Remote sensing
Lecture 16

Course of:
Signal and imaging acquisition and modelling in environment

08/05/2024

Federico De Guio - Matteo Fossati

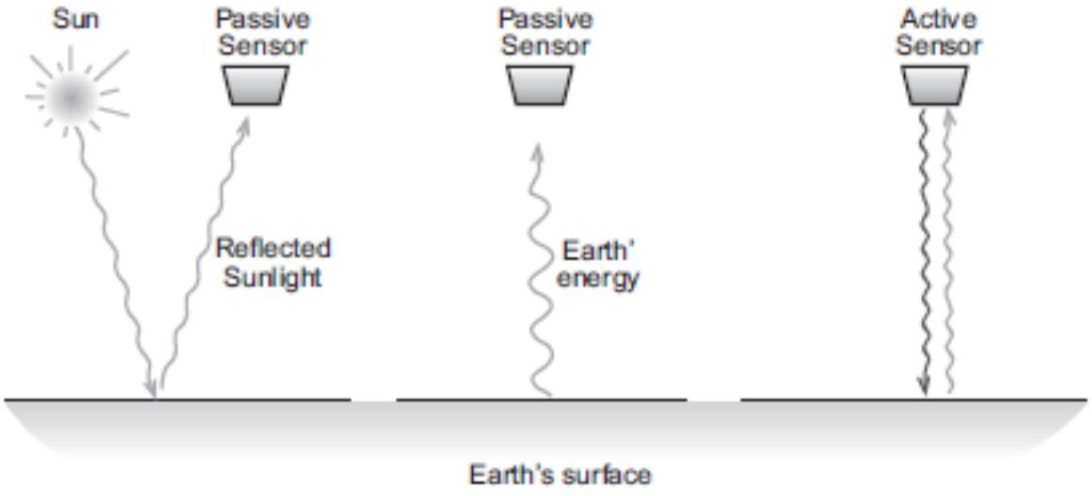
Remote sensing: definitions

- Considering the broadest definitions
    - “Remote Sensing is defined as the acquisition of information about an object **without being in physical contact with it.**“ [C. Elachi]
    - “Remote Sensing is the science and art of obtaining information about an object, area, or phenomenon through the analysis of data acquired by a device that is not in contact with the object, area, or phenomenon under investigation” [Lillesand and Kiefer]

- Fine and common definitions
    - "Remote Sensing is the science of acquiring, processing and interpreting images that record the **interaction between electromagnetic energy and matter**.“ [F.F. Sabins]
    - "The term Remote Sensing means the sensing of the Earth's surface from space by making use of the properties of electromagnetic waves emitted, reflected or diffracted by the sensed objects, for the purpose of improving natural resources management, land use and the protection of the environment.“ [UN Nations]

Different approaches

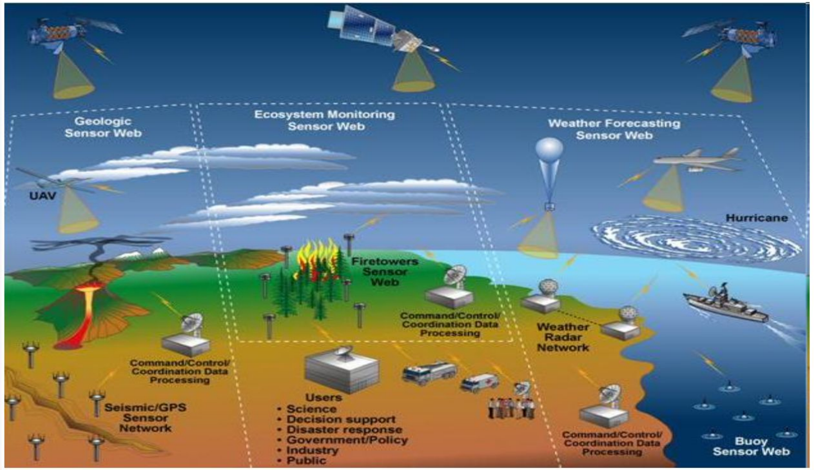
- Remote Sensing is the non-contact recording of information from the **UV, VIS, NIR, MW of the EM spectrum** by means of instruments such as **cameras, scanners, lasers, linear arrays, and/or area arrays** located on platforms such as **aircraft or spacecraft**, and the analysis of acquired information by means of **visual and digital image processing**

