Food and Agriculture
Organization of the
United Nations

Introductory course to Google Earth Engine

Introductory course to

Google Earth Engine

by
Gianluca Franceschini and Mehwish Ali

Food and Agriculture Organization of the United Nations
Rome, 2022

Required citation:
Franceschini, G. and Ali, M. 2022. *Introductory course to Google Earth Engine*. Rome, FAO. https://doi.org/10.4060/cb9049en

ISBN 978-92-5-135907-5

Cover photograph: ©FAO/Shahid Ahmed

Contents

Figures

Tables

Acknowledgements

We would like to acknowledge the support from FAO Pakistan communications team and geospatial team, as well as the support from the geospatial unit in the Land and Water Division (FAO NSL). The layout, typesetting, and graphic designing was done by Mr. Shahid Ahmad (FAO Pakistan) while Ms. Areesha Asghar (FAO Pakistan) improved the editing of the document.

Abbreviations and acronyms

API	Application Programming Interface
DEM	Digital Elevation Model
ESA	European Space Agency
EW	Extra-Wide swath
GEE	Google Earth Engine
GRD	Ground Range Detected
IDE	Integrated Development Environment
IW	Interferometric Wide swath
OCN	Level-2 Ocean
LTA	Long-term average
NASA	National Aeronautics and Space Administration
NIR	Near-Infrared Red
NDVI	Normalized Difference Vegetation Index
S1	Sentinel-1
S2	Sentinel-2
STA	Short-term average
SLC	Single Look Complex
SM	Stripmap
SAR	Synthetic Aperture Radar
VH	Vertical-Horizontal
VV	Vertical-Vertical
WV	Wave

INTRODUCTORY COURSE ON GOOGLE EARTH ENGINE FOR LAND COVER CLASSIFICATION

The Food and Agriculture Organization (FAO) of United Nations in Pakistan in collaboration with the Geospatial Unit of the Land and Water Division is inviting you to an introductory course on Google Earth Engine for land cover mapping. The objective is to provide the basic skills to operate the platform, select, pre-process and analyse satellite imagery relevant to agriculture and food security, in particular for the identification of specific crops in the land and more broadly for land cover mapping, by using an automatic classification approach. The book is thought for specialists on land use planning, agronomists and food security experts. It requires an understanding of the main satellite missions, basic concepts of remote sensing and scripting languages.

1. Remote sensing principles and applications for agriculture

1.1 Principles

Remote sensing allows to continuously detect information on land that is used in a large number of agricultural applications. The particular response of a plant to the incoming solar radiation allows to identify an agricultural land respect to other land uses and with some limitations a specific crop type. This response defines the spectral signature of the plant, and depends on the amount of surface solar radiation that is absorbed or reflected.

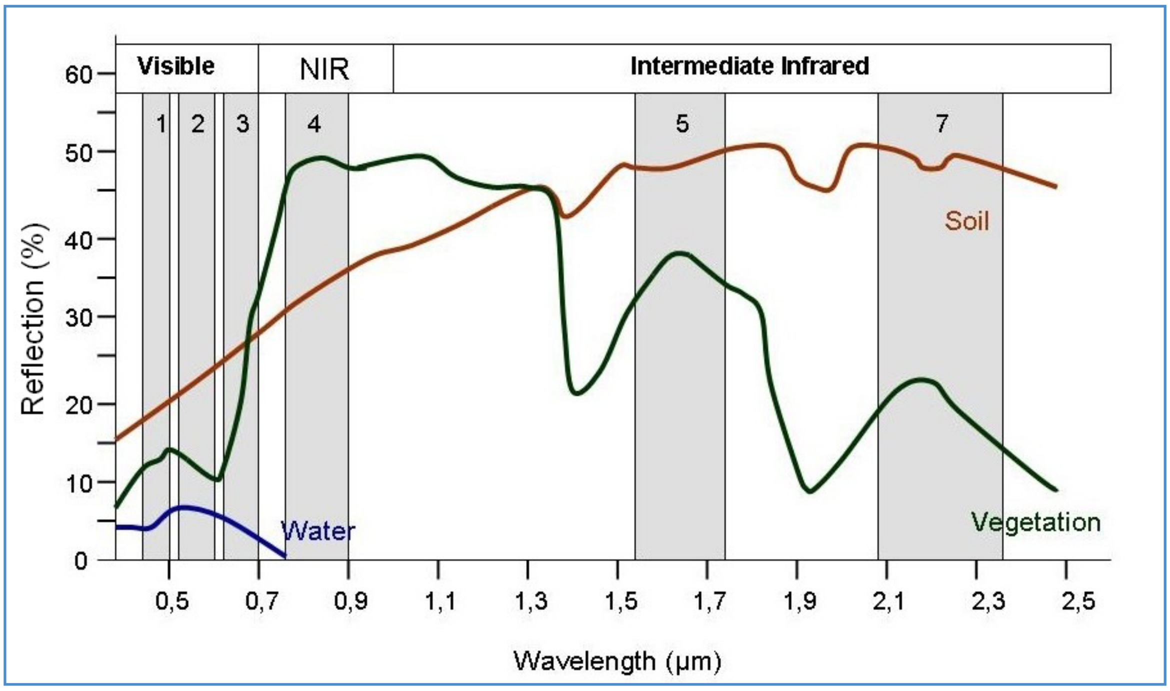


Figure1. Spectral signature of different land cover features

Source: Science Education through Earth Observation for High Schools (SEOS). 2022. Introduction to Remote Sensing. In: SEOS. European Commission. https://seos-project.eu/remotesensing/remotesensing-c00-p01.html

At the wavelength ranges of the light (the so-called visible bands), the reflectance from vegetation is relatively low as the majority of light is absorbed by the leaf pigments. Chlorophyll strongly absorbs energy in the blue and red wavelengths and reflects more green wavelengths. This is why healthy vegetation appears green. The Near-Infrared Red (NIR) part of the solar radiation is not absorbed by the plant (it does not have enough energy content to start photosynthesis but it may harm leaf internal structure). The strong difference between red and NIR bands in healthy vegetation is captured

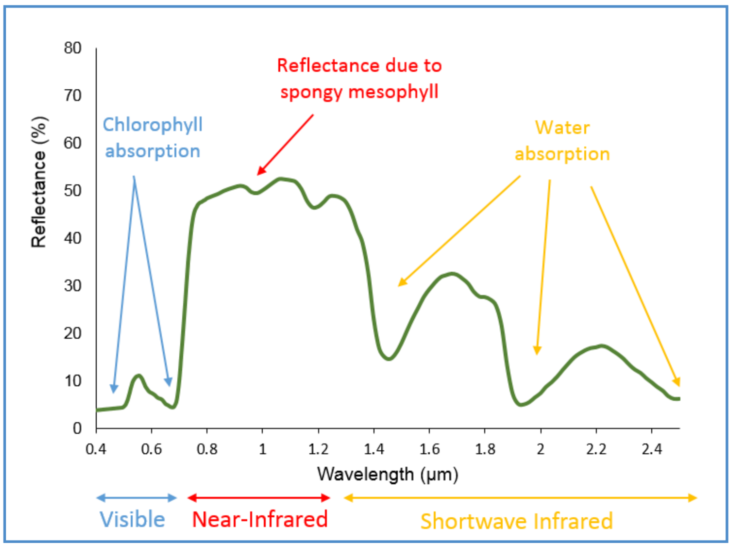


Figure 2. Spectral signature of vegetation

Source: Humboldt State University. 2022. Spectral reflectance curve of vegetation. In: Geospatial science program. California, Humboldt State University. http://gis.humboldt.edu/gsp/programs.html by the Normalized Difference Vegetation Index (NDVI). The other two peaks of absorption are determined by the water content in the vegetation.

The described spectral signature is in general common to healthy vegetation, however variation on the magnitude of the reflectance can be an important discriminator for individual crops.

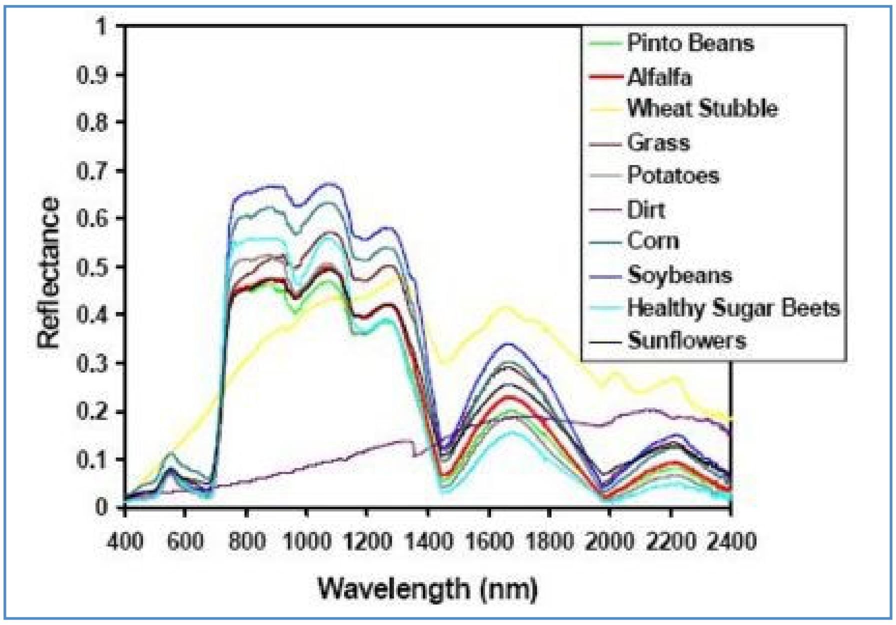


Figure 3. Spectral signature of crops and soil

Source: Kyllo, K. P. 2003. Department of Space Studies, University of North Dakota.In: Geospatial technology. https://mapasyst.extension.org/agricultural-remote-sensing-basics/

The other important aspect that helps us to distinguish different type of crops is related to the different phenology of each plant. The number of days from emergence to full maturity and harvest vary for each crop and allows to match a temporal sequence of satellite images to the specific crop calendars.

Other characteristics that are important to discriminate crops are the texture, spatial arrangements, contrast and in some case, presence of specific management practices (only with very high resolution images).

For the sake of completeness, we also have to mention all those elements that may create misclassification when identifying agricultural land and in particular specific crops:

- the complexity of topography that alters the geometry and the reflectance of the image;
- management practices, namely inter-cropping or continuous rotation as in the slash and burn agriculture;
- small and interspersed agricultural fields;
- heterogeneity of crop varieties and management practices on the same crop;
- atmospheric conditions such as clouds, aerosols, dust, smog (mainly for optical images).

Depending on a specific area, many of these factors can coexist and interact together making the classification process harder and prone to errors.

Another important aspect that may classification fail is due to an incorrect or not representative number and types of samples required to build a classification model (errors in location, inability to capture the whole ranges of spectral signatures, mistakes in classification).

Nevertheless, advances in remote sensing and computer science, increased availability of satellite images make remote sensing technology apt for many agricultural applications.

1.2 Remote sensing applications for agriculture

Detection of crop-types and acreage identification is an important application in agriculture. This has many direct implications related to food prices, import-export and more generally with national food security policies.

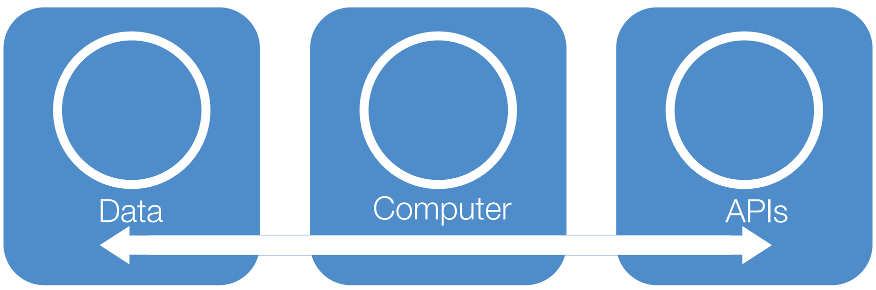
Similarly to the quantification of full mature crops in the land, the early forecast of crop yields is an important requirement for food security and investments. Many techniques exists: one simple approach is through the use of empirical relationships with vegetation indices (e.g. NDVI) at early stages of crop development.

Quantification of crop losses and damages relates to emergencies plan, disaster risk management and food security. This is typically done by comparing “normal” averages (often long-term series) to annual indices in order to detect anomalies in affected areas. It may involves biotic damages (e.g. pests and disease) as well as abiotic disasters (floods, fires, and droughts).

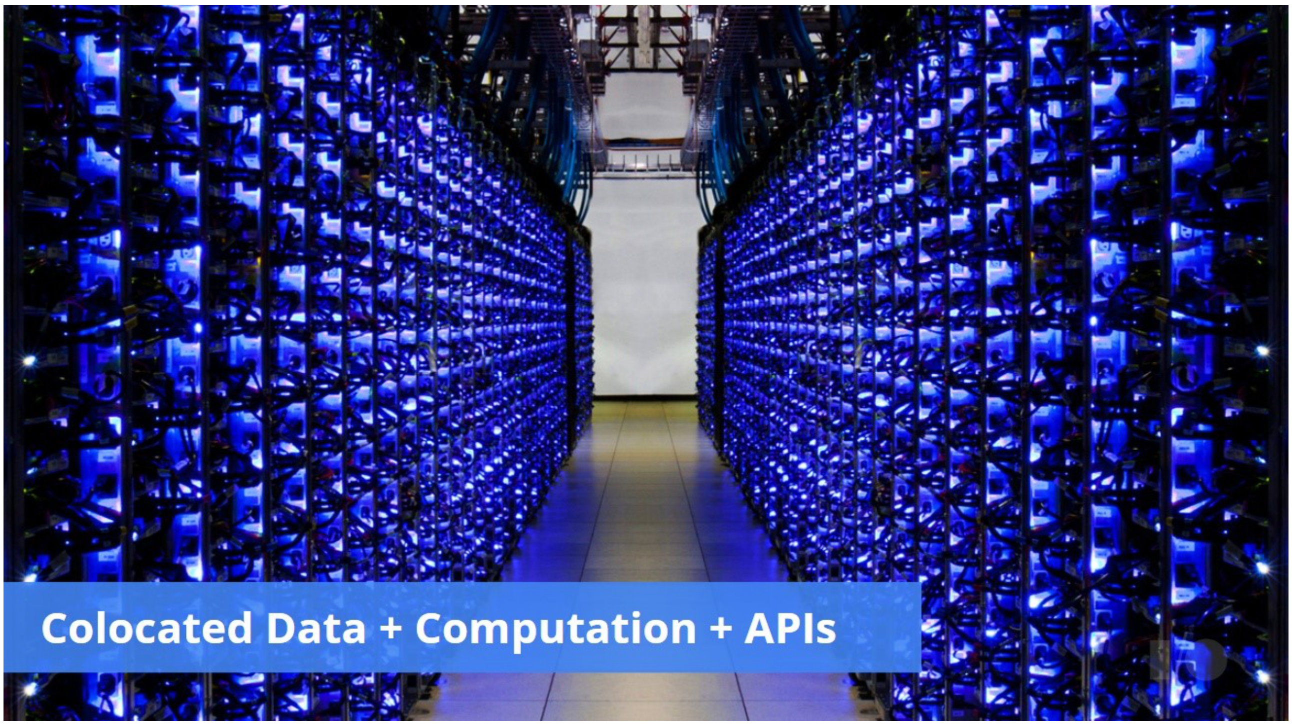
Remote sensing information contributes to irrigation research by either identifying the irrigated areas or quantifying the amount of water required/supplied. Evapotranspiration can be estimated with energy balance models in order to assess the specific crop water requirements and irrigation needs.

