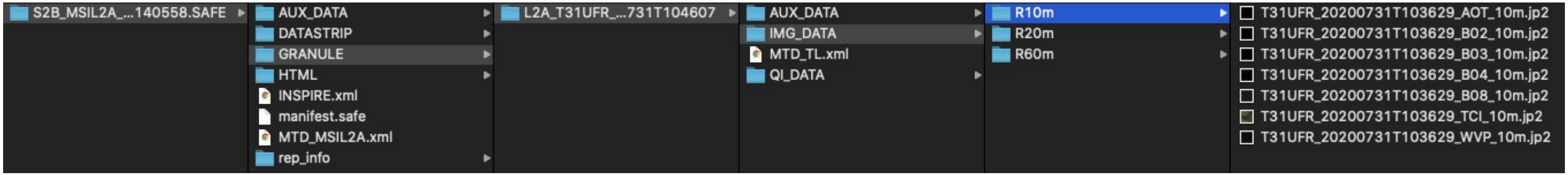
10 metre spatial resolution:



20 metre spatial resolution:



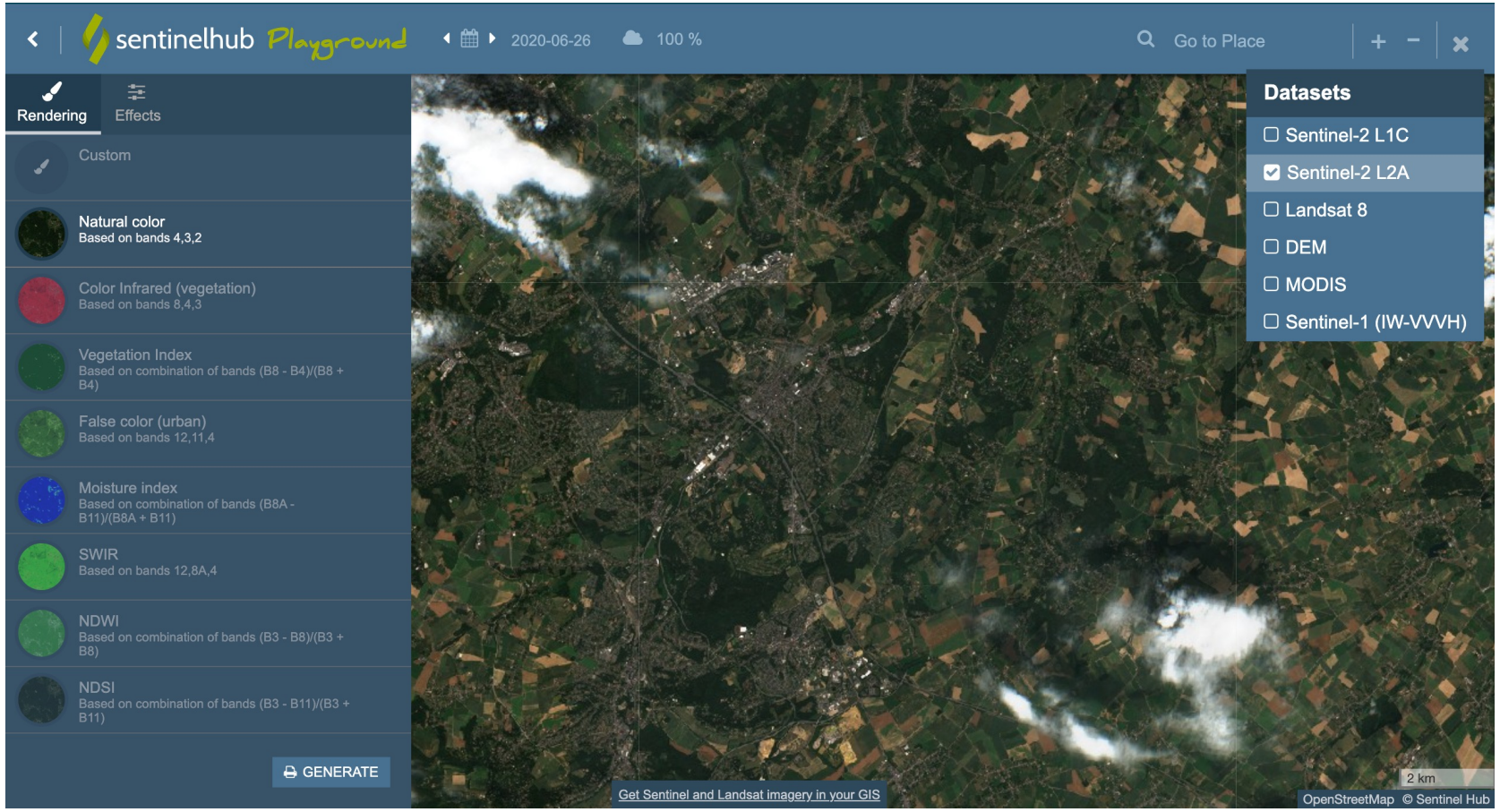




Sentinel Playground

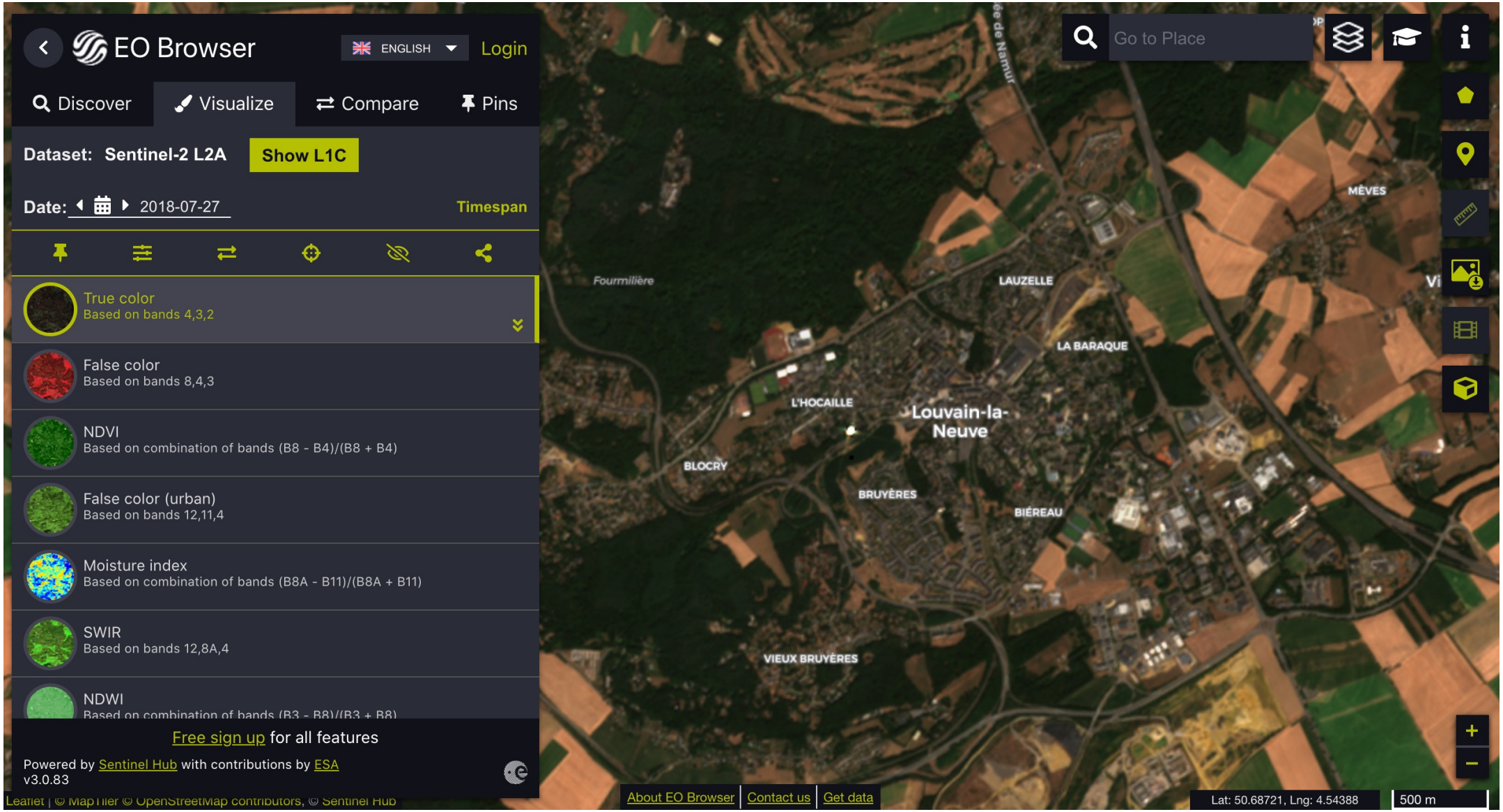
https://www.sentinel-hub.com/explore/sentinelplayground/

- Easy-to-use discovery
- Exploring of full-resolution Sentinel-1, Sentinel-2, Landsat 8, DEM and MODIS imagery
- Complete and daily updated Sentinel-2 archive