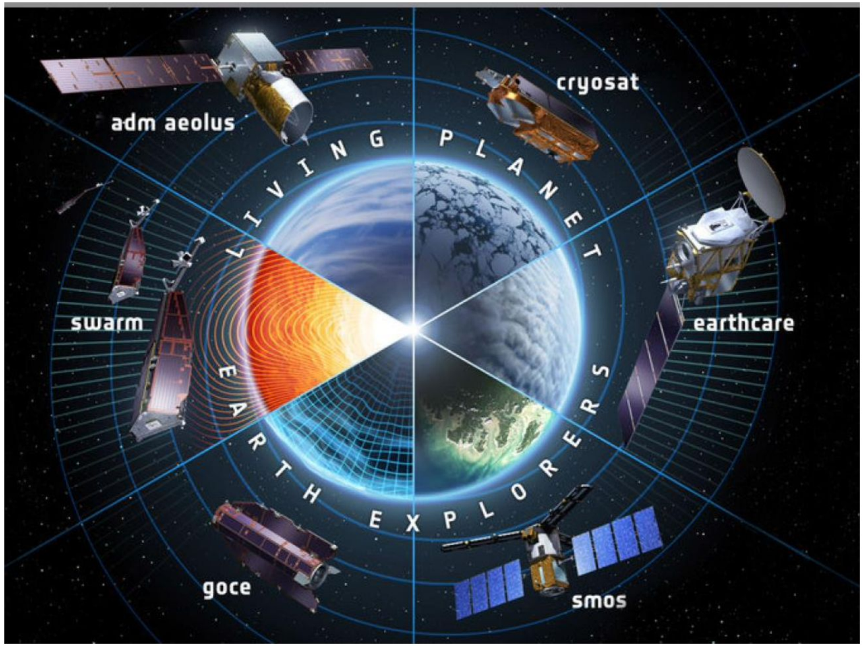


Satellite missions

- Different applications, different instruments and sensor configurations
  - Meteo, land cover, ocean monitoring, topographic

- In Europe: [Copernicus Programme]





The Sentinel satellites

- In **2014** the European Space Agency (ESA) launched **Sentinel-1 and Sentinel-2 satellites**, part of the Space component of the Copernicus program of Earth observation

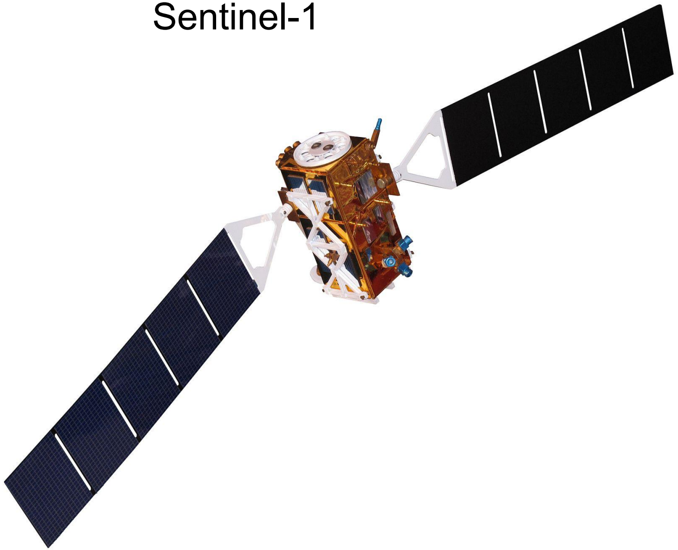
- **Sentinel-1 performs C-band synthetic aperture radar (SAR) imaging**
    - Operates both day and night, acquiring images regardless of the weather
    - Sentinel-1 interacts with the structural properties of elements in different ways and through different signal polarizations (VV, VH) in relation to their roughness and moisture content
    - absence of optical images, revisit period of six days

- **The Sentinel-2 constellation includes Sentinel-2A and Sentinel-2B**
    - provides 13-band multispectral images from visible to Short Wave Infrared (SWIR)
    - three spatial resolutions from 10 to 60 m
    - 5-day revisit cycle