2. What is Google Earth Engine

Google Earth Engine (GEE) is a cloud-based platform for planetary-scale geospatial analysis that brings Google's massive computational capabilities to bear on a variety of high-impact societal issues including deforestation, drought, disaster, disease, food security, water management, climate monitoring and environmental protection.



2.1 Cloud processing with built-in functions

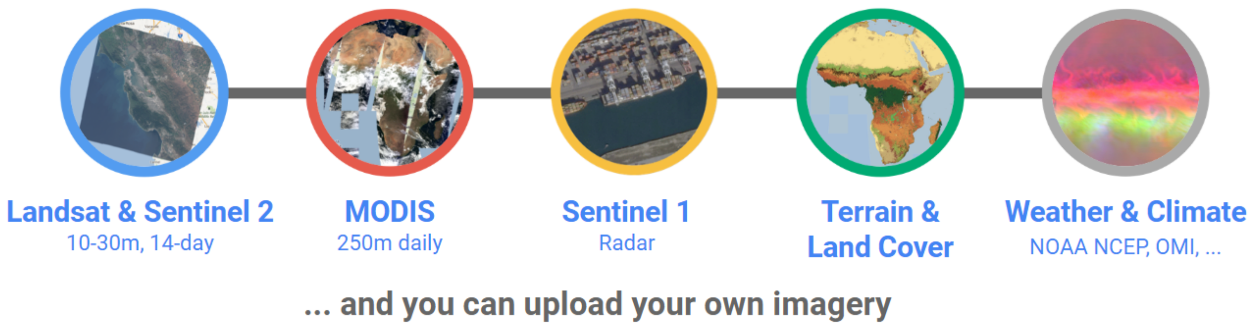


GEE is designed for cloud-based, parallelized geospatial data analysis, and it takes care of all the infrastructure and parallelization decisions on the back end for you. Those operations are called “server-side”.

Using GEE, you can call a wide set of functions that have been developed specifically for computing in Earth Engine and apply them over many images simultaneously using Google computational infrastructure. No more downloading and analyzing individuals tiles at a time or stressing about your local storage.

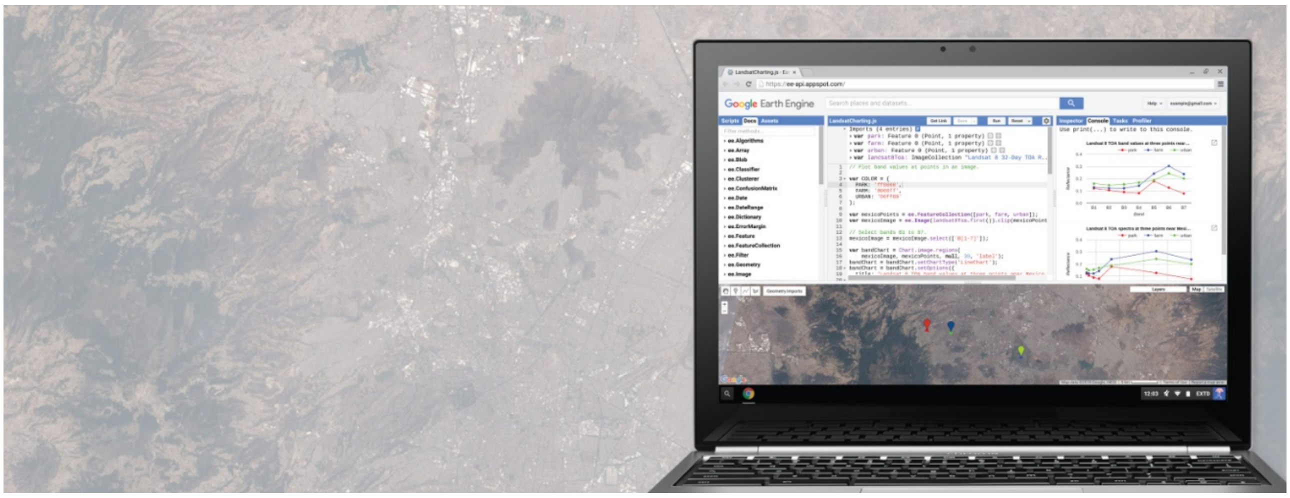
2.2 ONLINE PUBLIC DATA ARCHIVE

It contains constantly updated time-series of satellite imagery (MODIS, LandSat, Sentinel) and climatic data



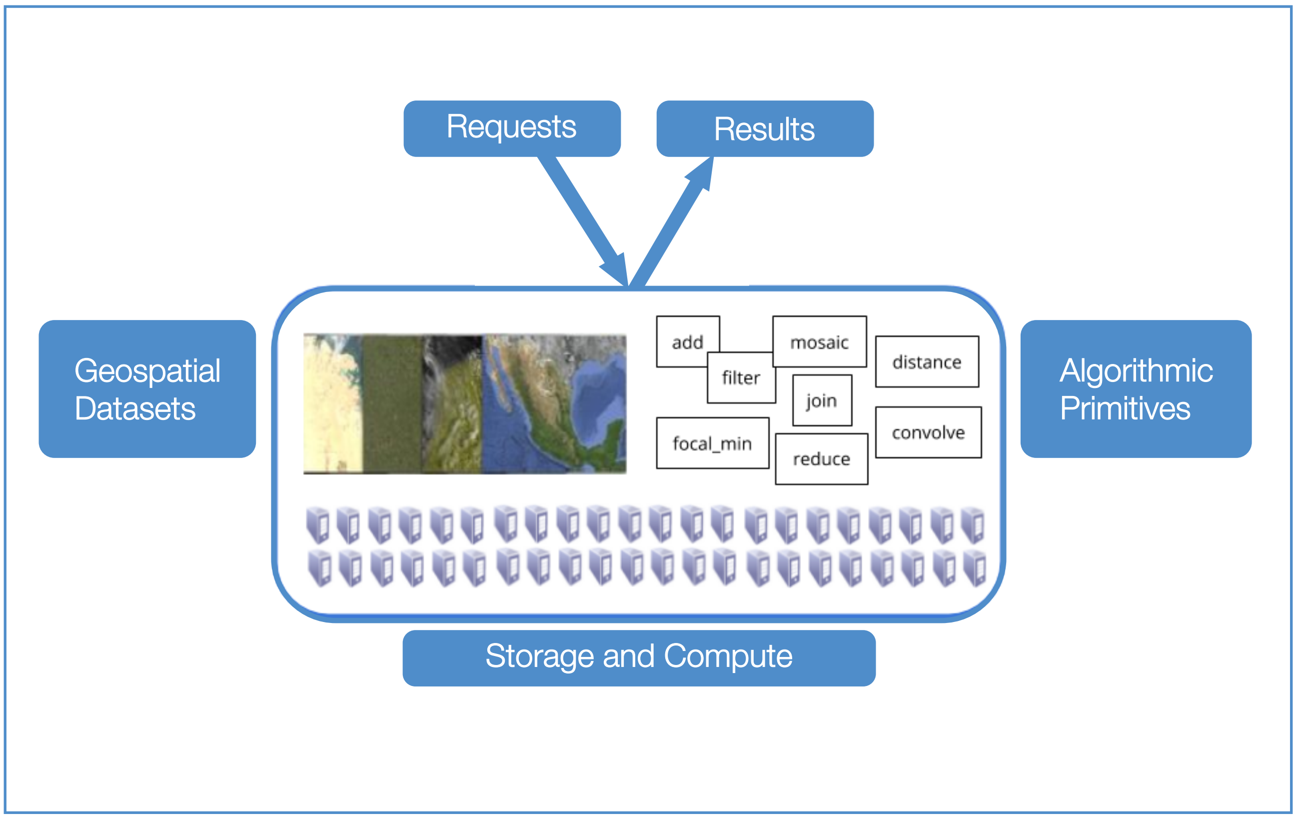
2.3 Interacting with data

The Javascript Application Programming Interface (API) is a set of functions that allows you to interact with the data through a dedicated code editor.



2.4 How it works

Using the code editor, you write commands that are sent as an object to Google for processing in parallel in their cloud (server-side). Users can visualize results from Google in their browser (client-side), including objects like maps, charts or statistical results.



3. The code editor

The code editor at code.earthengine.google.com is a web-based Integrated Development Environment (IDE) for the Earth Engine JavaScript API. Code editor features are designed to make developing complex geospatial workflows fast and easy. The code editor has the following elements:

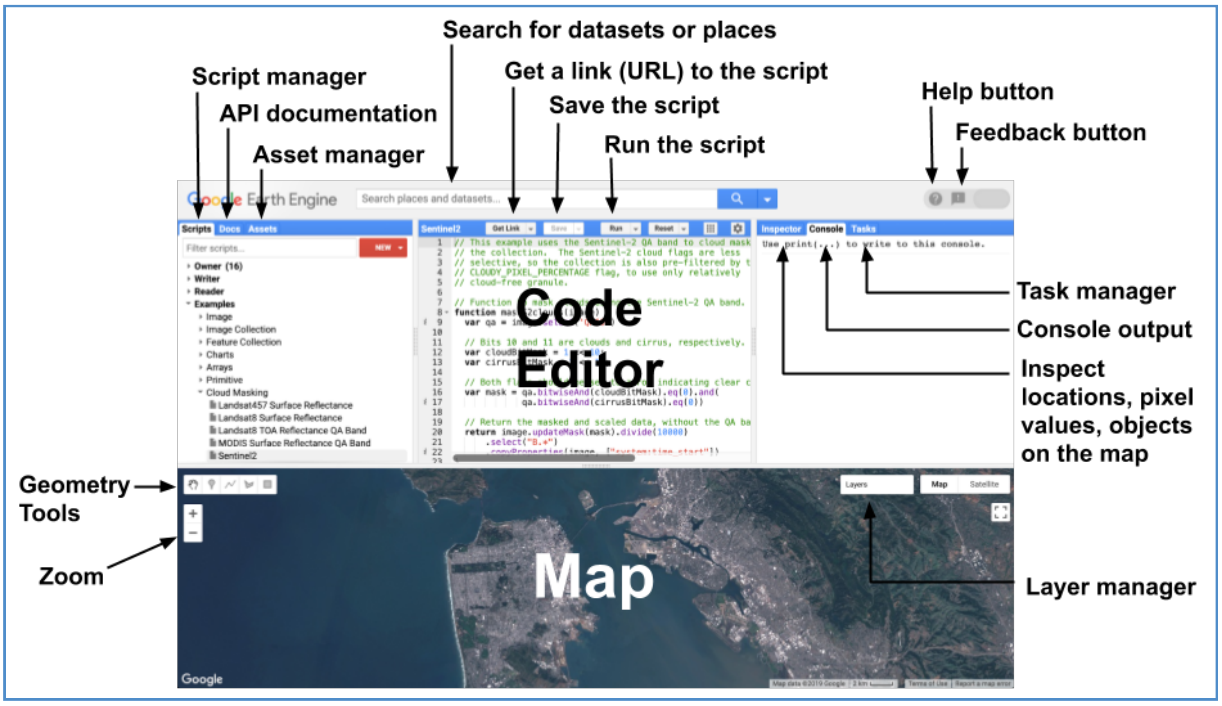


Figure 4. Interface of Google Earth Engine

Source: Google Earth Engine. 2022. In: Earth Engine Code Editor. https://developers.google.com/earth-engine/guides/playground

Code editor panel

- The editor panel is where you write and edit your Javascript code.
- Note that the run button executes the code.

Right panel

- Console tab for printing output.
- Inspector tab for querying map results.
- Click on the map and note that there is a scale in meters associated with the zoom level.
- Tasks tab for managing long-running tasks.

Left panel

- Scripts tab for managing your programming scripts. These are git repos hosted at Google.
- Docs tab for accessing documentation of Earth Engine objects and methods, as well as a few specific to the code editor application. This is the definitive API reference and is populated by the server.
- Assets tab for managing assets that you upload. You get 250 GB free.

Interactive map
- For visualizing map layer output.
- Note layer controls.
- Note the geometry tools.

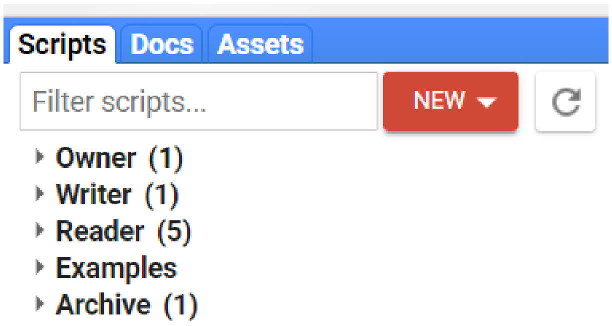
Search bar
- For finding datasets and places of interest.

Get link button
- A static snapshot of the code editor at the time the button is clicked. If you change the code, get a new link. You can email these around for easy collaboration.

Help menu
- User guide - reference documentation.
- Help forum - Google group for discussing Earth Engine.
- Shortcuts - Keyboard shortcuts for the code editor.
- Feature tour - overview of the code editor.
- Feedback - for sending feedback on the code editor.
- Suggest a dataset - GEE intention is to continue to collect datasets in a public archive and make them more accessible.

EXERCISE: CLICK ON THE FEATURE TOUR AND TAKE A TOUR OF THE CODE EDITOR

4. The script manager



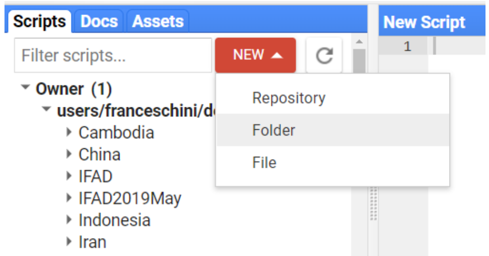
The scripts tab is next to the API docs in the left panel of the code editor. The script manager stores private, shared and example scripts in Git repositories hosted by Google. The repositories are arranged by access level, with your private scripts stored in a repository you own in the owner folder: users/username/default. You (and only you) have access to the repositories in the owner folder unless you share them with someone else. The repositories in the writer folder are repositories for which write access has been granted to you by their owner. You can add new scripts to, modify existing scripts in, or change access to (you may not remove their owner) the repositories in the writer folder. The repositories in the reader folder are repositories for which read access has been granted to you by their owner. The examples folder is a special repository managed by Google which contains code samples. The archive folder contains legacy repositories to which you have access but have not yet been migrated by their owner from an older version of the script manager. Search through your scripts using the filter bar at the top of the scripts tab.

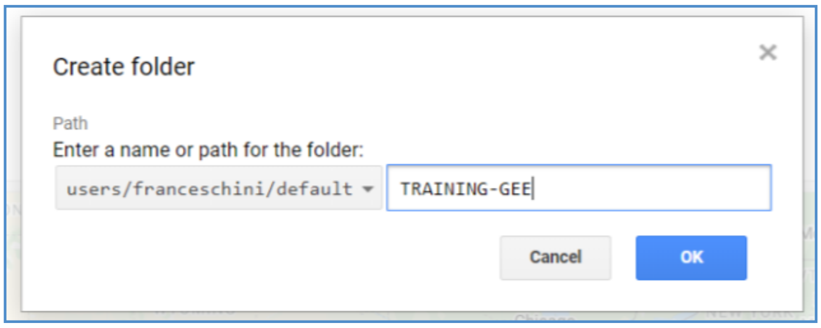
EXERCISE: CREATE A FOLDER

We will create a folder to store the scripts of the training:

1. Click on the NEW button.
2. Select Folder.
3. Type: TRAINING-GEE on the dialog that will appear.

The folder is now created in your default root directory.





