

Digital Earth: Big Earth Data Concepts

Exercise/Assignment 6: Temporal analysis in Google Earth Engine

Martin Sudmanns & Dirk Tiede

Summary

In this assignment you should modify an existing Google Earth Engine (GEE) script.

Widgets in GEE

- Within the GEE exercise, the widget example showed a split screen for comparing two different datasets
- Check the exercise for more information and execute again the code in your GEE environment (the GEE code is available in the exercise and also copied below).
- **Your task is to change the code to fulfil the following criteria:**
 - Replace the area of interest (here: volcano event) to any other place of the world, where a significant change can be observed (e.g. glacier retreat, deforestation, flooding, urban expansion....)
 - Change the date to match your example/event
 - Change – if needed for your example – the band combination or use e.g. the NDVI.
 - Change – if needed for your example – the data (e.g. if you want to use Landsat data)
 - Make a screenshot of your resulting change visualisation
 - Upload the code and the screenshot in one document

Add-on (optional):

- You can try to use a different widget instead of the split panel (e.g. the *dateslider*) for change visualisation, see https://developers.google.com/earth-engine/guides/ui_widgets

In the following example, the slider widget is used to switch between two linked maps, which show different images from different dates for the same location (change comparison):

```
// First define an area of interest
var lat = -17.85;
var lng = 28.66;
var point = ee.Geometry.Point(lat, lng); //
var aoi = point.buffer(1000); // Create an area (1km buffer around point)
Map.setCenter(lat, lng, 10); // Center the map on this location, zoom level 10
// Download the Sentinel-2 imagery collection
var s2 = ee.ImageCollection('COPERNICUS/S2');
// Filter the collection by location, date boundaries and cloud cover percentage
var s2_filtered = s2.filterBounds(aoi)
  .filterDate('2021-07-01','2021-08-31')
  .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 30);

// This will sort lowest cloud cover image first.
var sorted = s2_filtered.sort('CLOUDY_PIXEL_PERCENTAGE', true);
// Pick the first image from the sorted and filtered collection
var image = ee.Image(sorted.first());
// Define the visualisation parameters to display the image (change bands for different visualisations)
var vizParams = {
  bands: ['B12', 'B11', 'B8'],
  min: 0,
  max: 3000,
  gamma: [1, 1, 1]
};
// Add the image to the default Map.
Map.addLayer(image, vizParams, "Sentinel-2 before");

// Make another map and add a after event image to it.
var linkedMap = ui.Map();
// Filter the collection by location, date boundaries and cloud cover percentage
var s2_filtered = s2.filterBounds(aoi)
  .filterDate('2021-10-01','2021-11-30')
  .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 30);

// This will sort lowest cloud cover image first.
var sorted = s2_filtered.sort('CLOUDY_PIXEL_PERCENTAGE', true);
// Pick the first image from the sorted and filtered collection
var image = ee.Image(sorted.first());

// Add the after event image to the linked map - use same visualisation as defined for the before image
linkedMap.addLayer(image, vizParams, "Sentinel-2 after");

// Link the default Map to the other map.
var linker = ui.Map.Linker([ui.root.widgets().get(0), linkedMap]);

// Create a SplitPanel which holds the linked maps side-by-side.
var splitPanel = ui.SplitPanel({
  firstPanel: linker.get(0),
  secondPanel: linker.get(1),
  orientation: 'horizontal',
  wipe: true,
  style: {stretch: 'both'}
});
// Set the SplitPanel as the only thing in root.
ui.root.widgets().reset([splitPanel]);
```