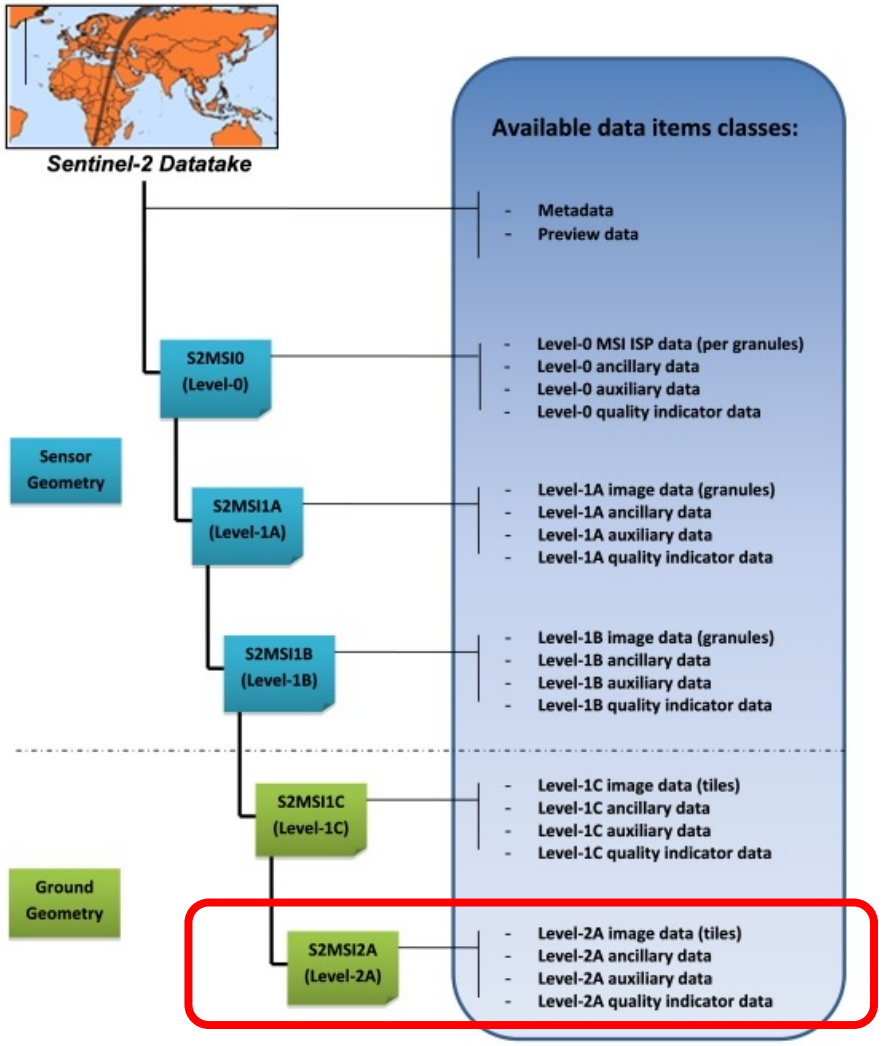


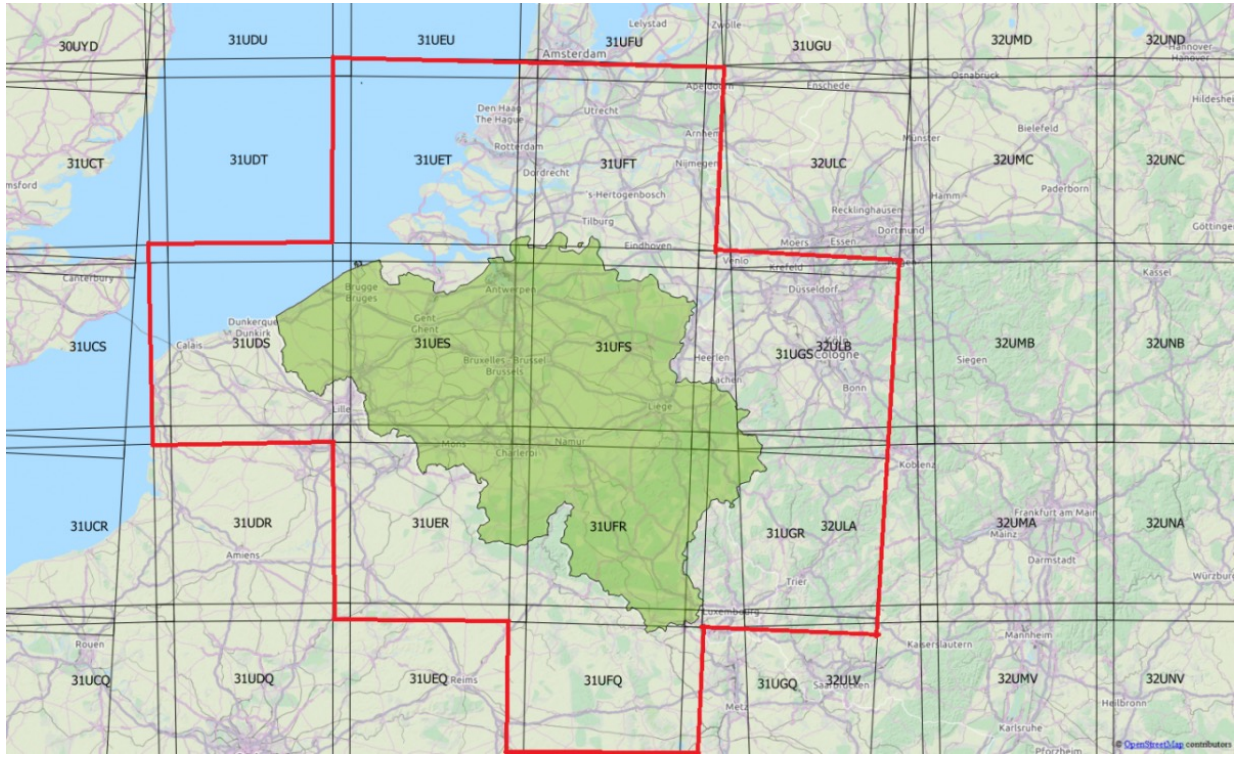
Sentinel EO Browser

https://www.sentinel-hub.com/explore/eobrowser/



Sentinel-2 – Processing Levels





The image is divided into 100 km tiles in UTM/WGS84 projection.