EXERCISE: CREATE THE “Hello World” SCRIPT

https://code.earthengine.google.com/18964ca9f7214c7ebeda2a1ea8ffdabb

In the script panel, type the following:

```
var greetings = "Hello World"; //String
var number = 1; //Number
print (greetings);
var list = [2.6, 8, -3];
print(list[2]);
var dictionary = {
a: 'Hello',
b: 10,
c: 0.1343,
d: list
};
print(dictionary.a);
```

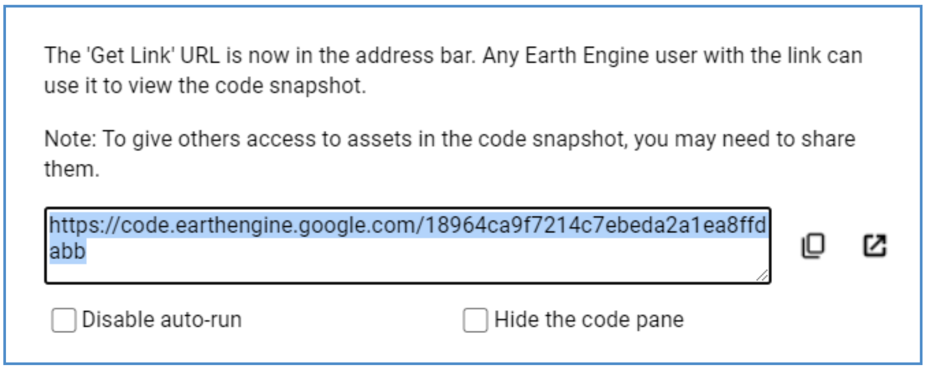
Then, click on the run button

There are few concepts that this script highlights:

- each variable has to be declared with the var operator;
- string characters are within quotes (single or double but keep consistency);
- lists are between square brackets and comma-separated;
- dictionaries are within braces and with couples of key:values;
- each line ends with a semicolon;
- outputs are written in the console panel;
- double slash is used to comment.

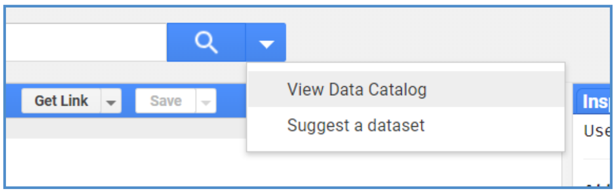
Then, click on the SAVE button and give a name to the script. You can specify in the name a subfolder of the repository.

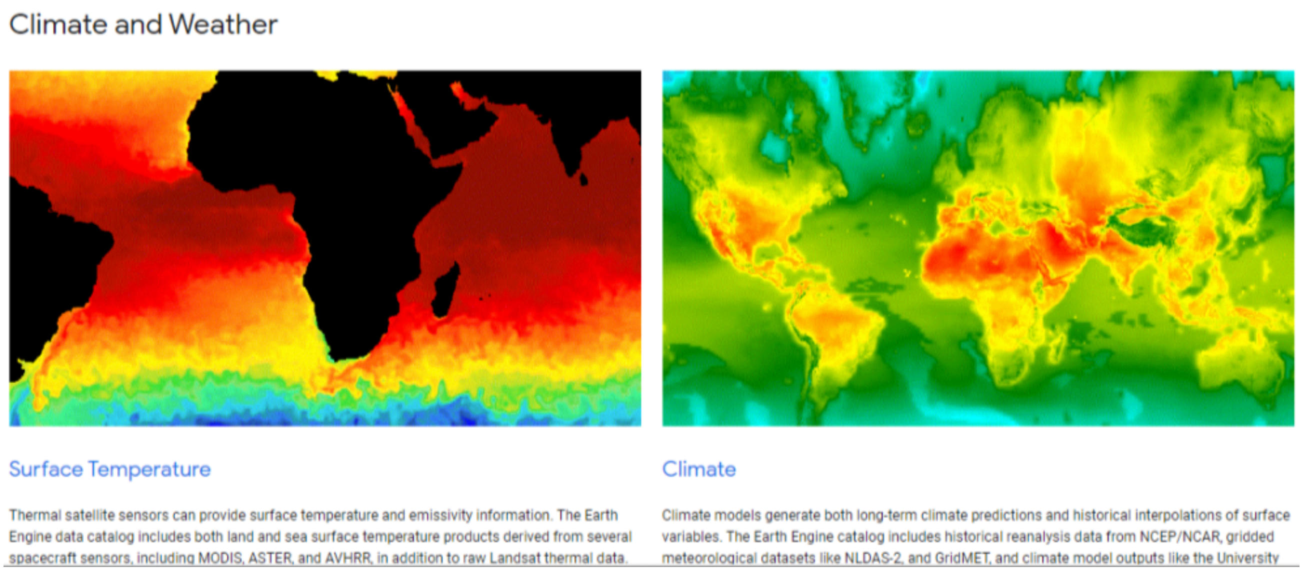
The Get Link button, allows to generate an URL link of the script that can be shared with other users



5. Explore the data catalog

To open and explore the data catalog (i.e. the data uploaded and accessible from GEE), you have to click on the arrow to the right of the search text box, and select view data catalog. This brings you to the data catalog https://developers.google.com/earth-engine/datasets/ which contains a description of the available datasets by theme.





Climate and Weather

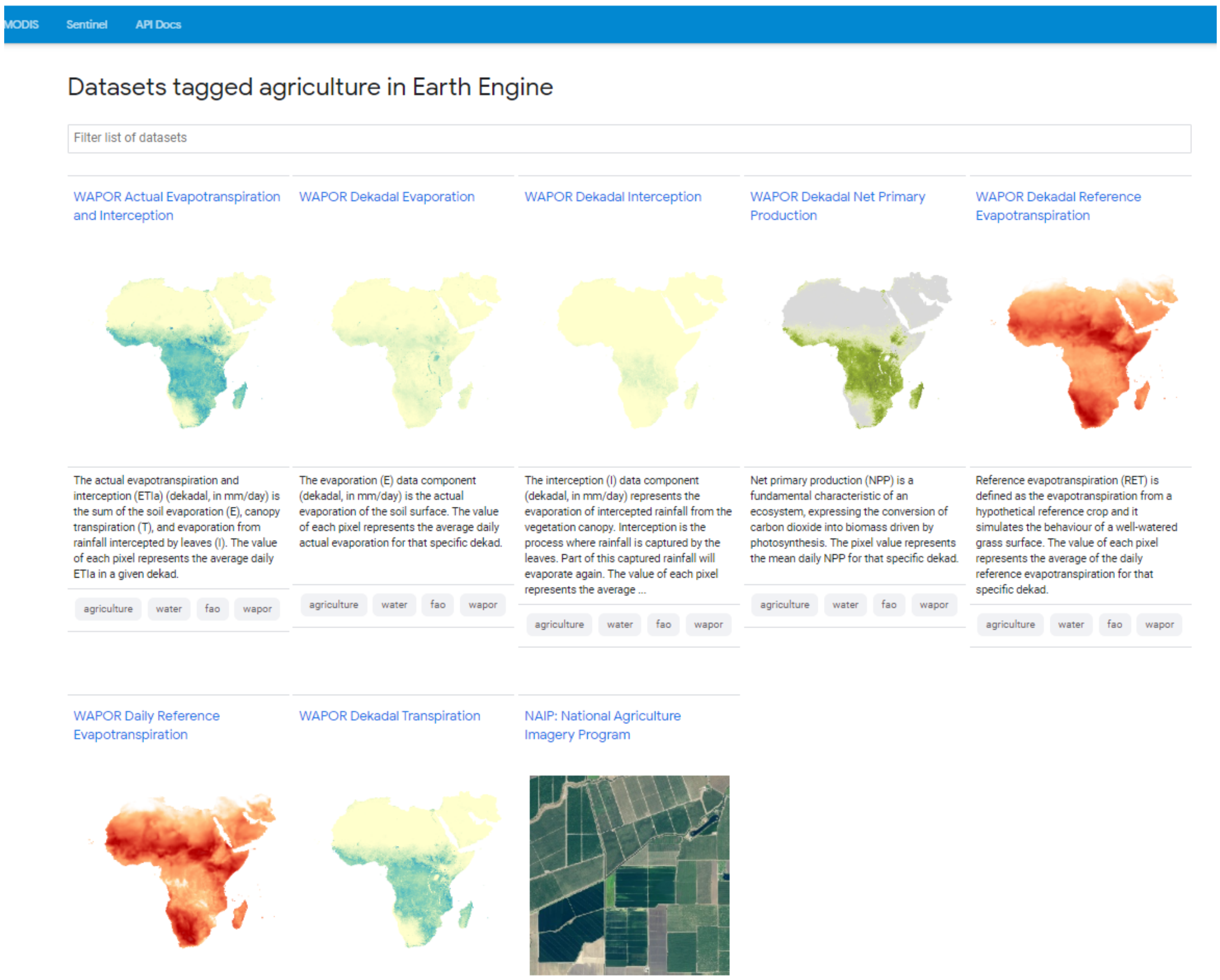
Surface Temperature

Thermal satellite sensors can provide surface temperature and emissivity information. The Earth Engine data catalog includes both land and sea surface temperature products derived from several spacecraft sensors, including MODIS, ASTER, and AVHRR, in addition to raw Landsat thermal data.

Climate

Climate models generate both long-term climate predictions and historical interpolations of surface variables. The Earth Engine catalog includes historical reanalysis data from NCEP/NCAR, gridded meteorological datasets like NLDAS-2, and GridMET, and climate model outputs like the University

Collections are both in raster or vector format and may represent a time-series stack of global or regional data up to one single layer of information. Main collections (MODIS, Sentinel, and Landsat) have their own specific page, alternatively data can be seen by tags (keywords).



MODIS Sentinel API Docs

Datasets tagged agriculture in Earth Engine

Filter list of datasets

WAPOR Actual Evapotranspiration and Interception

The actual evapotranspiration and interception (ETIa) (dekadal, in mm/day) is the sum of the soil evaporation (E), canopy transpiration (T), and evaporation from rainfall intercepted by leaves (I). The value of each pixel represents the average daily ETIa in a given dekad.

agriculture water fao wapor

WAPOR Dekadal Evaporation

The evaporation (E) data component (dekadal, in mm/day) is the actual evaporation of the soil surface. The value of each pixel represents the average daily actual evaporation for that specific dekad.

agriculture water fao wapor

WAPOR Dekadal Interception

The interception (I) data component (dekadal, in mm/day) represents the evaporation of intercepted rainfall from the vegetation canopy. Interception is the process where rainfall is captured by the leaves. Part of this captured rainfall will evaporate again. The value of each pixel represents the average ...

agriculture water fao wapor

WAPOR Dekadal Net Primary Production

Net primary production (NPP) is a fundamental characteristic of an ecosystem, expressing the conversion of carbon dioxide into biomass driven by photosynthesis. The pixel value represents the mean daily NPP for that specific dekad.

agriculture water fao wapor

WAPOR Dekadal Reference Evapotranspiration

Reference evapotranspiration (RET) is defined as the evapotranspiration from a hypothetical reference crop and it simulates the behaviour of a well-watered grass surface. The value of each pixel represents the average of the daily reference evapotranspiration for that specific dekad.

agriculture water fao wapor

WAPOR Daily Reference Evapotranspiration

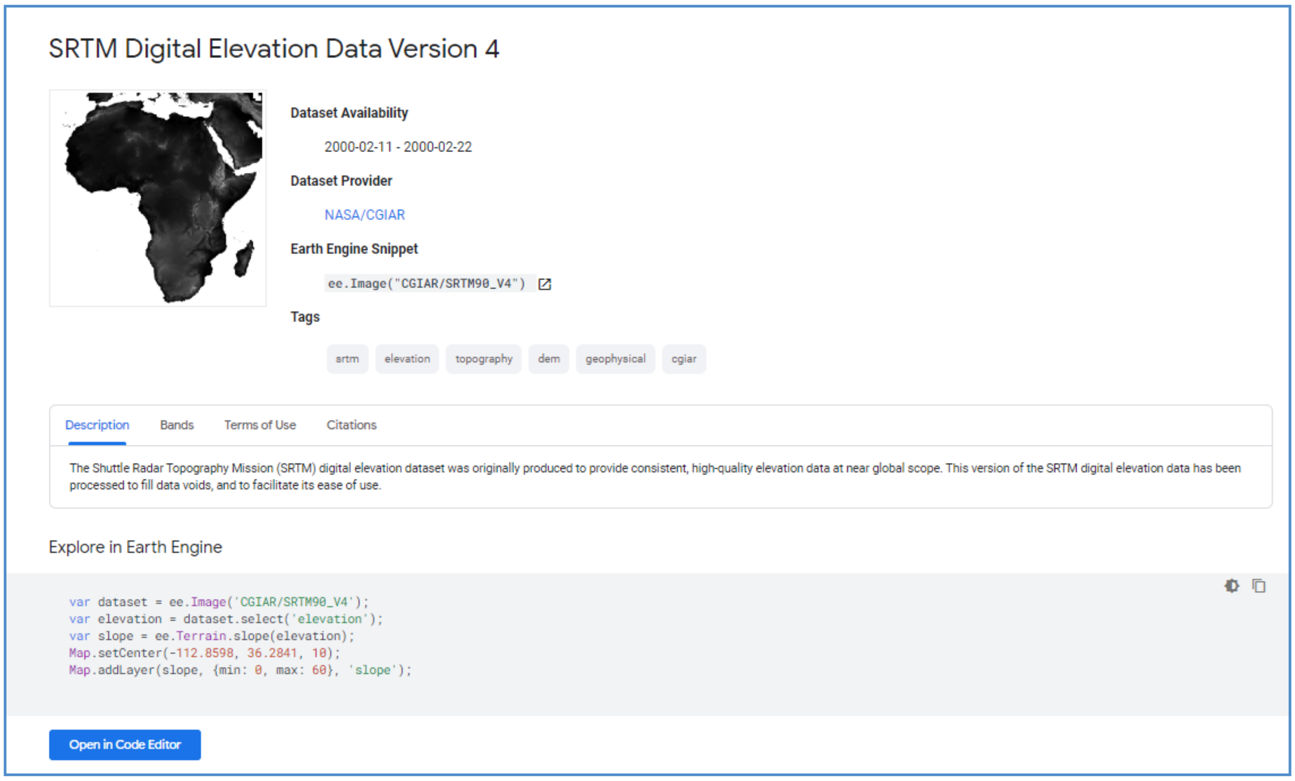
WAPOR Dekadal Transpiration

NAIP: National Agriculture Imagery Program

EXERCISE: FINDS ALL THE DATASETS RELATED TO “Elevation”

1. On the data catalog, click on browse by tags.
2. Select Elevation.

Once a dataset has been located, you can click on it to get the dataset page. The dataset page contains many interesting information:



- Dataset availability: For time-series collections is a crucial information. Data in GEE are uploaded respect to the data of acquisition with a limited lag of time.
- Dataset Provider: The owner of the dataset.
- Earth Engine Snippet: The code that identify that particular dataset.
- Tags: Keywords to locate the dataset.

