

Machine learning with GEE

The aim of this exercise is to classify the pixels of a cloud-free Landsat 7 TOA composite generated from raw Landsat 7 images (at-sensor radiance) collected in a given area of interest (AOI) for a temporal period of interest.

I applied a K-mean algorithm.

This is an iterative algorithm that calculates the mean of each cluster and then assign the point to a cluster with the minimum distance.

I will focus on the following area in the period between 2002-04-01 and 2002-10-31.

```
▼ var geometry: Polygon, 4 vertices ⚙️ 🔍  
  type: Polygon  
  ▼ coordinates: List (1 element)  
    ▼ 0: List (5 elements)  
      ▶ 0: [20.8477028233494,69.71793138998997]  
      ▶ 1: [26.4012916905369,69.71793138998997]  
      ▶ 2: [26.4012916905369,71.00562410221487]  
      ▶ 3: [20.8477028233494,71.00562410221487]  
      ▶ 4: [20.8477028233494,69.71793138998997]  
  
  geodesic: false
```

After looking at the different results I got I decided to use 3 clusters.

I'll attach the code I used and the visualization before and after the K-means.

```

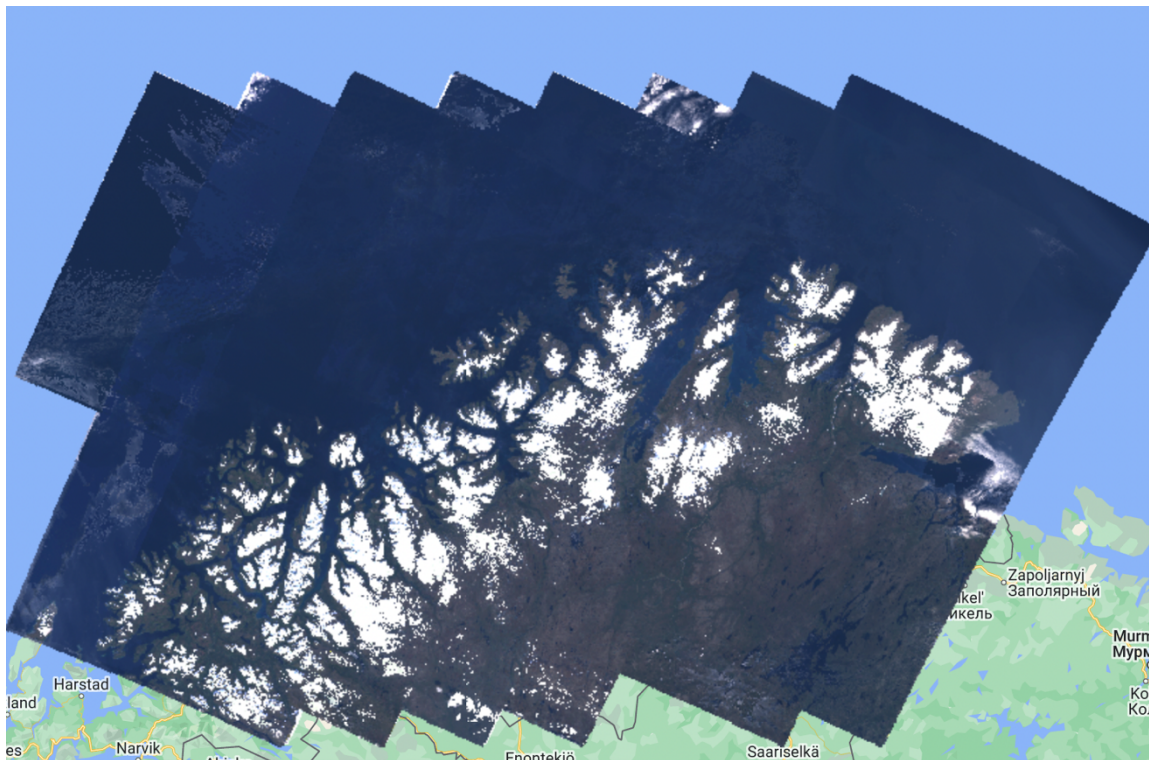
Imports (1 entry)
var geometry: Polygon, 4 vertices

1
2 // first step is to load Landsat 7 raw image collection
3 var l7 = ee.ImageCollection("LANDSAT/LE07/C02/T1");
4
5 // Filter the collection using the date, location and cloud coverage
6 var l7_filtered = l7.filterDate('2002-04-01', '2002-10-31')