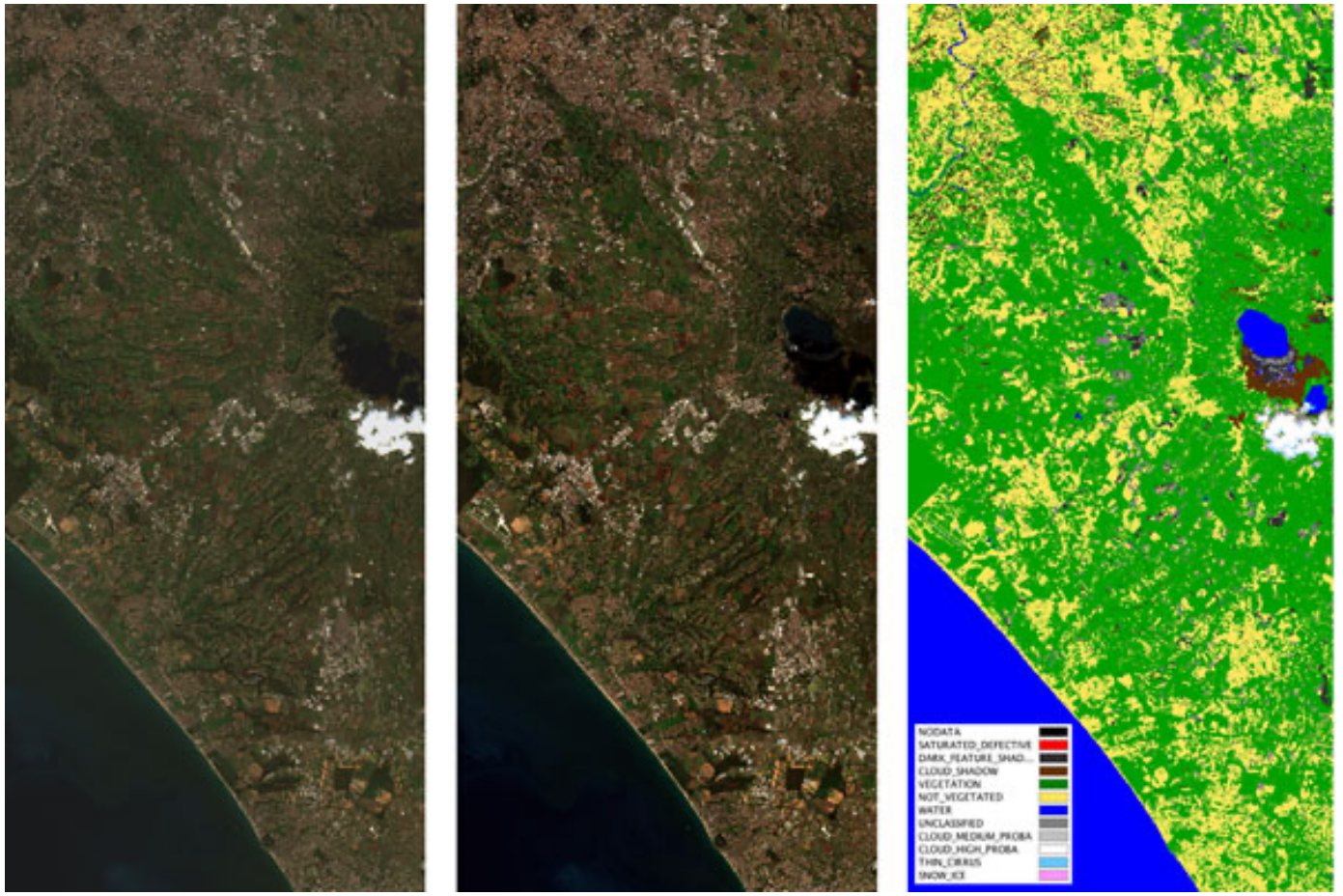
Sentinel-2 – Level-2A

Level 2A-processing is split into two parts:

- **Scene Classification (SC)** aims at providing a pixel classification map
- **Atmospheric Correction (S2AC)** aims at transforming TOA reflectance into BOA reflectance.

Label	Classification
0	NO_DATA
1	SATURATED_OR_DEFECTIVE
2	DARK_AREA_PIXELS
3	CLOUD_SHADOWS
4	VEGETATION
5	NOT_VEGETATED
6	WATER
7	UNCLASSIFIED
8	CLOUD_MEDIUM_PROBABILITY
9	CLOUD_HIGH_PROBABILITY
10	THIN_CIRRUS
11	SNOW

Scene Classification Value



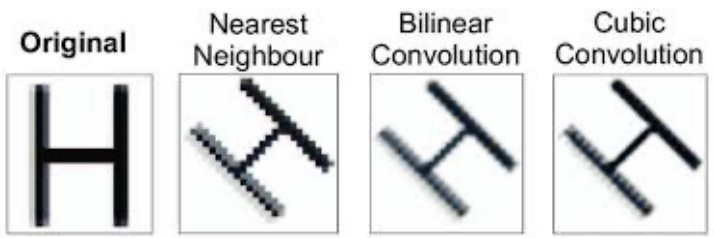
(1) (2) (3)

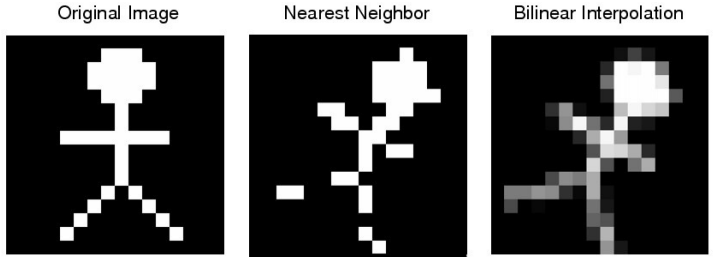
(1) Sentinel-2 Level-1C TOA reflectance input image,
(2) the atmospherically corrected Level-2A BOA reflectance image,
(3) the output scene classification of the Level-1C product

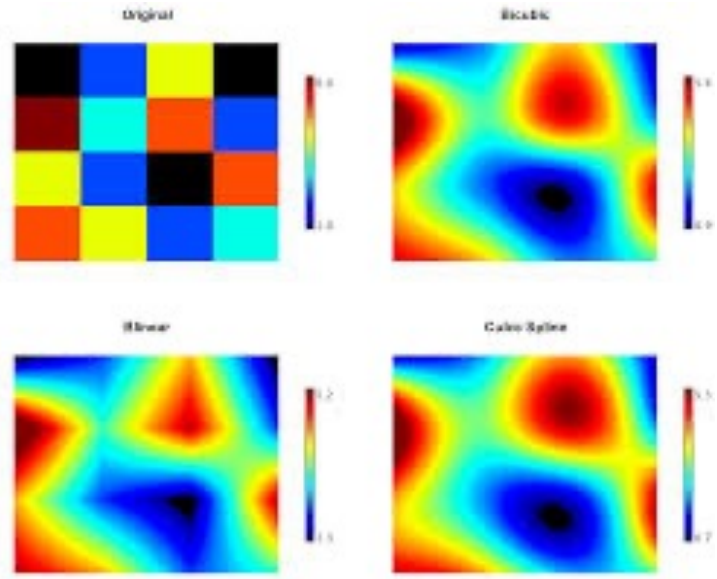
Extract ROI from satellite image

Draw a perfectly rectangular ROI

Coordinates Reference System (CRS)	
Geographic CRS	Projected CRS
degrees	meters
Lat/Lon	X/Y
World Geodetic System 1984 (WGS 84)	Universal Transverse Mercator (UTM)
EPSG:4326	EPSG:32631