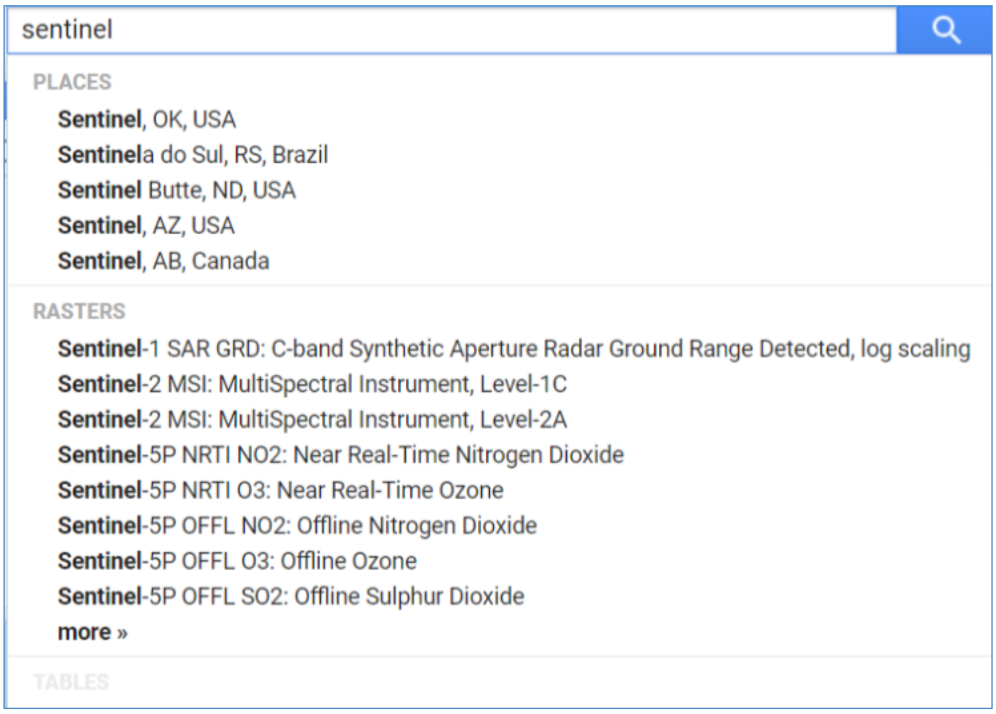
Then, you have a brief description of the dataset and the number of bands in each image, terms of use and citations.

Finally, you have a simple script which may contains some useful code to display and pre-process your image.

EXERCISE: FIND THE SENTINEL-2 SURFACE REFLECTANCE COLLECTION

Can you say the difference between an image and an image collection?

An alternative, is to Search a dataset directly from the search text box of the code editor.



Once a dataset is located, you can click on it to have a description (similar to the data catalog) or import it into a script.

EXERCISE: IMPORT THE SRTM (ELEVATION) AT 30 METER RESOLUTION FROM THE SEARCH TEXT BOX

1. On the search box, write “SRTM”.
2. You can explore the description of results to understand the difference between the datasets.
3. Once located, click on Import (at the right of the name) to Import the data in the script.

The results of importing datasets to your script are organized in an imports section at the top of your script, hidden until you import something.
To copy imports to another script, or convert the imports to Javascript, click the icon next to the imports header and copy the generated code into your script. You can delete the import with the delete icon.

EXERCISE: CREATE A NEW SCRIPT, CALLED ELEVATION

Copy the following code:

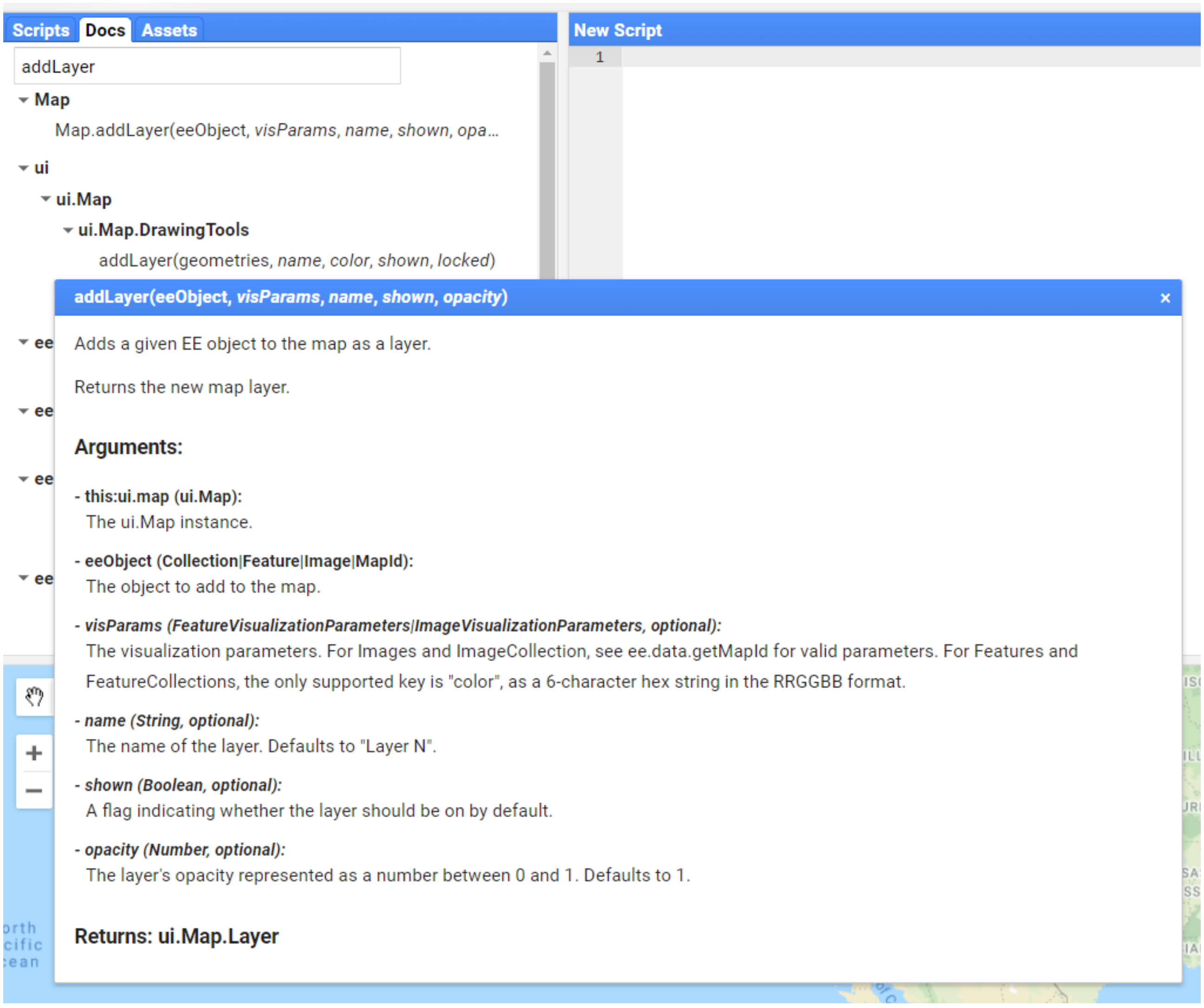
```
var image = ee.Image(“USGS/SRTMGLI_003”)
```

6. Display an image

In GEE, images are composed of one or more bands and each band has its own name, data type, scale (i.e. resolution), mask and projection. Each image has metadata stored as a set of properties.

If you have run the script of elevation, nothing will be displayed in the map window. In reality, nothing will happen at all. With the previous code, you created a variable named image where you have placed the Image corresponding to the specified code but you have not explained to the editor what you want to do with the image.

To display the image in the map panel, you have to add the layer to the map window. Type in the docs panel “addlayer” and click on the addLayer function. You will see a description of the function and the arguments required: the first one is mandatory (the object to map), while the others (in italic) are optional and add specific behavior to the result



Now, type in the script the following instruction:

```
Map.addLayer(image);
```

You will notice that the image is now displayed in the map panel, but the layer is just called "Layer 1" and the display does not look attractive (is a gray scale). You can modify the visualization parameters and add a title to your layer by specifying additional parameters. visualization parameters allow to control how an image is displayed on the map.

There are two ways of setting the visualization parameters:

- from the script; Map.addLayer(image);
- dynamically from the map window and then imported in the script.

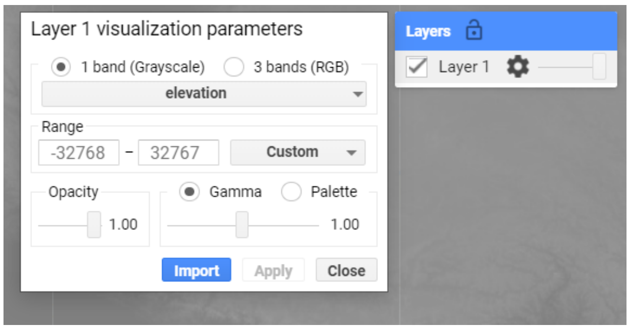
Several parameters can be set. All the settings are defined within a dictionary, that means the whole settings are enclosed by braces {} and each parameter (key) is specified, followed by colon and the value of the parameter. The value can be a single number or a list. When is a list, it needs to be enclosed by brackets [].

Visualization parameters for `Map.addLayer()`

Parameter	Description	Type
bands	Comma-delimited list of three band names to be mapped to RGB	list
min	Value(s) to map to 0	number or list of three numbers, one for each band
max	Value(s) to map to 255	number or list of three numbers, one for each band
gain	Value(s) by which to multiply each pixel value	number or list of three numbers, one for each band
bias	Value(s) to add to each DN	number or list of three numbers, one for each band
gamma	Gamma correction factor(s)	number or list of three numbers, one for each band
palette	List of CSS-style color strings (single-band images only)	comma-separated list of hex strings
opacity	The opacity of the layer (0.0 is fully transparent and 1.0 is fully opaque)	number
format	Either "jpg" or "png"	string

```
var vizPar = {bands:["elevation"],min:0,max:8000};
```

EXERCISE: CHANGE THE VISUALIZATION PARAMETERS OF A LAYER



1. Click on the icon next to the layer name. This will open its visualization parameters.
2. Change the range of the data (e.g. put 0 and 5 000).
3. Change the color to palette and experiment with a range of colors of your choice.
4. Once done, click on the import button.
5. Copy the imported code directly into your script.
6. Modify the script, by adding a second parameter (the imported visualization parameters) and a third parameter with the title of the layer. Each parameter must be separated by a comma.
7. Run the script again

7. Access the metadata of an image

Images (and other data formats such as image collections) have metadata that describe the technical content (for instance bands, resolution, acquisition time) as well as description of content, source, terms of uses. All this information is defined as property in GEE. Set of properties is different for each image.

There are two ways to access the properties of an image:

- printing the image to the console;
- from the script, using the propertyNames() function.

EXERCISE: ACCESS THE METADATA OF AN IMAGE

In your script type print(image)

You will see that in the console panel, information about the image will appear. The property section, will contain all the metadata of that image

In your script type:

```
var metadata = image.propertyNames()
print(metadata)
var property = image.get(property)
```

8. The inspector\console\tasks panel

Metadata of the image can be written with the print statement.

Add this line to the code:

```
print(image);
```

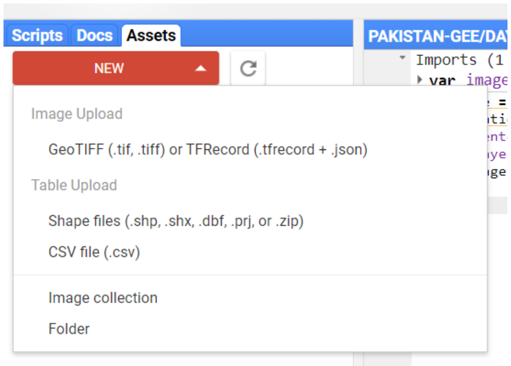
The output will be shown in the console panel, where all the textual and graphics (e.g. charts) are shown.

The inspector panel allows to click on the map panel to have the value of layers (similarly to the identify tool on ArcGIS).

Tasks allow to run specific methods that are or might be time-consuming such as the import and export of data.

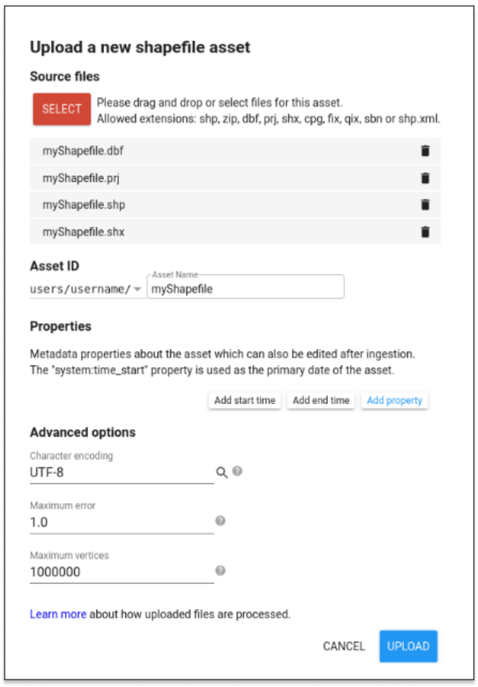
9. Upload a Shapefile as asset

In many cases, it may be required to upload specific information to use in the scripts. For instance, we want to clip the image only to specific boundaries of interest. To this end, we first need to upload a Shapefile with this boundary. The file will appear on the list of assets.



EXERCISE: UPLOAD A SHAPEFILE

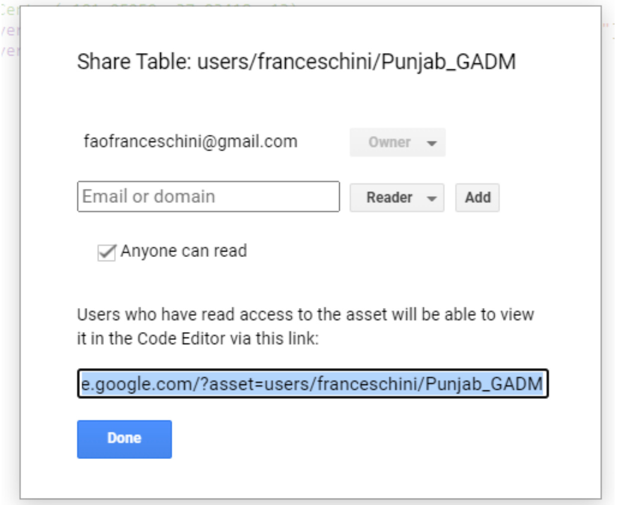
1. Click on the asset tab on the top-left panel.
2. Select Shapefiles (under Table Upload).
3. On the upload dialog window, click on select to choose the files to be uploaded.
4. Assign a name to the asset.
5. Click on upload.



When the uploading is completed, it will appear in the list of assets (you may need to refresh the list). Then, click on Import into the script to use the Shapefile. The uploaded asset is private and only accessible by the user that uploaded the Shapefile. If the script needs to be shared with other colleagues, the asset will not be accessible and the code will not work. To make the asset accessible to other users:

6. In the asset panel, move next to the asset to share (three logos will appear) and click on share.

- In the pop-up window, check **"Anyone can read"**.



10. Clip the image to the area of interest

In GEE, an image is presented uniform around the globe, however is internally stored in single tiles. If we want to display a particular region of interest, there are many possible ways. One solution is to clip the image, by using a reference geometry.

EXERCISE: CLIP AN IMAGE

Add the following code:

```
var clippedElevation = image.clip(AOI);
```

And then:
- display the new image on the map panel;
- adjust the visualization parameters to a better range.

11. Apply an algorithm to an image

You can use specific algorithms to apply to an image to generate other outputs. These functions can be mathematical operations (band a + band b), morphological operations (focal functions), edge detections and many others. In our next example, we will use a slope function to calculate a slope layer from a Digital Elevation Model (DEM).

EXERCISE: CALCULATE THE SLOPE

1. On the top right panel, click on docs.
2. Type slope (this will produce a number of results).
3. Apply the slope function to the DEM of the previous exercise.

EXERCISE: CALCULATE THE HILLSHADE AND ADD IT TO THE ELEVATION

1. On the top right panel, click on docs.
2. Type hillshade (this will produce a number of results).
3. Apply the hillshade function to the DEM of the previous exercise.
4. Define a visualization parameter with an opacity = 0.2 and overlay it to the elevation image.