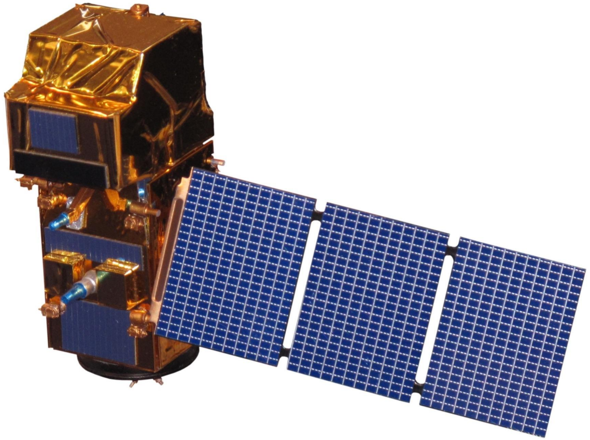
- The **integration of these two data sources** can improve land cover detection, for example for grassland, urban areas and land cover changes or hazard assessment.

The Sentinel satellites

Sentinel-1



Sentinel-2

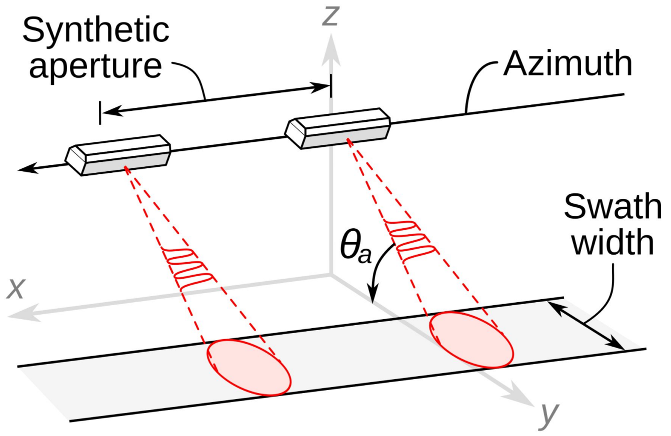


- Other Sentinel satellites exist for monitoring specific variables
  - **Sentinel-3** measures sea-surface topography, sea- and land-surface temperature, ocean color and land color
  - **Sentinel 5-P** collects data useful for assessing air quality

Synthetic Aperture Radar (SAR)

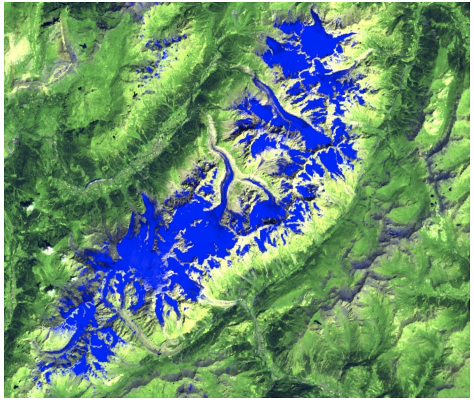
- **2D-3D reconstruction of landscapes with active sensors**
  - Successive pulses of radio waves (1m-10mm) are transmitted to "illuminate" a target scene, and the echo of each pulse is received and recorded
  - As the SAR device on board the aircraft or spacecraft moves, the antenna location relative to the target changes with time
  - Signal processing of the successive recorded radar echoes allows the combining of the recordings from these multiple antenna positions

- The SAR process forms the **synthetic antenna aperture** and allows the creation of higher-resolution images than would otherwise be possible with a given physical antenna.



Multi-spectral cameras

- Multispectral imaging captures image data within **specific wavelength ranges** across the electromagnetic spectrum
- The wavelengths may be separated by filters or detected with the use of instruments that are sensitive to particular wavelengths, including light from frequencies beyond the visible light range, i.e. **infrared** and **ultra-violet**