Original
Nearest Neighbour
Bilinear Convolution
Cubic Convolution


Original Image
Nearest Neighbor
Bilinear Interpolation

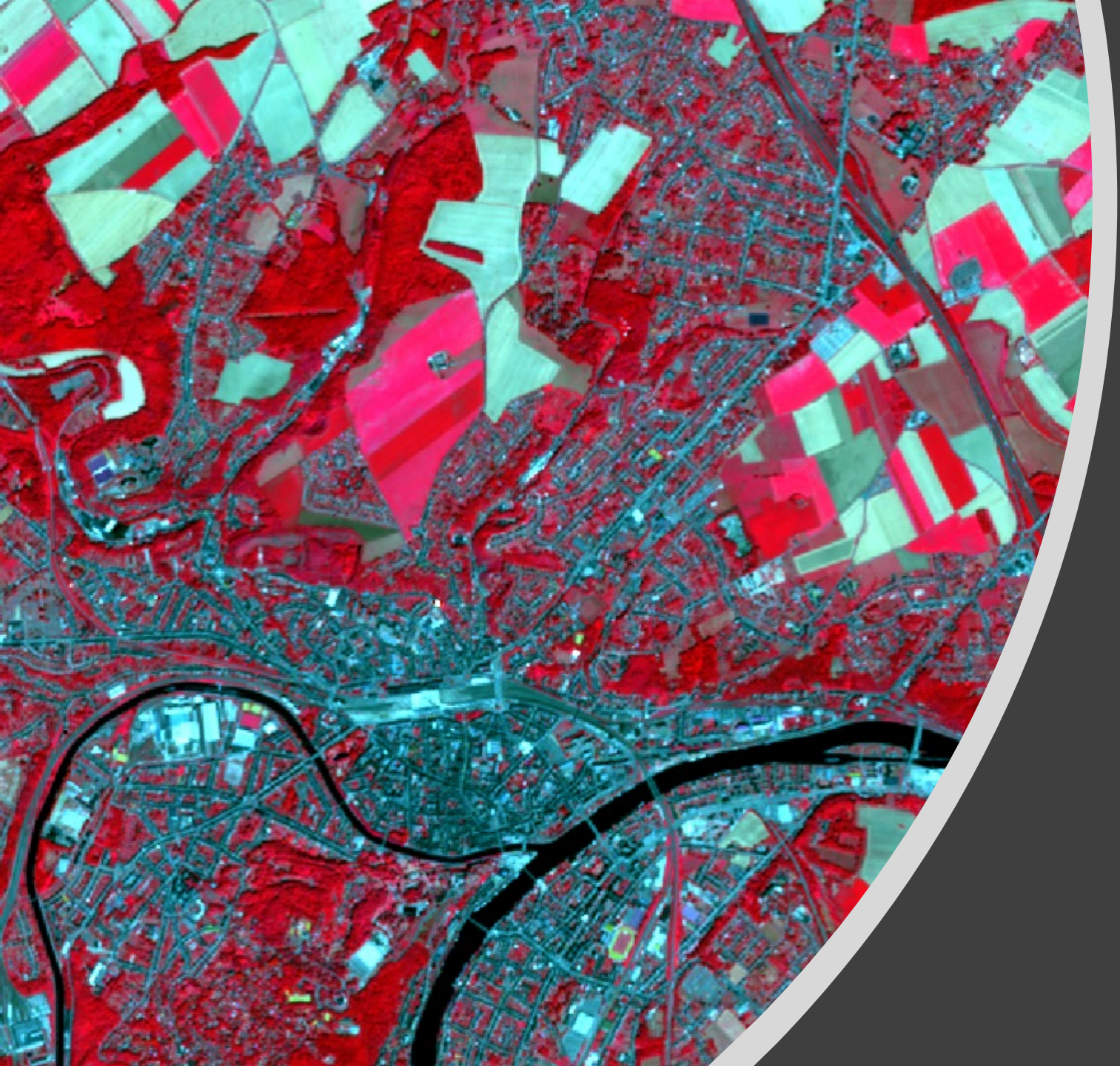


Resampling

Resampling methods

Determine how the cell values of an output raster are determined after a geometric operation is done

Nearest Neighbor	- Best used for categorical data like land-use classification or slope classification. - The value of of the output cell is determined by the nearest cell center on the input grid
Bilinear Interpolation	- Weighted average of the four nearest cell centers. The closer an input cell center is to the output cell center, the higher the influence of its value is on the output cell value. - The output value could be different than the nearest input but is always within the same range of values as the input. - Since the values can change, Bilinear is not recommended for categorical data. Instead, it should be used for continuous data like elevation and raw slope values.
Cubic Convolution	- Looks at the 16 nearest cell centers to the output and fits a smooth curve through the points to find the value. - Not only does this change the values of the input but it could also cause the output value to be outside of the range of input values (imagine a sink or a peak occurring on a surface). - This method is also not recommended for categorical data but does an excellent job of smoothing continuous data

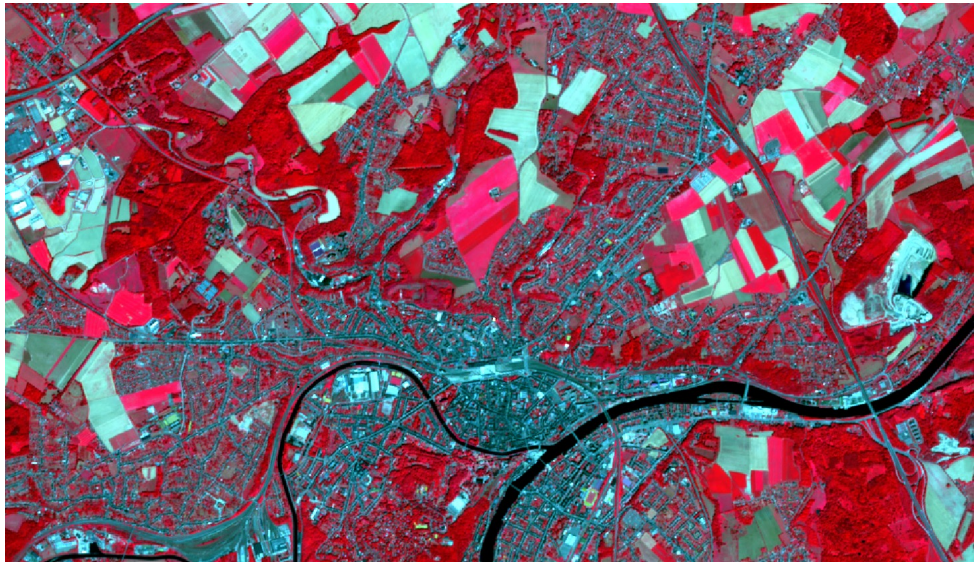


Create composite image

Simple RGB Composites

True Color

- Visible light bands : Red (B04), Green (B03) and Blue (B02) in the corresponding red, green and blue color channels
- Natural-colored result = good representation of the Earth as humans would see it naturally



False Color

- Combination of standard near infra-red (B08), red (B04) and green (B03) band.
- To assess plant density and health, as plants reflect near infrared and green light, while absorbing red.
    - Since they reflect more near infrared than green, plant-covered land appears deep red. Denser plant growth is darker red.
    - Cities and exposed ground are gray or tan, and water appears blue or black.