12. Save the results to your computer

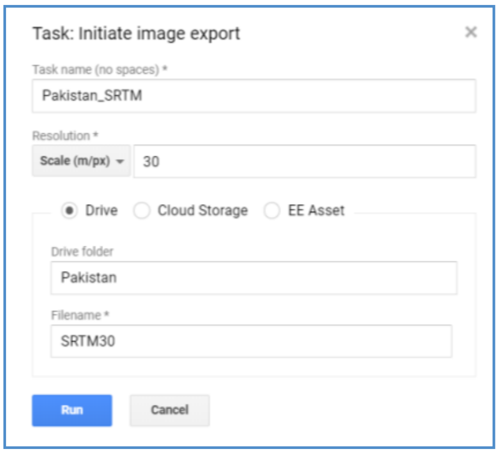
Final outputs of an analysis such as images, map tiles, tables and video can be exported from GEE. The exports can be sent to:
- your Google Drive account;
- a Google Cloud Storage;
- a new GEE asset.

EXERCISE: EXPORT AN IMAGE

Add the following code:

```
Export.image.toDrive({

image: clippedElevation, //the image to export

description: 'Pakistan_SRTM', //description of the task

scale:30, //The resolution

folder: 'Pakistan', //The folder in Google Drive

fileNamePrefix: 'SRTM30', //The prefix of the raster

maxPixels: 1e13 //Max number of pixels to export

});
```

Then, click on run, next to the task created. The export may take several time. At the end the image will be available on your Google Drive.



https://code.earthengine.google.com/2649f48fb35e64220e5a5be402839aae

13. Display an image from an image collection

An image collection, contains a set of individual images. We will look at how we can select multiple images within an image collection, but we can also access an individual image within an image collection. Type the following code and comment the results.

EXERCISE: DISPLAY AN IMAGE FROM AN IMAGE COLLECTION

```
var image =

ee.Image("COPERNICUS/S2_SR/20200622T055641_20200622T060249_T42RXT");

var vp = {bands:["B8","B4","B3"],min:500, max:3500};

Map.addLayer(image,vp,"Sentinel-2");

print(image);
```

https://code.earthengine.google.com/a5e0fef3956fdc74b9ff9f1d523f9389

A common task in GEE is to find existing scripts and modify them to be used in your actual work. You will use the image you selected in the previous example to customize an existing script.

EXERCISE: CUSTOMIZE AN EXISTING SCRIPT

1. Find the script Canny Edge detector in the examples.
2. Find how the Canny Edge detector works.
3. Update the script to make it work with your image.

https://code.earthengine.google.com/7210f28f984a12930c307a1df2ae7f17

14. Remote sensing data

During most of the training, we will focus on data from the Copernicus program of the European Space Agency (ESA). Let's first have a look at the program and at some basic related information related.
Watch the following video for introduction to Copernicus

https://dlmultimedia.esa.int/download/public/videos/2014/03/018/1403_018_AR_EN.mp4

Sentinel-1 (S1) is an imaging radar mission providing continuous all-weather, day-and-night imagery at C-band. The S1 constellation provides high reliability, improved revisit time, geographical coverage and rapid data dissemination to support operational applications in the priority areas of marine monitoring, land monitoring and emergency services.

S1 key characteristics are:
- Two satellites (S1A, S1B).
- Multiple acquisition modes.
- Band C.
- Dual polarization mode.
- Spatial resolution up to 10 m.
- Revisit around six days (for both satellites).

Sentinel-2 (S2) provides high-resolution optical imaging for land services (e.g. imagery of vegetation, soil and water cover, inland waterways and coastal areas).

S2 key characteristics are:
- Two satellites (S2A, S2B).
- Multi-spectral data with 13 bands.
- Spatial resolution up to 10 m.
- Revisit around five days (for both satellites).

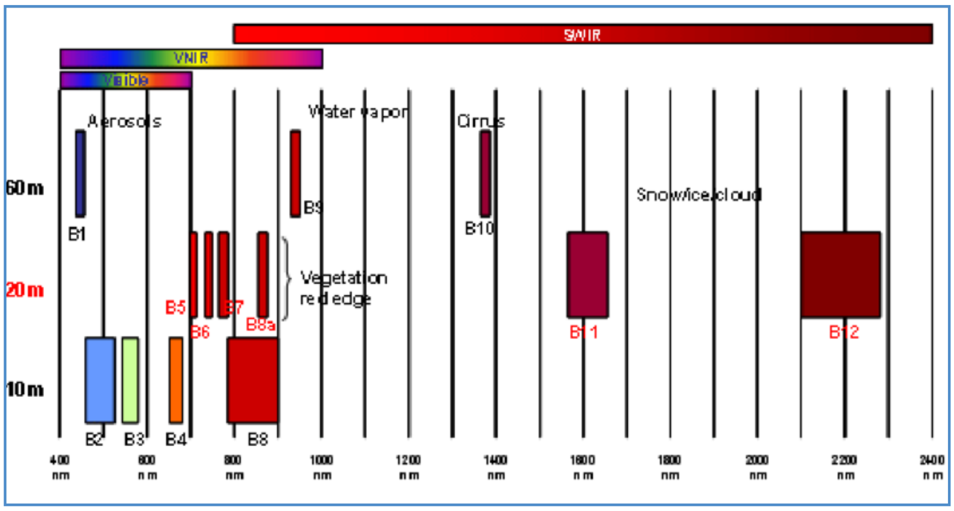


Figure 5. Bands of Sentinel-2

Source: European Space Agency (ESA). 2022. In: Sentinel Online. https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial

15. Image collections

Now that you are familiar with an image is time to move onto the main data structure you will deal with. As the name suggest, an image collection is just a collection……of images. Each image has a series of attributes that identify and differentiate it within the collection and the user can interact with these attributes to filter or limit the images within the collection.

While is useful to export a single image, in general you will not export an image collection (not even a portion of it). Generally the image collections are used within algorithms, some functions are applied, and the result of the workflow can produce a format (a single image, a video, a table) that is more suitable to export.

Can you think to a possible image collections? Remember how satellite works. They collect information travelling across their orbit. This information, converted in a digital raster format is then processed and transmitted as individual tiles, where each tile has its own date of acquisition, location and metadata. The image collection is not a perfect place to store all these images?

EXERCISE: FIND THE SENTINEL-2 IMAGE COLLECTION AND IMPORT IT IN A NEW SCRIPT

1. Create a new script in the training folder.
2. Go to the data catalog and click on the Sentinel Tab.
3. Locate the surface reflectance dataset and click on it.
4. Copy the following code in the new script.

```
var S2 = ee.ImageCollection("COPERNICUS/S2_SR");
```

The last exercise allowed you to create a variable that contains the whole archive of images acquired from the beginning. It is not handy to work on the whole collection, while you may be interest only to a specific period, or location or quality.

Let's start to filter (limit the number of images according to some criteria) the image collection.

15.1 Filter by a period

Typically, you will not need to work on the whole archive, but rather in a selection of only images acquired across a defined interval. Although a generic filter() method exists, the method filterDate() is more specific for this purpose.

EXERCISE: SELECT ONLY SENTINEL-2 IMAGES ACQUIRED IN 2020

Type the following code:

```
var S2_2020 = ee.ImageCollection("COPERNICUS/S2_SR")
.filterDate('2020-01-01', '2020-08-07');
```

On the previous code, you may have noted that the method filterDate has been assigned to the selected image collection just concatenating the two functions with a dot. This is typical in GEE and allows to avoid to create multiple variables. Just be careful that the semicolon must be assigned just at the end of the functions.

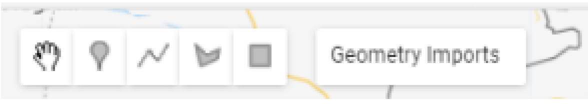
Use the concatenation to create variables for each line of code. Below is how would appear the selection of images without concatenation:

```
var S2 = ee.ImageCollection("COPERNICUS/S2_SR");
var S2_2020 = S2.filterDate('2020-01-01', '2020-08-07');
```

15.2 Filter by a location

EXERCISE: RE-SELECT ONLY SENTINEL-2 IMAGES COVERING THE AREA OF INTEREST

In most of the cases, you will also need to refine your selected images only to those that overlap specific geometries. This time, we will use a geometry digitized from the screen. Click on Draw a shape from the top-right tools of the map window:



Draw a polygon within the area to analyze. Click on the last vertex to close the geometry. Once the geometry is closed, you will see within the Import area that the geometry appeared. You can now copy and paste the new geometry in your script. Then:

1. Rename the new geometry as “AOI”.
2. Type the following code to refine the search:

```
filterBounds(AOI)
```

Note that filterBounds does not clip the images; it just selects those images that overlap the geometry.

15.3 Filter by metadata

Each image contains a number of metadata. We will use some metadata to refine the selection of images. To view the metadata of a single image, you have to refer not to the ImageCollection (which has its own metadata) but rather to a specific image within the ImageCollection. To do this you can pick the first image within the image collection by using the .first() method. Then, you can print the image to the console.

Then, we will use the filterMetadata() method with the 'CLOUD_COVERAGE_ASSESSMENT' property. To understand how the filterMetadata works, type the function in the top-right doc tab.

EXERCISE: RE-SELECT ONLY SENTINEL-2 IMAGES WITH CLOUDS < 80%

1. Check the metadata of one image with the .first() method on the image collection.
2. Type filterMetadata in the doc tab to understand which operator to use.
3. Type the corresponding code to refine the selection.

EXERCISE: COUNT THE NUMBER OF MATCHING IMAGES FROM EACH FILTER

In order to count how many individual images matches each filter, you can use the size() function after a filter is selected. You will see that after each filter is applied the resulting number of matching images will decrease.

15.4 Select one image from an image collection

While, is theoretical possible to add to the map layer an image collection (but you will easily exceed the computation time receiving an error) is simpler to extract an image from the image collection. We have seen already how each image has its unique ID that can be used to access and display the specific image. There are other options that can be used dynamically from the script: the first() function, returns the first image from the image collection. The sort() function returns the images in ascending or descending order, based on a property. Combining sort and first, you can return the most recent image matching your filters.

```
var mostRecent = S2_2020.sort('system:time_start', false).first();//False is in descending order
```

Using the acquired information and try to display the image with the least cloud coverage in your area of interest.

15.5 Filter the same period over multiple years

In some circumstances, it may be useful to select the same period across multiple years. While is possible to create several image collections and then merge them, the calendarRange() function allows to accomplish this task easily. The function goes over the image collection and filter the images that are within the assigned range. The range can be expressed in various formats: e.g. year, month, day of year, etc.

To filter an image collection of images acquired in a specific month of a specific range of years, you can apply the following example:

```
var S2_TS = ee.ImageCollection("COPERNICUS/S2")
.filterDate('2010-01-01', '2020-01-01')
.filter(ee.Filter.calendarRange(7,7,'month'))
```

15.6 Select bands

So far, we have limited the number of images from the whole image collection, only to those that matches some spatial, temporal or textual attributes. Now, we want to select only some of the bands from those images. To this end, we will use the select() method over the filtered image collection. The select method requires a list of band names that will be selected.

Type the following code to select only specific bands.

```
.select(["B2","B3","B4","B8" ]);
```

Finally, as shown before, you may need to adjust the visualization parameters to properly show the image. Use the false color combination (B8, B4, B3) to display the image with correct stretching.

https://code.earthengine.google.com/4c7fa899a2faa9e6b732d6acfcd3da5a

16. Feature collections

In previous examples, we have seen how we can create a geometry (a point or a polygon) and import it into a script. The geometry object requires a type to be specified (line, polygon, point) and couples of coordinates. We have also seen, how we can use these geometries to select images (filterBounds) or to apply specific functions (e.g. clip).

In addition to geometries, in GEE you also have features. A feature is an object which has a geometry property and a dictionary of properties to store additional attributes. A feature collection is simply a set of individual features.

ee.Geometry, ee.Feature and ee.FeatureCollection are the constructor for each of the object.

Table datasets exist in the data catalog and can be added in a script as a feature collection.

1) Geometry (type, coordinates)

2) Feature — Geometry (type, coordinates)
  Dictionary of properties

3 Feature collection
  feature
  ... ..
  feature

4) Table dataset
ee.FeatureCollectio ID')

EXERCISE: CREATE GEOMETRY, FEATURE AND FEATURE COLLECTION

1. Add two points as geometry in your script.
2. Create two features, using the geometry you created, adding Province and Enumerator.
3. Create a feature collection with the two features.

Feature collections can also be added from table datasets and from assets.

Similarly to image collections, feature collections can be added to the map, with specific visualization parameters, however most of parameters are a bit more complex to set.

https://code.earthengine.google.com/18f46ad5dfa3f66f5ba6f8517ee04575

16.1 Specify the color

The color of the feature collection can be changed using the color key, similarly to assignments of parameters in a image collection (by default the feature collection will be displayed with solid black lines and semi-opaque black fill).

16.2 Use the featureCollection.draw

With featureCollection.draw three parameters can be defined:

- color: color of the collection;
- pointRadius: the radius in pixels of the point markers;
- strokeWidth: the width in pixels of lines and polygon borders.

Note that feaureCollection.draw is a function of a feature collection, hence is not part of a visualization parameter.

16.3 Use image.paint()

For additional control on visualization parameters, you need to create an empy image, and on that image you paint the feature collection, specifying color and width. The advantage is that you can also use an attributes of the image collection for color and width:

```
var empty = ee.Image().byte();
var GAUL_IMAGE = empty.paint({
  featureCollection: PakistanAdm1,
  color: 1,
  width: 1
});
Map.addLayer(GAUL_IMAGE,{palette: 'FF0000'},"Pakistan adm1");
```

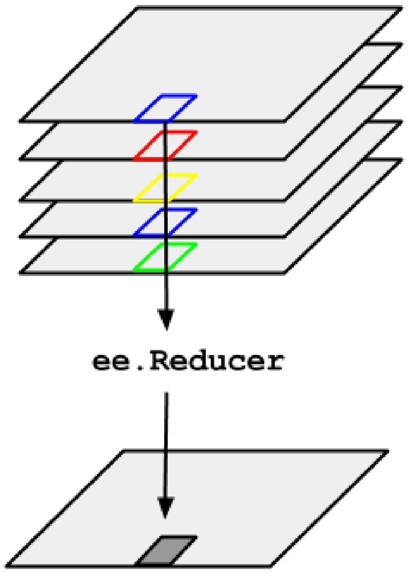
Similarly to an image collection, you can filter the metadata of a feature collection (its properties), to select only specific features.

EXERCISE: DISPLAY LEVEL-1 ADMINISTRATIVE BOUNDARIES OF PAKISTAN

1. Select GAUL admin 1 from the data catalog.
2. Identify the property to select Pakistan.
3. Create a new feature collection, only with the selected features.
4. Add the image collection to the map.

17. Reducers

Reducers are a specific group of functions applied to an image collection that aggregate data over time, space or in other ways. Often, the result of a reducer is a single image.



Consider the example of needing to take the median over a time series of images represented by an ImageCollection. To reduce an ImageCollection, use imageCollection.reduce(). This reduces the collection of images to an individual image. Specifically, the output is computed pixel-wise, such that each pixel in the output is composed of the median value of all the images in the collection at that location.