B12, B11, B2 are combined

Example available here

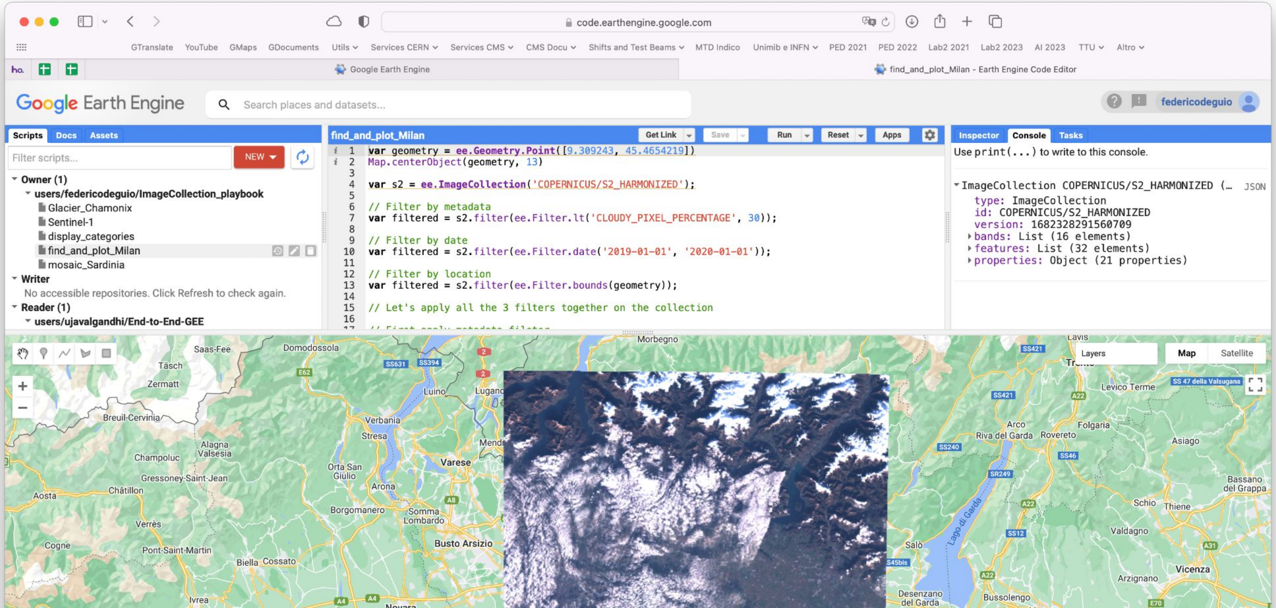
Sentinel-2 bands

Name	Units	Min	Max	Scale	Pixel Size	Wavelength	Description
B1				0.0001	60 meters	443.9nm (S2A) / 442.3nm (S2B)	Aerosols
B2				0.0001	10 meters	496.6nm (S2A) / 492.1nm (S2B)	Blue
B3				0.0001	10 meters	560nm (S2A) / 559nm (S2B)	Green
B4				0.0001	10 meters	664.5nm (S2A) / 665nm (S2B)	Red
B5				0.0001	20 meters	703.9nm (S2A) / 703.8nm (S2B)	Red Edge 1
B6				0.0001	20 meters	740.2nm (S2A) / 739.1nm (S2B)	Red Edge 2
B7				0.0001	20 meters	782.5nm (S2A) / 779.7nm (S2B)	Red Edge 3
B8				0.0001	10 meters	835.1nm (S2A) / 833nm (S2B)	NIR
B8A				0.0001	20 meters	864.8nm (S2A) / 864nm (S2B)	Red Edge 4
B9				0.0001	60 meters	945nm (S2A) / 943.2nm (S2B)	Water vapor
B11				0.0001	20 meters	1613.7nm (S2A) / 1610.4nm (S2B)	SWIR 1
B12				0.0001	20 meters	2202.4nm (S2A) / 2185.7nm (S2B)	SWIR 2

Access to data

- One way to access the data is to use Google Earth Engine which provides ready-to-use datasets from different sources including the Sentinel missions

- Needs registration → use your Bicocca account → select Unpaid registration for use in academia
- Create a default GEE project and open the GEE editor



Choose the data source

- The GEE catalogue offers the access to many datasets
- The full list is available here while the Sentinel datasets are described here

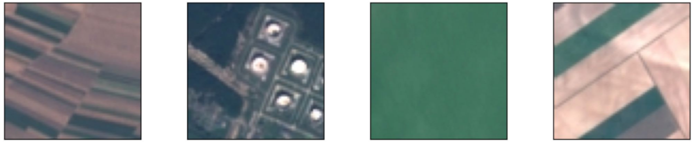
- Each dataset is pre-processed and images are harmonized
  - Variables such as "CLOUD_COVERAGE" are already available