Biophysical Indicators

Normalized Difference Vegetation Index

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Normalized Difference Water Index

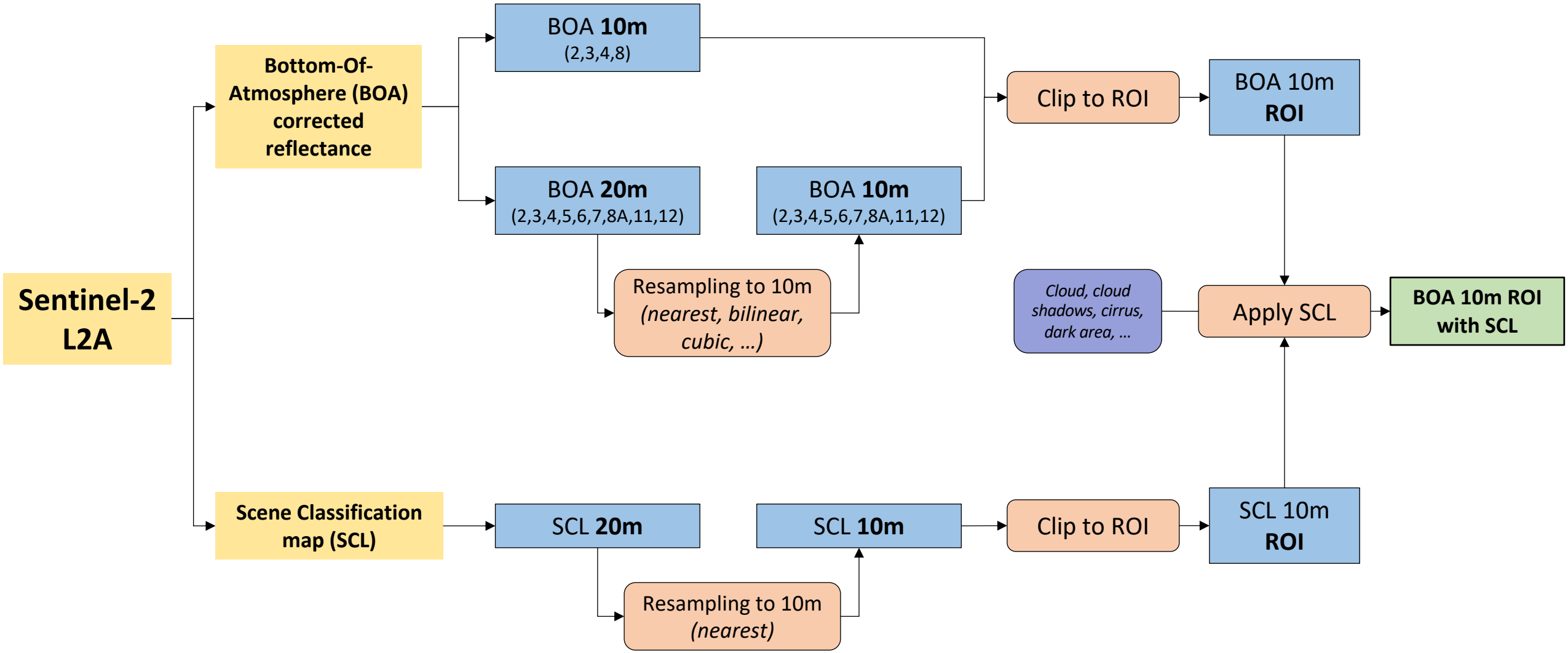
$$NDWI = \frac{GREEN - NIR}{GREEN + NIR}$$

Brightness Index

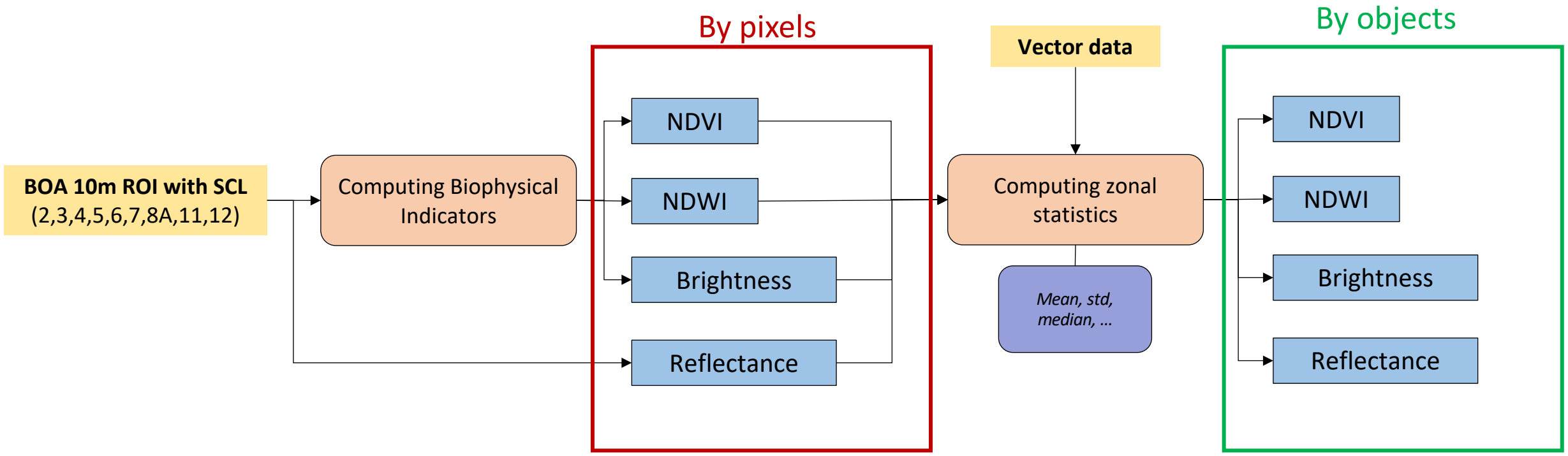
$$Brightness = \sqrt{GREEN^2 + RED^2 + NIR^2 + SWIR^2}$$

The use of NDWI and Brightness provided complementary information improving the discrimination between crop and no-crop areas.