To apply a reducer that calculate the mean pixel over an image collection, you can type the following:

```
var imageCollectionMean = imageCollection.reduce(ee.Reducer.mean());
```

Note that for some common reducing functions, a shortcut method is possible.

```
var imageCollectionMean = imageCollection.mean();
```

EXERCISE: REDUCE THE IMAGE COLLECTION

On the already selected image collection, create three new images with min, max and median values.
Note also that the bands names of a reduced image change accordingly to the applied function.

https://code.earthengine.google.com/9b7c3b28a1a3cfb0396c1269038e791f

18. Calculate NDVI on an image

A common task on an image is the calculation of a vegetation index such as the NDVI. There are many ways to perform this task: we will use the normalizedDifference() function, that only requires the definition of the NIR and red band.
Type the following:

```
var ndvi = S2_2020median.normalizedDifference(['B8_median',
'B4_median']).rename('NDVI');
```

Note that when you reduce an image collection, the name of the bands has changed.
You can also add the calculated NDVI within the reduced image, using the addBands() function.
Type the following:

```
var S2_2020_NDVI = S2_2020median.addBands(ndvi);
```

https://code.earthengine.google.com/281fbda678b3d98e7e9fcba6b3b17c19

19. Apply a reducer over a feature collection

To calculate the statistics of an image over a geometry or a feature collection, the reduceRegion reducers can be applied. This reducers is applied over a specific image (not an image collection) and requires to define a number of parameters:

```
var meanDictionary = ndvi.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: geometry,
  scale: 10,
  maxPixels: 1e9
});
```

- Reducer is the function used to calculate the statistics.
- Geometry is the geometry used to define the region(s).
- The scale express the spatial resolution of an image.
- maxPixels is the maximum number of pixels to compute.

The result is stored as a dictionary that can then printed or exported.
https://code.earthengine.google.com/3c4d7bde7d145eabf9a2a22eae7fca6d

20. Apply a mask on an image

In many cases, you will apply a process only to pixels that meet certain criteria. This is different from filtering images within an image collection, as the whole image is included or not in the analysis according to the filter, while with masks, we will consider specific pixels within an image.

Masking pixels is a two-steps approach. First, you will define the criteria to evaluate each pixel value. Second, you will apply the filter to the image. The result is a new image where masked pixels will be transparent and treated as no data.

There are many operators used for defining specific criteria. Probably the most common are the “great than” and “less than” operators (gte and lte) that evaluate whether a pixel is greater or lesser a specific threshold:

```
var filter = ndvi.gte(0.5);
```

Then, you will apply the filter to a specific image, with the updateMask():

```
var vegetation = ndvi.updateMask(filter);
```

The two steps can also be embedded in one single line of code:

```
var vegetation = ndvi.updateMask(ndvi.gte(0.5));
```

EXERCISE: FIND PIXELS WITH VEGETATION AND ELEVATION HIGHER THAN 250 METER

With the acquired information, try to develop a script that finds pixels with vegetated land (according to the mask of the previous chapter) AND with an elevation of 250 meter.

https://code.earthengine.google.com/7d357fea75b02a19dfb636fcbe024396

21. Calculate NDVI on a image collection

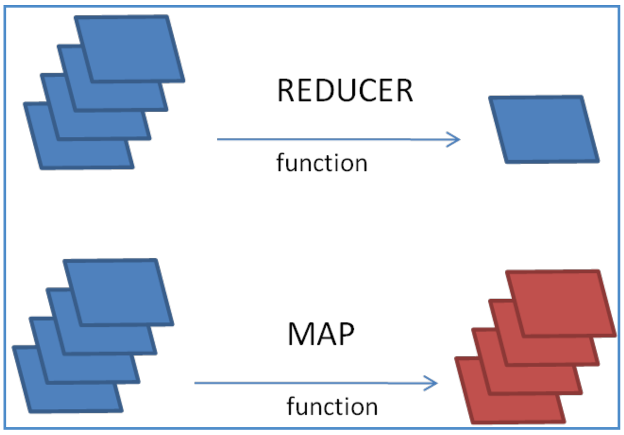
In many cases, you will need to apply a process to all the images within an image collection, and as a result, to have a new image collection, containing the new updated images. This is achieved by using the map function over the image collection. The map function is different from a reducer, where you apply a function over the whole image collection and as a result you obtain a single image.

The process that you will apply is defined within a function, which contains some instructions that will be returned as a result of the function.

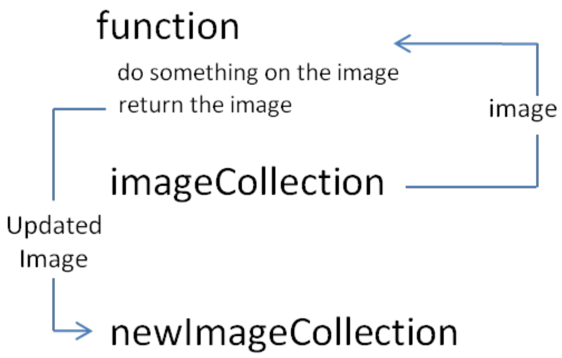
To map a function over a collection, you will need the following:

A declaration of the function: This is done using: the keyword function, followed by the function name and by the parameter on which you map the function (e.g. an image). Then, all the instructions are included by curly brackets. The last instruction is a return.

A map statement on a collection that invokes the declared function and define the new collection.



The following image presents a schema of how it works: you start from an image collection (a series of images) and you invoke an existing function, that apply some manipulation and return each image to the new function.



The same schema with proper syntax is as follows:

```
function function_name(image){
  do something
  return image;
 }

var newImageCollection = imageCollection.map(function_name);
```

Let's begin with a simple function. This function is applied to a collection of a numbers within a list. These numbers represent years. I want to create a new list with the number of years to reach 2080 for each number of the original list. Note that we will use a map over a list (not an image collection).

Let's define the initial list of years:

```
var startingYears = [2000, 2010, 2020, 2050];
```

Then, let's create the function:

```
function calcYearsTo2080 (year){
var difference = 2080 - year
return difference;
}
```

Finally, let's map the function over the list and print the results.

```
var yearsTo2080 = startingYears.map(calcYearsTo2080);
print(yearsTo2080);
```

The concept of mapping over an image collection, can be easily applied to generate an NDVI band on the images within an image collection.

EXERCISE: CALCULATE NDVI OVER A S2 IMAGE COLLECTION

1. Select an image collection of S2
2. Write a function to calculate the NDVI.

- Remember: you must first type function followed by the function name, the image by parenthesis (this is the parameter of the function) all encapsulated by curly brackets { }.
- Within the curly brackets, you can use the image.normalizedDifference(["B8","B4"]) function to generate the NDVI.
- Return the image back to the new image collection.

3. Map the function over the image collection and assign the result to a new imageCollection

The next example uses a more complex logic: it maps over a list of years, to select images within an image collection, apply a reduction and reselect some pixels. Note how the result of mapping over a list is another list. To convert the list to a new image collection, you have to apply a specific command.

```
//Define the image collection
var dataset = ee.ImageCollection('MODIS/006/MCD64A1').select('BurnDate');
//Define a list of years
var years = ee.List.sequence(2001, 2019);
//Map over each year
var results = years.map(function(year) {
  return dataset.filterDate(ee.Date.fromYMD(year, 1, 1), ee.Date.fromYMD(year, 12,
31)).sum().gte(1);
});
//Convert the results to an image collection
results = ee.ImageCollection(results);
//Map the layers
var burnedAreaVis = {
  min: 30.0,
  max: 341.0,
  palette: ['4e0400', '951003', 'c61503', 'ff1901']
};
Map.addLayer(dataset, burnedAreaVis, 'Burned Area');
Map.addLayer(results.count(), imageVisParam, "Frequency of burnt areas");
//Map.addLayer(results.mean(), {}, "results");
print(results);
```

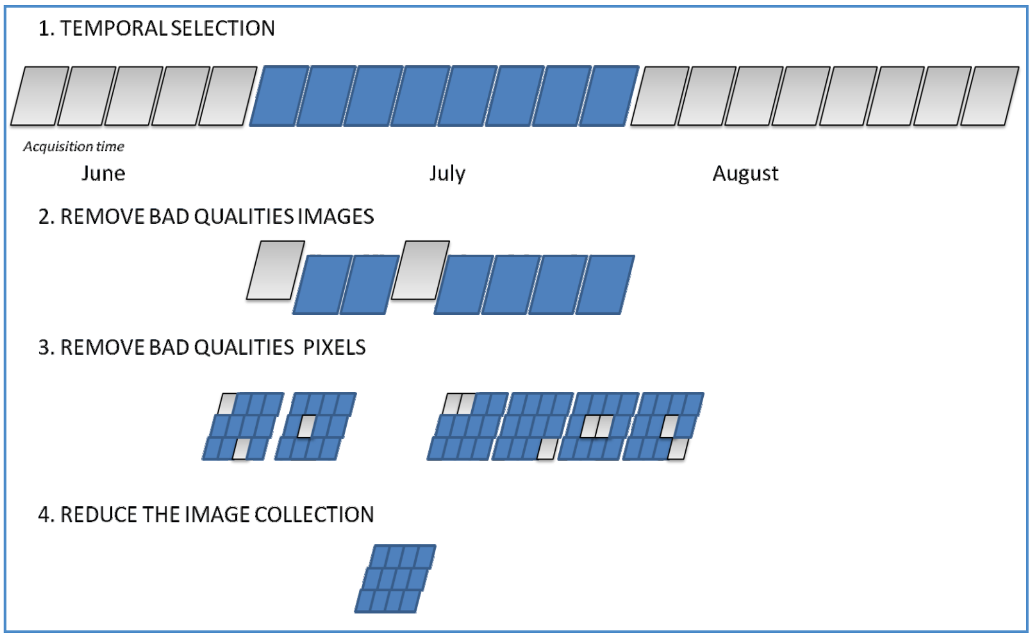
https://code.earthengine.google.com/03f23f0207a511f580bb7cdc6b2d1155

22. Cloud removal

When working with optical images for land cover assessment, one of the main pre-processing task, is to remove (or reduce) atmospheric effects caused by clouds, cloud shadows and other phenomena that may alter the signal recorded by the satellite. While is practical impossible to remove these effects on a single image, an increased availability of multiple images due to shortened revisit time and a redundancy of the information allow to define a strategy for cloud removals.

The first obvious step is to increase the acquisition time range to include in the image collection a larger number of images and hence to increase the probability to have cloud free images. This approach needs to be balanced with the expected temporal behavior of the features under investigation. If crop monitoring is the purpose, to have a wide period of analysis (e.g. a year), would result on degrading the information related to the change of spectral signature due to the phenology of crops. Typically, a multi-temporal monthly or seasonal window is suggested.

Then, metadata of the images can be used to remove from the image collections those images that are considered of bad quality. On the result image collection, all the pixels that are of bad quality can be masked out. Finally on the resulting pixels, a reduction can be applied to generate the final single free-cloud image to analyse.



EXERCISE: REMOVE CLOUDS FROM AN IMAGE COLLECTION

1. Select images from S2.
2. Sub-select the images based on metadata.
3. Remove cloudy pixels.
4. Apply a median reducer.

https://code.earthengine.google.com/eca8f6f59992509ecfdba933d529e2f3

23. Export a video from a reduced cloud removed image collection

Displaying an animated time lapse of an image collection can be an interesting assessment of pressures on land and variability of natural resources, in order to have an exploratory analysis of the area of interest.

Take 5 minutes, to visit some examples on the following link:

https://earthengine.google.com/timelapse/

You will then explore how a thumbnail is generated. The example uses the image.visualize function that produces an RGB or grayscale visualization of an image.

https://code.earthengine.google.com/01bbdd3b5622d279bcf97dd959c4b274

24. NDVI anomalies

An anomaly is defined as a deviation from a normal behavior. In crop monitoring, is a quantitative measure that expresses how different a variable at a certain place and time (e.g. rainfall, NDVI is from reference (i.e. normal) conditions. While anomalies look both at below and above normal conditions, in crop monitoring is important to look at below normal conditions (flood detection is one typical example that looks at above normal conditions).

Anomaly maps of NDVI can identify how and where vegetation conditions are below normal conditions, highlighting areas of potential concern, particularly when focusing on agricultural and pastoral land.

When considering normal conditions, we typically look at multi-annual averages: Long-Term Averages (LTA) usually refers to 30 years of observations (this is the standard for climate normals), while Short-Term Averages (STA) or Recent Average (RA) consider 5 or 10 years.

Anomalies can be calculated in different ways:

Absolute. Simple difference between current value and the normal average. The difference is expressed in the physical unit of the variable. (current value – average).

Relative. It is expressed as a percentage, where a value below 100% means that the variable is lower than normal conditions. It is the ratio between current value and normal average multiplied by 100. ((current value / average) * 100).

Standardized. It is the difference between current value and average, divided by the standard deviations for all the years. ((current value – average) / standard deviation(average)).

EXERCISE: STATISTICAL CALCULATION OF RAINFALL ANOMALY

Given a hypothetical value of 240 mm, in an Excel document calculate the three types of anomaly respects to a STA (200,220,200,240,230).

STA average = (200+220+200+240+230) / 5 = 218

Standard deviation = 17.89

Absolute anomaly = 240 – 218 = 22

Relative anomaly = (240/218)*100 = 110

Standardized anomaly = (240-218)/17.89 = 1.23

Table 1. NDVI anomalies and their units

Anomaly	Unit	Expresses	Advantages	Disadvantages
Absolute	mm	Absolute difference between current and normal.	Easy to interpret.	Not apparent how anomaly relates to normal rainfall.
Relative	%	Current as percentage of normal.	Easy to interpret.	Gives extreme results in areas that normally receive little rainfall during specific month.
Standardized	-	Number of standard deviations current is above/below normal level.	Takes into account the variability between years.	Interpretation is a bit more complex (unitless).

EXERCISE: CALCULATE NDVI ANOMALY IN A SPECIFIC AREA

1. Choose one area of interest and add a point on the map.
2. Import the point and use it to filter the image collection of S2.
3. Select the S2 image collection for the month of July for the years 2015, 2016, 2017, 2018 and 2019.
4. Calculate the median NDVI of S2 image collection STA.
5. Select the S2 image collection for the month of July 2020 and calculate the median of NDVI.
6. Use map algebra to calculate absolute, relative and standardized anomaly.
7. Compare the results.

https://code.earthengine.google.com/65cc6d4ba0440027dd12e30c3e795bbf

25. Introduction to Synthetic Aperture Radar images

While optical satellites record the Sun's energy reflected from the target surface, Synthetic Aperture Radar (SAR) satellites emit their own energy that reaches the land, interact with the target surface and is then scattered back to the satellite. The energy emitted is on the microwave range of the electromagnetic spectrum. Microwaves penetrate through clouds and smoke and because of that are defined as all-wheatear (regardless on weather conditions) and all-time sensors (as they collect data during day and night).

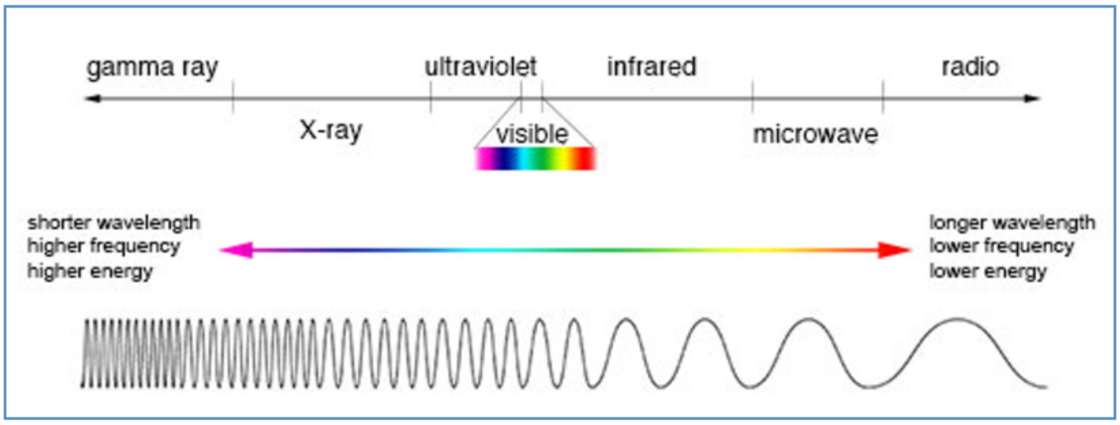


Figure 6. NASA's Imagine the Universe

Source: National Aeronautics and Space Administration (NASA). 2022. In: Imagine the universe! The electro-magnetic spectrum. https://imagine.gsfc.nasa.gov/

The strength of the backscattered signal is measured to discriminate between different targets and the time delay between the transmitted and reflected signals determines the distance (or range) to the target. The beam is oriented to send pulses oblique to the land, which requires some spatial adjustments when processing SAR images.