7   .filterBounds(geometry)
8   .filter(ee.Filter.lt('CLOUD_COVER', 50));
9
10 // Make a cloud-free Landsat 7 TOA composite
11 // (from raw imagery: DN values, representing scaled, calibrated at-sensor radiance)
12 var l7_composite = ee.Algorithms.Landsat.simpleComposite({
13   collection: l7_filtered,
14   asFloat: true
15 });
16 print('l7_composite', l7_composite);
17
18 // Display the input composite
19 Map.addLayer(l7_composite, {'bands': ['B3', 'B2', 'B1'], 'min': 0, 'max': 0.3}, 'Landsat-7 TOA composite');
20 Map.centerObject(geometry);
21
22
23 // Let's start the training
24 //here I generate a sample of the input landsat 7 compsite
25 var training = l7_composite.sample({
26   region: geometry,
27   scale: 30,
28   numPixels: 5000
29 });
30
31 // I start the actual training
32 var number_of_clusters = 3;
33 var clusterer = ee.Clusterer.wekaKMeans(number_of_clusters).train(training);
34
35 // Use the clusterin on all the input
36 var result = l7_composite.cluster(clusterer);
37 print('result', result);
38
39 // Display the clusters with random colors
40 Map.addLayer(result.randomVisualizer(), {}, 'clusters');
41

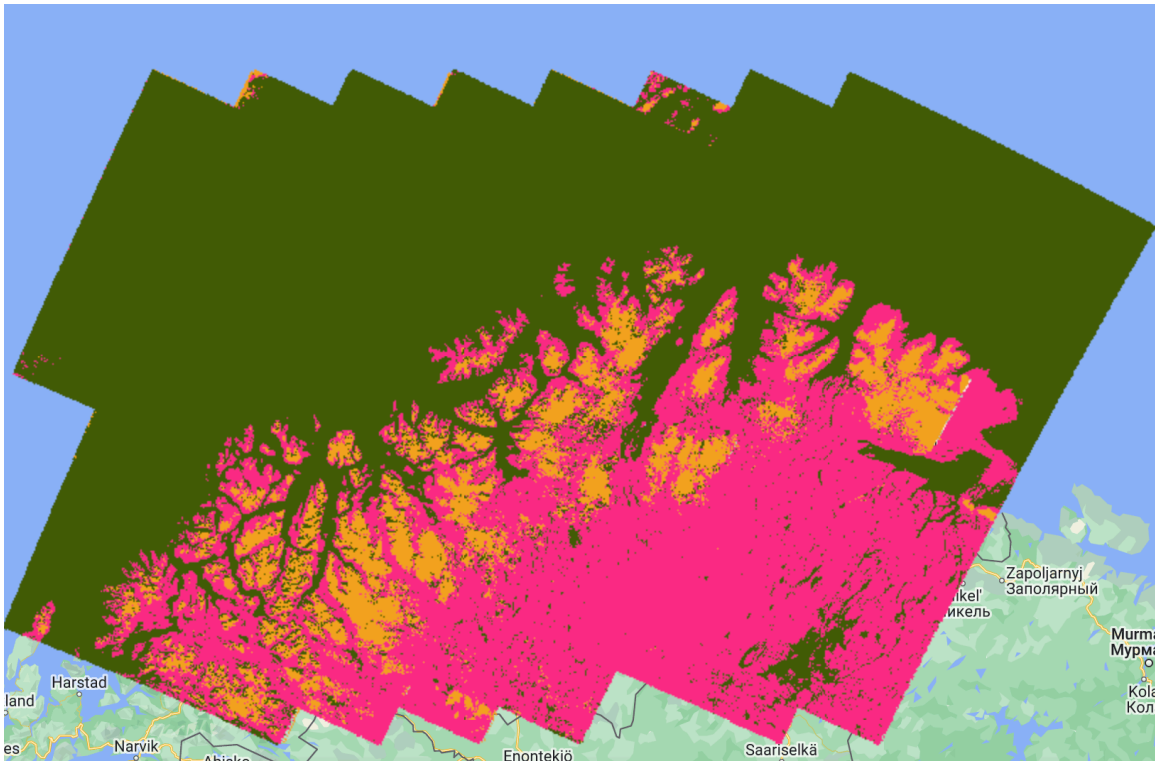
```

Here are the results:

This Is Landsat-7 TOA



And this after the K-means with 3 clusters:



Clearly the water is in green, the snow is in yellow and the earth's surface is pink
The code is [here](#)