- Let's see few examples **using multi-spectral images from Sentinel-2**:
  - Eruption of the Klyuchevskoy volcano on 8 November 2020, Russia
  - Monitoring of the Mont Blanc glacier
  - Wildfires in Sardinia
- The same can be done using the **Python API and GeeMap**
  - See an example here

- Useful source of information and examples at https://custom-scripts.sentinel-hub.com

Classification of Milan area using a pre-trained network

- **Transfer learning**
  - Use Xception architecture pre-trained on the Imagenet dataset (>1M images)
  - Use the EUROSAT dataset (27000 Sentinel-2 RGB 64x64 images with labels) to train the last layers of the DNN

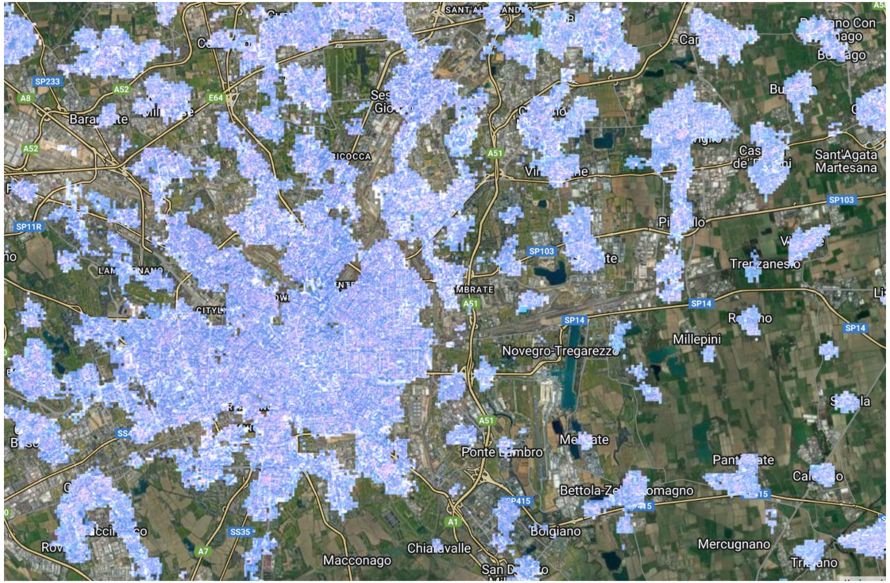


AnnualCrop (0) | Industrial (4) | SeaLake (9) | AnnualCrop (0)

- Divide a large image into 64x64 pix images and run the classification on them
  - 10 output classes

- Full workflow [here]

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6



Utilities: how to find coordinates of an area easily

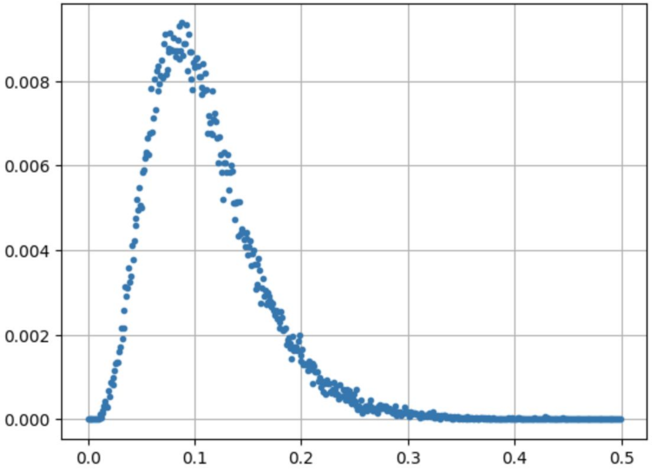
- http://geojson.io
- In GEE:
  - `coords = geoJSON['features'][0]['geometry']['coordinates']`
  - `aoi = ee.Geometry.Polygon(coords)`


```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [
          [
            [
              9.162947976416405,
              45.49093632581773
            ],
            [
              9.162947976416405,
              45.43327902302485
            ],
            [
              9.323399930962438,
              45.43327902302485
            ],
            [
              9.323399930962438,
              45.49093632581773
            ],
            [
              9.162947976416405,
              45.49093632581773
            ]
          ]
        ],
        "type": "Polygon"
      }
    }
  ]
}
```