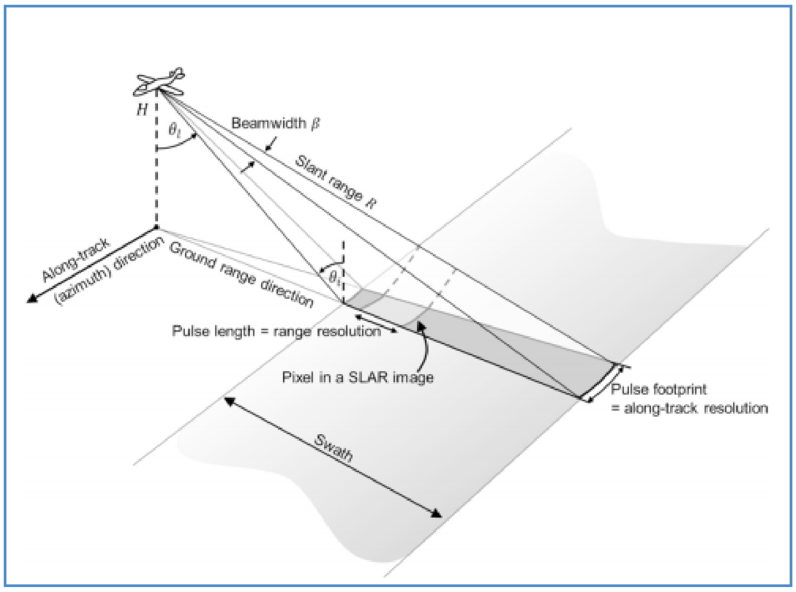


Figure 7. Observation geometry of SAR imager

Source: SAR Handbook: Comprehensive methodologies for forest monitoring and biomass estimation. 2019. In: Servir Global. https://servirglobal.net/Global/Articles/Article/2674/sar-handbook-comprehensive-methodologies-for-forest-monitoring-and-biomass-estimation

Differently from a True color optical imagery, the brightness of the pixel is not indicative of the color of the target object on the land. Instead, its intensity depends on a number of other factors:

- the amount of energy transmitted from the satellite;
- the properties of the target;
- the shape of the target;
- the angle from which the target is viewed.

The satellite's receiver records the backscatter coefficient (o), given by the following formula:
o (dB) = 10. Log10 (energy ratio)

whereby the energy ration is the ration between the received energy by the sensor and the energy reflected in an isotropic way.

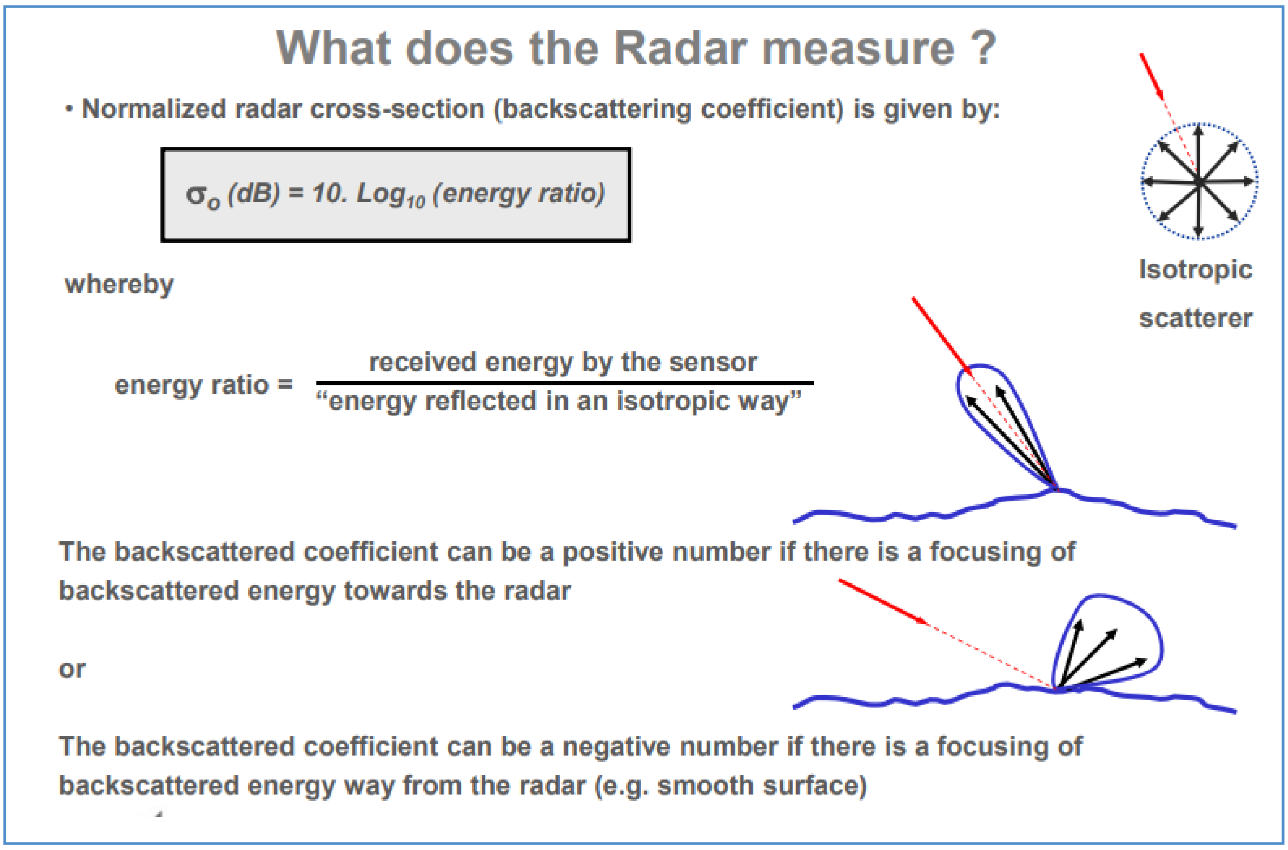


Figure 8. The radar measurement

Source: Moreira A. 2013. Conference: Advanced Training Course in Land Remote Sensing. Synthetic Aperture Radar (SAR). Principles and applications.

26. Characteristics of a sar system

There are three main parameters that characterize SAR systems: wavelength, look angle and polarization.

26.1. Wavelength

All the SAR systems emit energy in the microwave spectrum, however specific part of this range are used in different systems.

The wavelength is directly linked to the ability of the emitted energy to penetrate through the target objects, such that longer wavelength signals (L and P band) penetrate deeper into vegetation canopy and soils, hence the applications supported depend on the SAR wavelength used.

BAND	FREQUENCY	WAVELENGTH	TYPICAL APPLICATION
Ka	27 – 40 GHz	1.1 – 0.8 cm	Rarely used for SAR (airport surveillance)
K	18 – 27 GHz	1.7 – 1.1 cm	Rarely used (H_2O absorption)
Ku	12 – 18 GHz	2.4 – 1.7 cm	Rarely used for SAR (satellite altimetry)
X	8 – 12 GHz	3.8 – 2.4 cm	High-resolution SAR (urban monitoring; ice and snow, little penetration into vegetation cover; fast coherence decay in vegetated areas)
C	4 – 8 GHz	7.5 – 3.8 cm	SAR workhorse (global mapping; change detection; monitor-ing of areas with low to moderate vegetation; improved pen-etration; higher coherence); Ice, ocean, maritime navigation
S	2 – 4 GHz	15 – 7.5 cm	Little but increasing use for SAR-based Earth observation; agriculture monitoring (NISAR will carry an S-band channel; expands C-band applications to higher vegetation density)
L	1 – 2 GHz	30 – 15 cm	Medium resolution SAR (Geophysical monitoring; biomass and vegetation mapping; high penetration; InSAR)
P	0.3 – 1 GHz	100 – 30 cm	Biomass. First P-band spaceborne SAR will be launched ~2020; vegetation mapping and assessment. Experimental SAR.

Figure 9. Designation of radar bands

Source: SAR Handbook: Comprehensive methodologies for forest monitoring and biomass estimation. 2019. In: Servir Global. https://servirglobal.net/Global/Articles/Article/2674/sar-handbook-comprehensive-methodologies-for-forest-monitoring-and-biomass-estimation

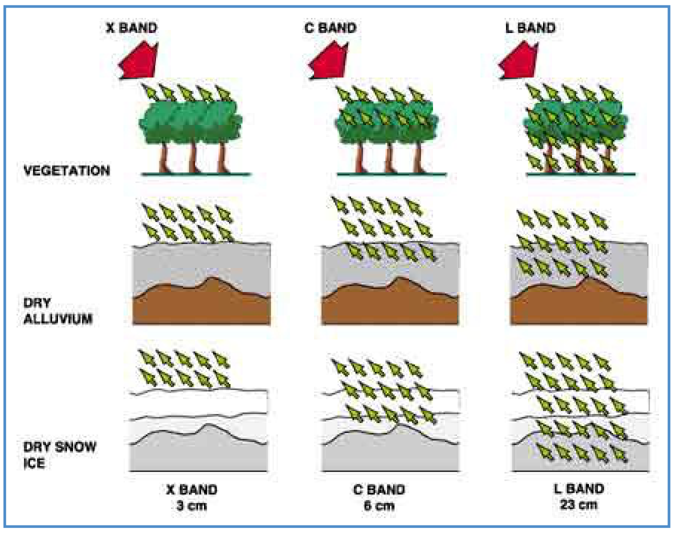


Figure 10. Reflectance mechanism of vegetation in different bands (SOURCE ESA Radar Course 2)

Source: European Space Agency (ESA). 2022. In: Training courses: Radar Course 3. https://earth.esa.int/eogateway/missions/ers/radar-courses/radar-course-3

26.2. Look angle

SAR sensors are always oriented at a certain angle respect to the satellite's orbit. The incident angle is the angle between the radar beam and the ground surface (A).

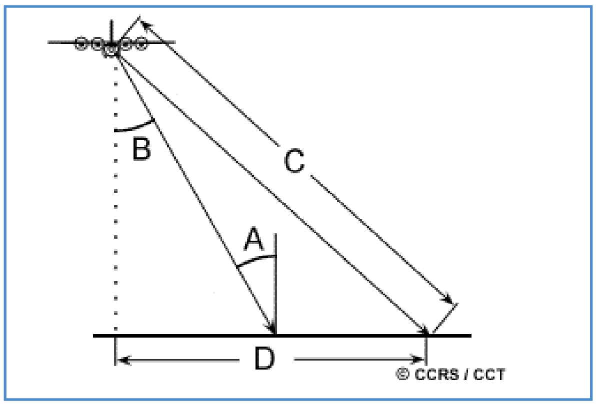


Figure 11. Incident angle

Source: Canada Centre for Remote Sensing. 2022. In: Fundamentals of Remote Sensing. https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/earthsciences/pdf/resource/tutor/fundam/pdf/fundamentals_e.pdf

The incident angle, coupled with the topography of the land may create artifacts (layovers, shadowing, foreshortening) that alter the brightness of the images.

Depending on the satellite's orbit, the image is acquired when satellite is ascending (moving closer to the North) or descending (moving closer to the South).

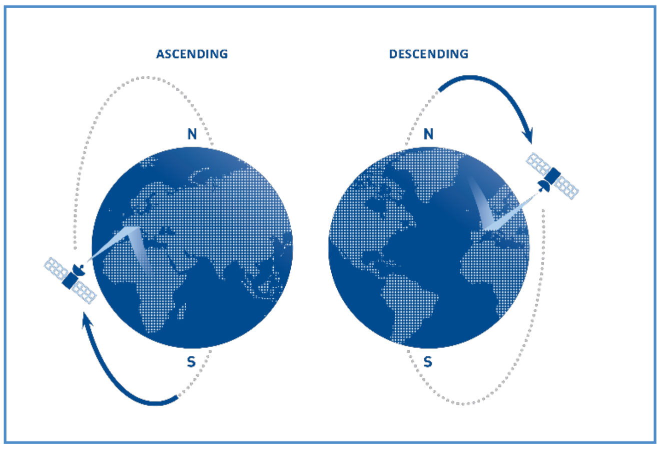


Figure 12. Satellite orbits

Source: Tre Altamira. 2022. In: InSar at a glance. https://site.tre-altamira.com/insar

26.3. Polarization

Imaging radars can transmit horizontal (H) or vertical (V) microwave radiation. At the same way, the receiving antenna can record horizontal and vertical polarizations. Based on this, four possible combinations of transmit and receive polarization exists:

- HH - for horizontal transmit and horizontal receive
- VV - for vertical transmit and vertical receive
- HV - for horizontal transmit and vertical receive, and
- VH - for vertical transmit and horizontal receive.

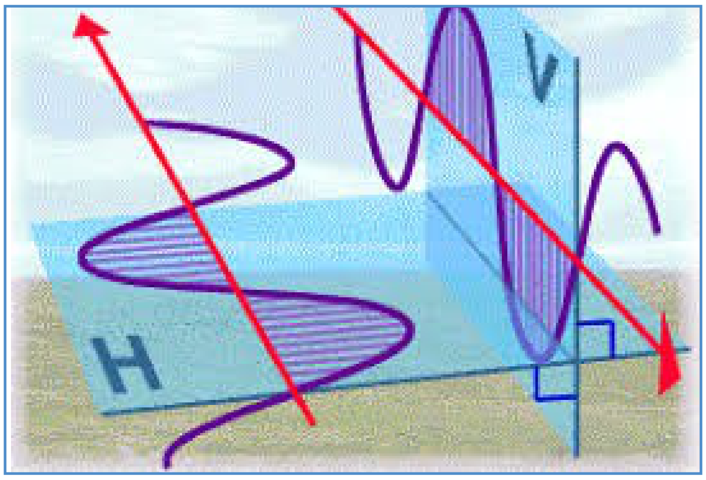


Figure 13. Polarization in SAR images

Source: South African National Space Agency (SANSA). 2022. In: Application of GIS to Fast Track Planning and Monitoring of Development Agenda. https://www.nepad.org/

27. Surface parameters

The response and intensity of a backscatter signal is dependent of the characteristics of the target land: surface roughness, material and shape affects this response.

27.1. Surface roughness

In general, smooth surfaces tend to scatter the emitted energy to a specular angle away from the satellite and as a result, the area tends to be dark (low values).

