Department of Geography

CCST9083 Earth as Seen by Satellite

Laboratory 2: Quantifying the health and density of vegetation - NDVI

In this lab exercise, you will use the web-based code editor to explore landscape changes over time using the Normalized Difference Vegetation Index (NDVI). To access the Code Editor, simply browse for https://code.earthengine.google.com/. Satellite images from Landsat 8 *(USGS Landsat 8 Collection 2 Tier 1 TOA Reflectance)* will be used for this study.

The NDVI is a vegetation index that uses the reflectance properties of healthy vegetation to distinguish vegetated areas from non-vegetated areas (e.g., urbanized areas), as well as estimate how healthy the present vegetation is. This is done by calculating the normalized difference of the Near Infrared (NIR) and the Red (R) band, as healthy vegetation highly reflects Near Infrared light, but barely reflects Red light:

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

The result is a range of values from -1 to 1, with high positive values indicating an abundance of healthy vegetation, values around zero low to none vegetation and high negative values most likely water.

1. Area of interest

Our study focuses on one of China's major cities – Shenzhen, and access our boundary-defining geometry using the GAUL-Level2 layer

```
var roi = ee.FeatureCollection("FAO/GAUL/2015/level2")
            .filter(ee.Filter.eq('ADM0_NAME', 'China'))
            .filter(ee.Filter.eq('ADM2_NAME', 'Shenzhen'))
Map.addLayer(roi, {color: 'green', width: 1}, 'ROI');
Map.centerObject(roi,10)
```

2. Filter the Landsat 8 Data by Area & Time

Load USGS Landsat 8 Collection through the code below:

```
//Period: 2014-2017
var l8_20142017=ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA')
                    .filterDate('2014-01-01','2017-12-31')
                    .filter(ee.Filter.lt('CLOUD_COVER', 15))
```

```
                    .filterBounds(roi);
var l8_20142017_median=l8_20142017.median()
                                    .clip(roi);
print(l8_20142017)

//Period: 2018-2021
var l8_20182021 = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA')
                    .filterDate('2018-01-01','2021-12-31')
                    .filter(ee.Filter.lt('CLOUD_COVER',  15))
                    .filterBounds(roi);
var l8_20182021_median = l8_20182021.median()
                                    .clip(roi);
print(l8_20182021)
```

Click **Run** button in the top menu. The right window will show the results for the period of 2014 – 2017 and 2018 – 2021 in selected study area, respectively.

3. Calculate NDVI and display results

Method 1: using the built in function ".normalizedDifference(NIR, RED)"

```
var ndvi_20142017=l8_20142017_median.normalizedDifference(['B5','B4']);
var ndviParams = {min: -1, max: 1, palette: ['red', 'yellow', 'green']};
Map.addLayer(ndvi_20142017, ndviParams, "2014-2017 NDVI");

var ndvi_20182021 = l8_20182021_median.normalizedDifference(['B5','B4']);
var ndviParams = {min: -1, max: 1, palette: ['red', 'yellow', 'green']};
Map.addLayer(ndvi_20182021, ndviParams, "2018-2021 NDVI");
```

Method 2: using the simple band operations

```
var nir = l8_20142017_median.select('B5');
var red = l8_20142017_median.select('B4');
var ndvi_20142017 = nir.subtract(red).divide(nir.add(red)).rename('NDVI');
var ndviParams = {min: -1, max: 1, palette: ['red', 'yellow', 'green']};
Map.addLayer(ndvi_20142017, ndviParams, "2014-2017 NDVI");

var nir = l8_20182021_median.select('B5');
var red = l8_20182021_median.select('B4');
var ndvi_20182021 = nir.subtract(red).divide(nir.add(red)).rename('NDVI');
var ndviParams = {min: -1, max: 1, palette: ['red', 'yellow', 'green']};
Map.addLayer(ndvi_20182021, ndviParams, "2018-2021 NDVI");
```

Click the **Run** button in the top menu. The map window will display the NDVI results. Layer manager enables you to check different layers.

4. Change detection

A simple subtraction is used to calculate the NDVI difference between 2014 – 2017 and 2018 – 2021:

```
var change_ndvi=ndvi_20182021.subtract(ndvi_20142017)
var ndvidiff = {min: -1, max: 1, palette: ['blue', 'white', 'green']};
Map.addLayer(change_ndvi,ndvidiff,"NDVI_change");
```

5. Pixel value filtering

```
var low_ndvi= change_ndvi.lt(0);
var lowNDVI = change_ndvi.updateMask(low_ndvi);
Map.addLayer(lowNDVI,ndvidiff,"Low NDVI");

var high_ndvi=change_ndvi.gt(0);
var highNDVI = change_ndvi.updateMask(high_ndvi);
Map.addLayer(highNDVI,ndvidiff,"High NDVI");
```

6. Statistics of NDVI

Calculate the basic statistics within the study area:

```
var MeansOfFeatures_20142017 = ndvi_20142017.reduceRegions({
   collection: roi,
   reducer: ee.Reducer.mean(),
   scale: 250,
});

print(ee.Feature(MeansOfFeatures_20142017.first()))

var MedianOfFeatures_20142017 = ndvi_20142017.reduceRegions({
   collection: roi,
   reducer: ee.Reducer.median(),
   scale: 250,
});

print(ee.Feature(MedianOfFeatures_20142017.first()))

var MinOfFeatures_20142017 = ndvi_20142017.reduceRegions({
   collection: roi,
   reducer: ee.Reducer.min(),
   scale: 250,
});
```

```
print(ee.Feature(MinOfFeatures_20142017.first()))

var MaxOfFeatures_20142017 = ndvi_20142017.reduceRegions({
    collection: roi,
    reducer: ee.Reducer.max(),
    scale: 250,
});

print(ee.Feature(MaxOfFeatures_20142017.first()))

var CountOfFeatures_20142017 = ndvi_20142017.reduceRegions({
    collection: roi,
    reducer: ee.Reducer.count(),
    scale: 250,
});

print(ee.Feature(CountOfFeatures_20142017.first()))
```

```
var MeansOfFeatures_20182021 = ndvi_20182021.reduceRegions({
    collection: roi,
    reducer: ee.Reducer.mean(),
    scale: 250,
});

print(ee.Feature(MeansOfFeatures_20182021.first()))

var MedianOfFeatures_20182021 = ndvi_20182021.reduceRegions({
    collection: roi,
    reducer: ee.Reducer.median(),
    scale: 250,
});

print(ee.Feature(MedianOfFeatures_20182021.first()))

var MinOfFeatures_20182021 = ndvi_20182021.reduceRegions({
    collection: roi,
    reducer: ee.Reducer.min(),
    scale: 250,
});

print(ee.Feature(MinOfFeatures_20182021.first()))
```

```
var MaxOfFeatures_20182021 = ndvi_20182021.reduceRegions({
    collection: roi,
    reducer: ee.Reducer.max(),
    scale: 250,
});

print(ee.Feature(MaxOfFeatures_20182021.first()))

var CountOfFeatures_20182021 = ndvi_20182021.reduceRegions({
    collection: roi,
    reducer: ee.Reducer.count(),
    scale: 250,
});

print(ee.Feature(CountOfFeatures_20182021.first()))
```

< End >