Pre-processing Sentinel-2 images



Working by pixels or by objects



Change detection

EO-derived data at time T-n (reflectance, NDVI, NDWI, …)

EO-derived data at time T-1 (reflectance, NDVI, NDWI, …)

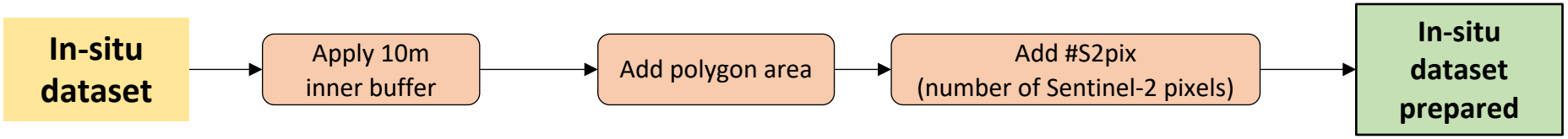
EO-derived data at time T (reflectance, NDVI, NDWI, …)

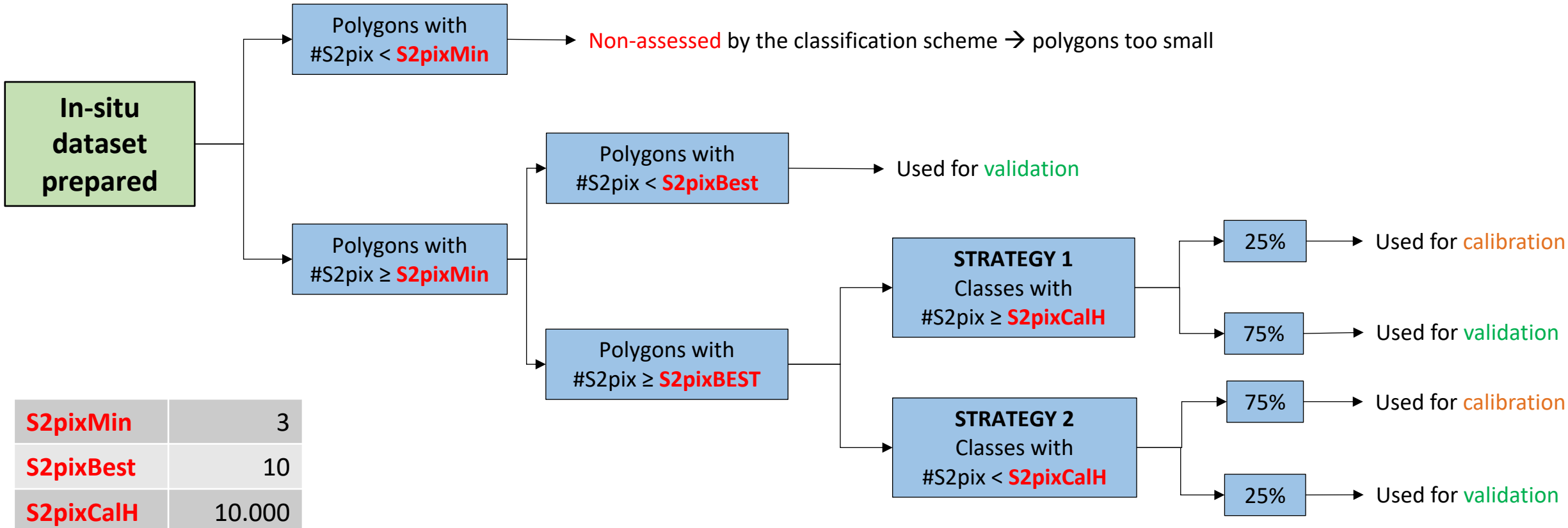
1. Open rasters with ***rasterio*** into array
2. Using ***numpy*** you can do all kinds of matrix calculations
3. Create a new matrix with the change values you want to observe
4. Save this matrix into a new GeoTIFF to display in QGIS

Exemple:

- calculate the difference between the NDVI values at time T-1 and time T
- if the difference is greater than a certain value then there is urbanization
- !! A low NDVI value corresponds to Urban but also to bare soil and water !!

In-situ Sampling Design





S2pixMin	3
S2pixBest	10
S2pixCalH	10.000

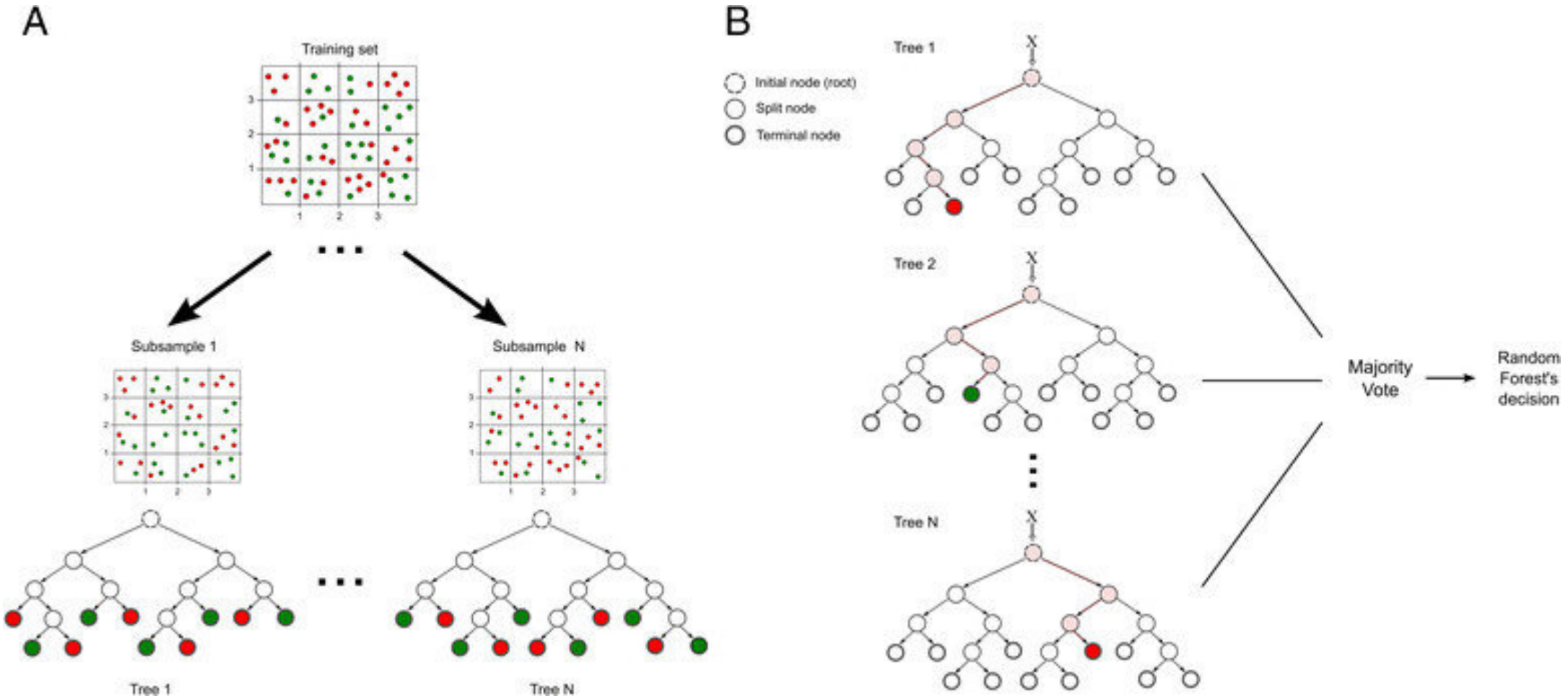
Random Forest Classification (1/2)