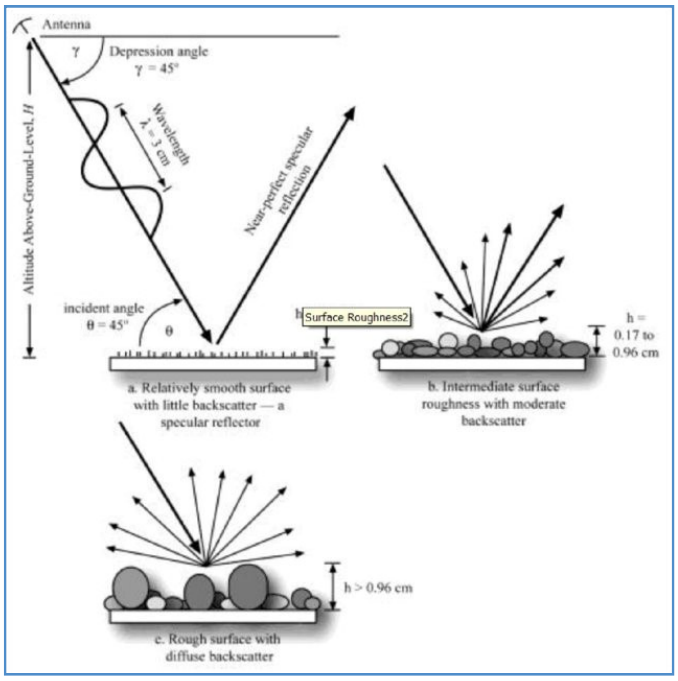


Figure 14. SAR imagery and field validation methods

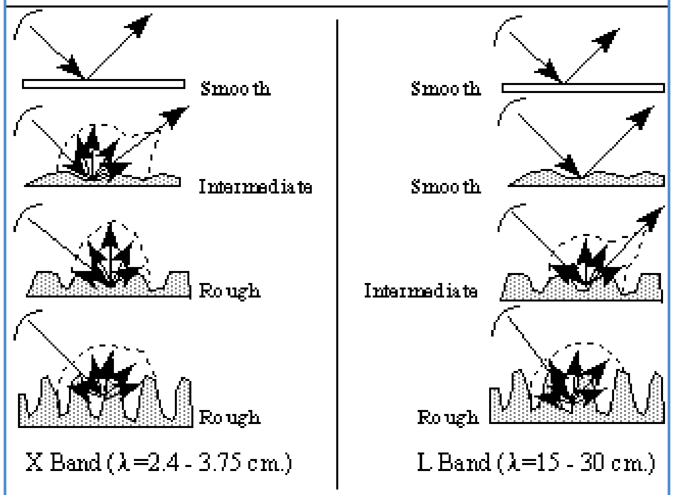
Source: de Jong T. 2013. Recent changes in glacier facies zonation on Devon Ice Cap, Nunavut, detected from SAR imagery and field validation methods. Thesis for: Msc

However, how the surface is perceived as smooth or rough, it depends also from the wavelength of the satellite. A surface appearing "rough" in a short wavelength will appear "smoother" in a longer wavelength signal. As shorter wavelengths are reflected by smaller features, they provide more detailed information at smaller scales.

SLAR Surface Roughness at a Local Incidence Angle of 45°

Root-Mean-Square Surface Height Variation (cm.)	K_a Band (λ = 0.86 cm.)	X Band (λ = 3.2 cm.)	L Band (λ = 23.5 cm.)
0.05	Smooth	Smooth	Smooth
0.10	Intermediate	Smooth	Smooth
0.5	Rough	Intermediate	Smooth
1.5	Rough	Rough	Intermediate
10.0	Rough	Rough	Rough

(From Lillesand and Kiefer, 1994.)



X-Band and L-Band radar reflection from surfaces of varying roughness. (After Lillesand and Kiefer, 1994.)

Figure 15. X-Band and L-Band radar reflections

Source: Messina P. 2022. Radar Mapping Techniques and Applications. In: Transmission and Return Characteristics of Radar Signals http://www.geo.hunter.cuny.edu/terrain/radariv.html

27.2. Target material

Target material properties play a major role in that target's SAR imagery. These properties include dielectric constant and permeability of the material. The dielectric constant is generally affected by moisture, thus increasing moisture is associated with an increased radar reflectivity.

27.3. Shape

The shape of certain surface features will cause a specular reflection back toward the sensor, by bouncing off multiple surfaces.

28. Scattering mechanisms

SAR characteristics and surface parameters interact together and determine the type and strength of backscattered energy.

We discussed already the conditions that affect the surface scattering (roughness and wavelength) and the double bounce. Volumetric scattering often occurs in vegetation where the signal is bounced back many times before reaching back the sensor's satellite.

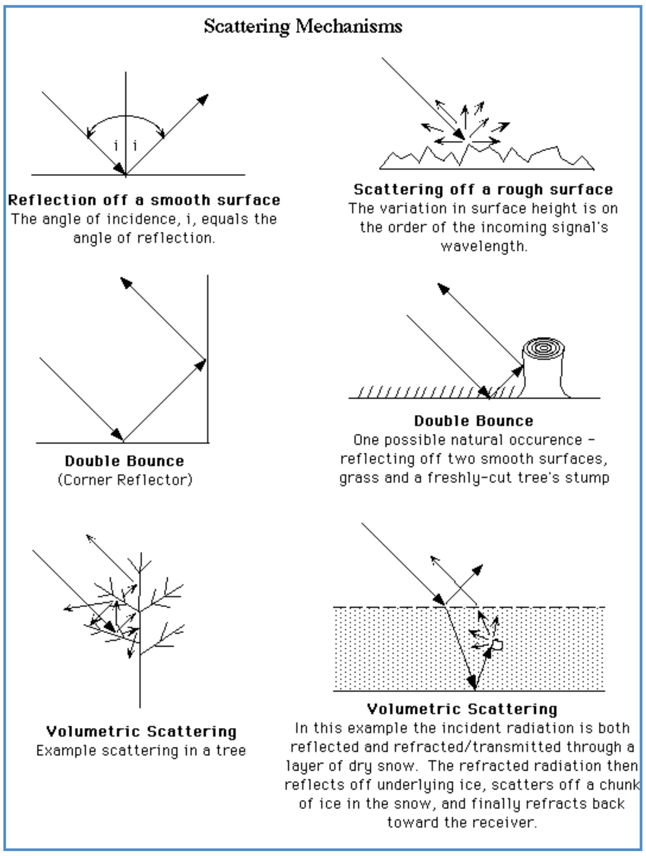


Figure 16. Scattering mechanisms

Source: European Space Agency (ESA). 2022. In: Sentinel Online. https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-1-sar

As a rule of thumb and to support the interpretation of a SAR image, we can generalize the behaviour of the main types of surfaces.

Table 2. Behaviour of main types of surfaces

Target	Backscatter coefficient	Polarization	Composite RGB (VV,VH,VV-VH)
Urban areas, very rough areas, terrain slopes towards radars	Above -5 dB	VV is stronger than VH due to double bounce	Towards pink (high VV)
Dense vegetation	Between -10 dB to 0 dB	VH is strong due to volume scatter	Towards green (high VH)
Crops	Between -20 to -10 dB	VV and VH are similar	Towards purple
Water, bare soil, sand	Below -20 dB	VH is lower than VV	Dark blue, black (difference between VV and VH)

29. Sentinel-1 main characteristics

S1 operates in four exclusive acquisition modes:

- Stripmap (SM).
- Interferometric Wide swath (IW).
- Extra-Wide swath (EW).
- Wave (WV).

The default and main acquisition mode over the land is the Interferometric Wide Swath. The other acquisition modes may be activated for specific circumstances and applications.

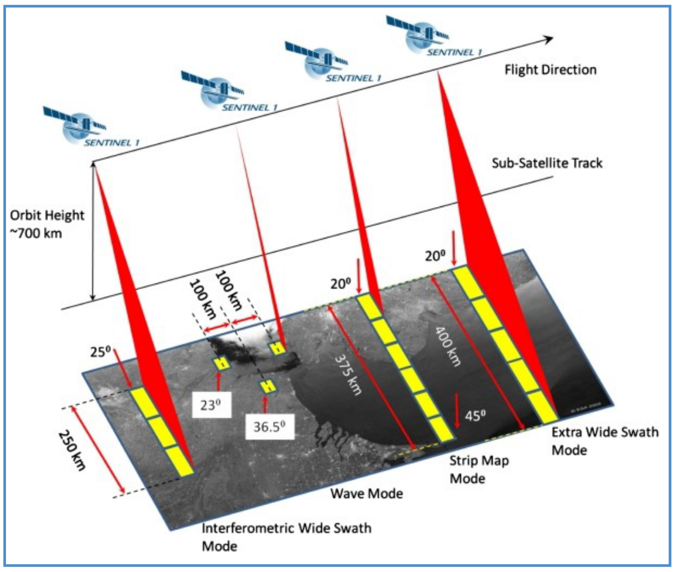


Figure17. Acquisition modes

Source: European Space Agency (ESA). 2022. In: Sentinel Online. https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-1-sar

Level-1 products can be one of two product types - either Single Look Complex (SLC) or Ground Range Detected (GRD). Level-2 Ocean (OCN) products can have different components available depending on the acquisition mode.

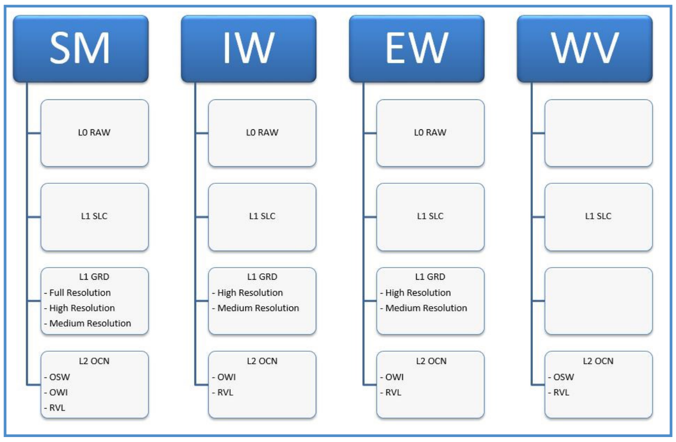


Figure 18. Product types processing levels

Source: European Space Agency (ESA). 2022. In: Sentinel Online. https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-1-sar

GRD products consist of focused SAR data that has been detected, multi-looked and projected to ground range using an Earth ellipsoid model. The ellipsoid projection of the GRD products is corrected using the terrain height specified in the product general annotation. The terrain height used varies in azimuth but is constant in range.

EXERCISE: DISPLAY A SINGLE S1 IMAGE

1. Add a point to the map.
2. Select the S1 image collection.
3. Print the metadata of the first result image.
4. Filter the image collection to your area of interest, the last year and select the instrument mode “IW”, the polarization VV and VH and the descending orbit direction.
5. Add to the map view two different images, for VV and VH.

https://code.earthengine.google.com/f1a8b0afa0bdf6cfdcdbd2df8827682d

30. Display a color composite SAR image

Despite the fact that SAR images can not represent true colors, as the energy recorded is not the visible portion of the spectrum, few color composite combinations may be applied to a single S1 image. The most common combinations use the VV band, the VH and the ratio (or difference) between the VV and VH. In the next execersize, we will use the same filtered image collection and we will display it as false color composite.

EXERCISE: DISPLAY A SINGLE SENTINEL-1 IMAGE IN COLOR COMPOSITE

1. On the previous image collection, select the first image and create a new image as difference between VV and VH.
2. Add the band to the image (addBands).
3. Display the resulting image as color composite.

https://code.earthengine.google.com/77845d5ff0b1e1fcdcfe79316dc4e87b

31. Pre-processing Sentinel-1 image collection

Different pre-process steps are applied to SAR images before using them in applications.

These steps depend on the type of application and on how those images are filtered. The following image presents the pre-processing steps suggested by SERVIR to be applied to SAR data before using.

You can apply and practise all the steps on a single image using the SNAP software.

ONE PAGER SERIES

SAR Data Pre-Processing Steps

A cheat sheet outling the workflow for pre-processing L1 Synthetic Aperture Radar data. For more information, check out the SAR Handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation and associated training materials at **SERVIRglobal.net**

Apply Orbit File
Defines the relationship between ground and image coordinates, improves accuracy of later orbit-based calibration steps.

Radiometric Calibration
Converts the image pixel values from digital number (DN) to a standard geophysical measurement unit of radar backscatter.

De-Bursting
SAR scenes can be made up of multiple swaths or sections. This step combines all swaths into a single image.

Multilooking
Uses spatial averaging to reduce image speckle noise and converts to ground range, producing an image with a standard pixel size. Reduces image resolution *(optional)*.

Speckle Filtering
Removes noise, or speckle, in an image. Many types of speckle filters can be applied, and different applications have specific filters that may work best. Unlike multilooking, this step does not reduce spatial resolution *(optional)*.

Terrain Correction: Radiometric Terrain Flattening (RTF) & Geocoding
RTF: Uses a DEM to remove geometry-dependent radiometric distortions; normalizes measured backscatter with respect to terrain slope.
Geocoding: Uses a DEM to remove geometric distortions such as foreshortening, layover, and shadow; connects the image to a geographic coordinate system.

Convert to dB
Linearly-scaled data is converted into decibels (dB) (optional).

SOURCE: Meyer, Franz. "Spaceborne Synthetic Aperture Radar – Principles, Data Access, and Basic Processing Techniques." SAR Handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation. Eds. Flores, A., Herndon, K., Thapa, R., Cherrington, E. NASA. DOI: 10.25966/ez4f-mg98

Figure 19. SAR data processing

Source: SAR Handbook: Comprehensive methodologies for forest monitoring and biomass estimation. 2019. In: Servir Global. https://servirglobal.net/Global/Articles/Article/2674/sar-handbook-comprehensive-methodologies-for-forest-monitoring-and-biomass-estimation

We will use functions written in an existing shared workspace, to apply some of those pre-process steps on a single image. To upload an external script, we will use the 'require' function, to retrieves the script found at a given path as a module. The module is used to access exposed members of the required script.

We will use a function that applies the following steps:

1. Remove the pixels in the borders.
2. Apply a calibration.
3. Add a VV, VH ratio band.
4. Add a VV, VH difference band.

Finally, we will use a second preprocessing that will add an extra-step with speckle removal.

https://code.earthengine.google.com/069dd5702e68d56af89d3f9049a08ab3
https://code.earthengine.google.com/720ee9103ae699a4fcbe7f02be500f79

32. Sentinel-1 time series

Another approach to display color composite Sentinel, is to select images over multiple periods and then, assign each period to the red, green and blue channels. The dominance of a band corresponds to the strength of the backscatter in the assigned period.

EXERCISE: DISPLAY A THREE-PERIODS TIME SERIES AS COLOR COMPOSITE

1. Filter three different S1 image collections for April, June, August (leave the other filtering criteria unchanged respect to the previous script) and select the first image.
2. Pre-process the three selected images.
3. Create a single image (both with VV and VH).
4. Display two color composite (VV and VH) and comment results.

https://code.earthengine.google.com/fe4fba5151bbb9e4e0c30b3b2074c60a

33. Land cover classification

In the following scripts, we will explore techniques for a classification of pixels into thematic classes. We will use a simplified legend with only five classes (urban, agriculture, water, sand and wetland). The two main techniques for image classifications are unsupervised and supervised classifications. In the first case, the pixels of the images are grouped into a number of classes (the number classes may not correspond initially to the land cover classes) based on their spectral similarity. Similar pixels belong to the same classes, although no thematic information (i.e. the name of the land cover class is provided). This technique does not require the use of training data to build a classification model. Is the use that in a post-process step, define the thematic classes to each group. The example on a limited area of interest uses progressively an increased number of time-series datasets: S2 only, S1 only, and then a combined image of S1, S2 and topography (elevation and slope).

The script is divided into two parts:
https://code.earthengine.google.com/faf72bdc83d6b88961a36944b19e0c8
https://code.earthengine.google.com/69874f4d0cb2e24b1d3d50b60987e609

The supervised technique requires that the user provides a set of training samples to build a model that is than applied to the rest of the image. Multiple steps are required:

1. Building the image collections (selection, pre-process, stack).
2. Upload a Shapefile of training samples.
3. Calculate statistics of each training sample over the image collection.
4. Define model parameters.
5. Apply the models over the training sample.
6. Use the model to classify the image.

Example of this technique is provided here:
https://code.earthengine.google.com/214a3945780da7c9c853d9c935a59cac

Introductory course to

Google Earth Engine

ISBN 978-92-5-135907-5

9 789251 359075

CB9049EN/1/06.22