Utilities: draw the histogram of a GEE image

```
hist = ffa_fl.select('VV').reduceRegion(
    ee.Reducer.fixedHistogram(0, 0.5, 500),aoi_sub).get('VV').getInfo()
mean = ffa_fl.select('VV').reduceRegion(
    ee.Reducer.mean(), aoi_sub).get('VV').getInfo()
variance = ffa_fl.select('VV').reduceRegion(
    ee.Reducer.variance(), aoi_sub).get('VV').getInfo()
```

```
a = np.array(hist)
x = a[:, 0]                 # array of bucket edge positions
y = a[:, 1]/np.sum(a[:, 1]) # normalized array of bucket contents
plt.grid()
plt.plot(x, y, '.')
plt.show()
```



Full example [here]

You turn

Monitoring the terrain evolution over time

- Start from a dataset from the GEE catalogue
    - Sentinel-1-2 images or other sources

- Identify a **region of interest**
    - We've seen some examples together

- Build a metric to quantify the 'amount of change'
    - Compare one or multiple snapshots over time → image ratio, compare histograms, etc

- Decide to go with a **supervised or unsupervised approach**
    - In the first case a labelled dataset is needed. One example is the EUROSAT dataset
    - In the second case a clustering algorithm can be used (already available in GEE)
    - **ML is actually not mandatory** to be able to detect changes

- Show in the classroom:
    - What is GEE
    - Intro to GEE with interactive examples: https://courses.spatialthoughts.com/end-to-end-gee.html#module-6-google-earth-engine-python-api
        - How to access satellite data and how to find your city
    - Transfer learning
    - Classification of big image downloaded with GEE using xception pre-trained on IMAGENET
        - Training of the last layers
        - Overall optimization of the training

- Project: find evolution of terrain
    - Pick a spot covered by sentinel2
    - Build the metric to spot the terrain evolution (ratio to start with)

Exercise with javascript on GEE to:

- Select coordinate
- Get an imageCollection
- Filter it using some of the features
- Retrieve a vector of pictures ordered in time
- Inspect it and play with .median or .mosaic

- Do the same with the python API

- Retrieve and classify image
    - Zoom 13
    - ee.Geometry.Point([9.309243, 45.4654219])

- Retrieve at different periods in the year or vs year and compare
    - Ratio, but also if the surface destinated to agriculture (or forest) is increasing
        - Use the classified information for this

Visualization:

- Either javascript in the dashboard
- Or geeMap (as an alternative to export to gDrive which is available from the GEE python API)

Crucial to be able to visualize stuff using python, inside a notebook
https://tutorials.geemap.org/Image/image_visualization/

Code snippet for GEE

```
var geometry = ee.Geometry.Point([9.309243, 45.4654219])
Map.centerObject(geometry, 13)

var s2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED');

// Filter by metadata
var filtered = s2.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30));

// Filter by date
var filtered = s2.filter(ee.Filter.date('2019-01-01', '2020-01-01'));

// Filter by location
var filtered = s2.filter(ee.Filter.bounds(geometry));

// Let's apply all the 3 filters together on the collection

// First apply metadata fileter
var filtered1 = s2.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30));
// Apply date filter on the results
var filtered2 = filtered1.filter(
  ee.Filter.date('2019-01-01', '2020-01-01'));
// Lastly apply the location filter
var filtered3 = filtered2.filter(ee.Filter.bounds(geometry));

// Instead of applying filters one after the other, we can 'chain' them
// Use the . notation to apply all the filters together
var filtered = s2.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30))
  .filter(ee.Filter.date('2019-01-01', '2020-01-01'))
  .filter(ee.Filter.bounds(geometry));

print(filtered);

var imageList = filtered.toList(filtered.size())
var ithImage = imageList.get(30)
```

Syllabus remote sensing course: https://elearning.unimib.it/course/info.php?id=30949#it

Slides available at: https://elearning.unimib.it/course/view.php?id=44629

- Freedom to decide what measurement to perform
    - Evolution of industrial/urban areas using ratio of labels
    - Evolution of forest/lake level (africa/south america) or glacier
        - Not always necessary to use ML, cal also use other algos and image ratio

- Use the same approach seen together
    - Transfer learning of complex pre-trained DNNs/CNNs
        - Either to classify or to perform segmentation