A. Training

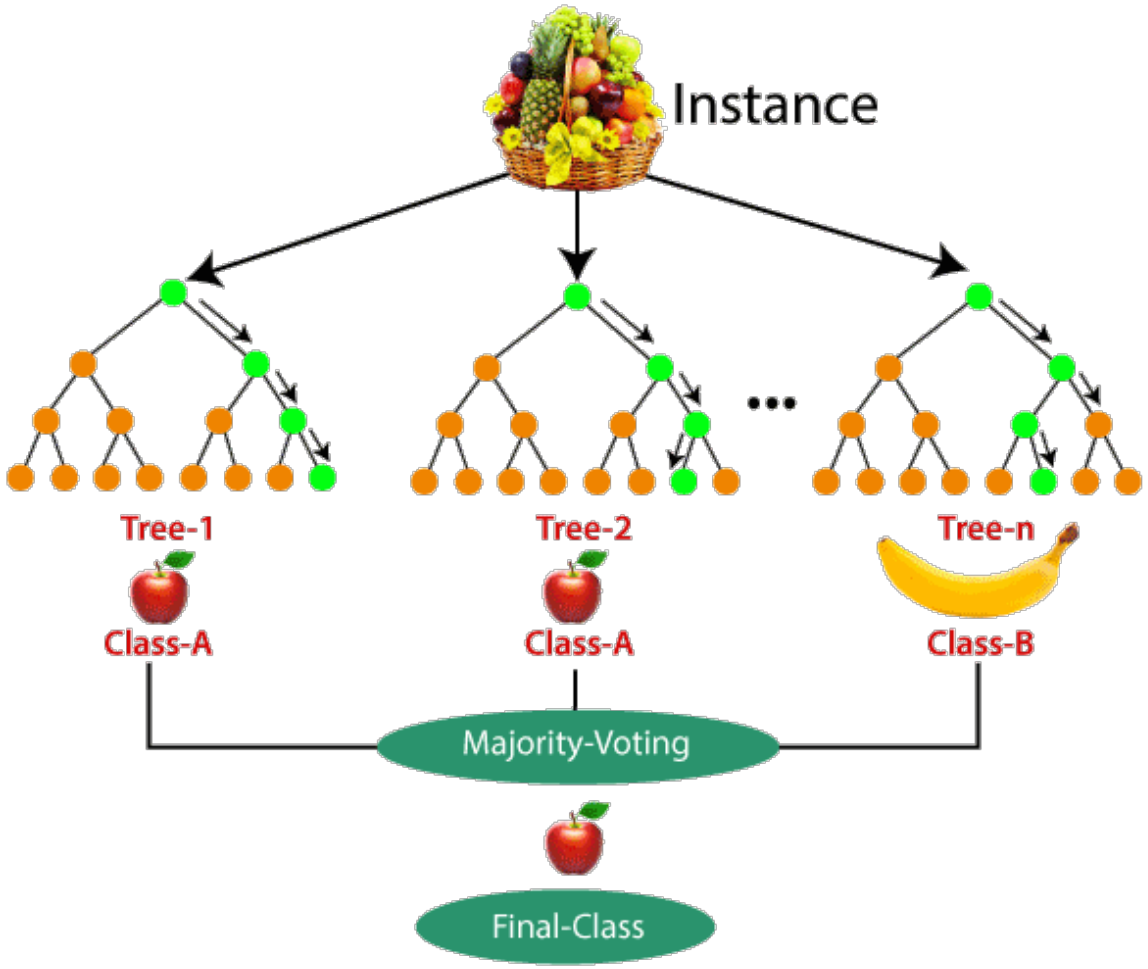
Each decision tree in the ensemble is built upon a random bootstrap sample of the original data, which contains positive (green labels) and negative (red labels) examples.

B. Prediction

Class prediction for new instances using a random forest model is based on a majority voting procedure among all individual trees.



Random Forest Classification (2/2)



The procedure carried out for each tree is as follows: for each new data point (i.e., X), the algorithm starts at the root node of a decision tree and traverse down the tree (green branches) testing the variables values in each of the visited split nodes (green nodes), according to each it selects the next branch to follow. This process is repeated until a leaf node is reached, which assigns a class to this instance.

At the end of the process, each tree casts a vote for the preferred class label. The random forest classifier collects the majority voting to provide the final prediction. The majority of the decision trees have chosen *apple* as their prediction. This makes the classifier choose *apple* as the final prediction.