Google Earth Engine

Hands on Experience

Project 5 Mapping Urban Green Space

Introduction

Urban Green Space

Urban green spaces are areas in an urban environment reserved for ‘green’ or ‘blue’ areas such as residential greenery, parks, water features. Defined by the World Health Organization as “all urban land covered by vegetation of any kind”, these landscapes can range from relatively natural landscapes to highly maintained environments.

Urban green spaces are of interest because it is recognized as a means for city dwellers to relieve stress, improve social cohesion, reducing the city’s air pollution, heat, traffic noise and emissions and improves a city’s aesthetics and ecology. The result of this is reduced morbidity and mortality of urban residents. There is thus an increased attention from policymakers and the public to actively plan for such spaces.

Leveraging on the power of Google Earth Engine, this project utilizes optical (multispectral band) images from the Sentinel 2 satellite and geospatial analytics to map urban green space locations. This is a simple tool to provide insight for land-use planning purposes.

Central park in New York, USA. Picture obtained from: https://www.ecomatcher.com/the-importance-of-incorporating-green-spaces-in-city-planning/

Sentinel-2 MSI

Mapping of green urban areas can range from the use of simple techniques such as analyzing a vegetation-type index, to more complicated methods such as machine learning techniques like maximum likelihood classification (MLC), random forest, convolutional neural networks (CNNs). Ideally, higher resolution (order of meters or less) are used for such mapping purposes to identify and delineate boundaries of the 'green' locations within urban areas. In this project, we will be using Sentinel-2 multispectral imagery which is readily available within GEE and provides a good spatial resolution of 10 m. As a start, a simple vegetation index will be used to map green urban areas.

Sentinel-2 Multi-Spectral Instrument (MSI)

- Comprises of 2 satellites: S-2A and S-2B
- 13 spectral bands
- Spatial Resolution: Visible & NIR (10 m), red-edge & SWIR (20 m), atmospheric bands (60 m)
- Temporal Resolution: 5 day revisit with 2 satellites

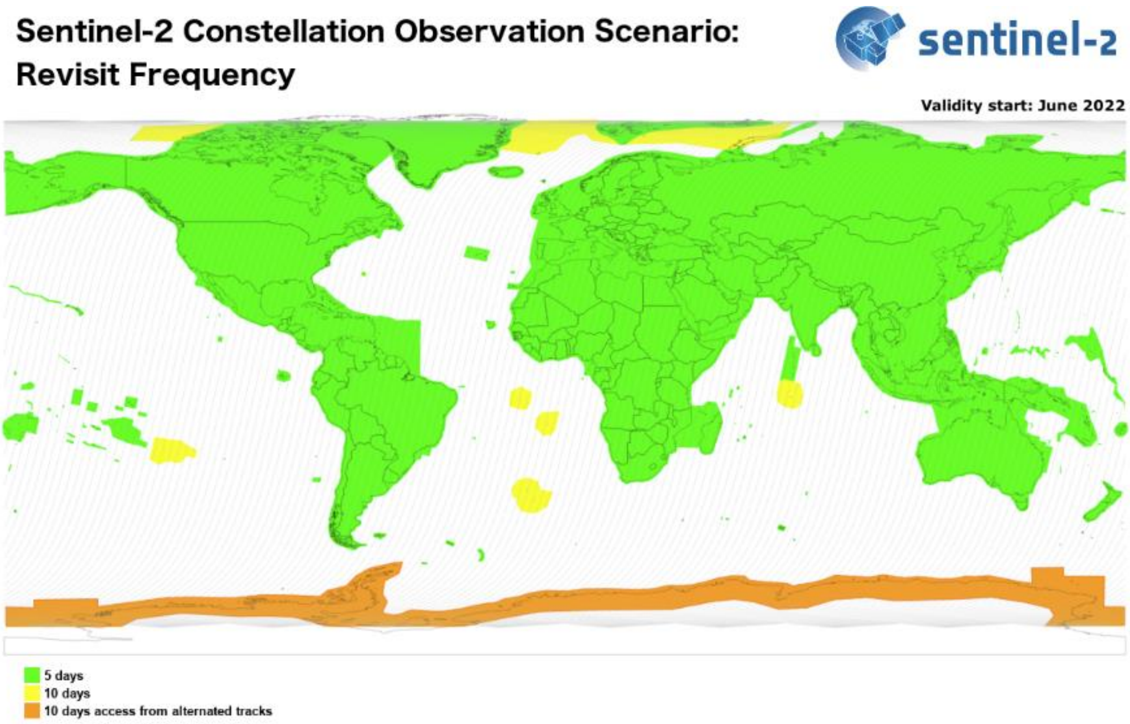

Figure 1: Illustrates the coverage and revisit time foreseen for SENTINEL-2 MSI acquisitions.

Extracted from: https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/revisit-coverage

Guided Lab

Note: Highlighted in red are variables you may try some simple modification

1. Import Datasets

```
// Import S2 dataset
var s2 = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED');

// Import FAO GAUL 2 dataset
// City names may be obtained from HERE.
var gaul2 = ee.FeatureCollection('FAO/GAUL/2015/level2');
```

2. Define cities and years of interest

```
// Capital cities list
// Below are a list of cities in Indonesia
var cities = ['Kota Semarang','Kota Yogyakarta','Kota
Surabaya','Kota Jakarta Utara'];

// Define the year of analysis
var years = ['2019','2020','2021','2022'];

// Define an empty Feature collection called featureGreen
var featureGreen = ee.FeatureCollection([]);
```

3. Functions

```
// Function for cloud masking
function cloudMask(image){
  var scl = image.select('SCL');
  var mask = scl.eq(3).or(scl.gte(7).and(scl.lte(10))).eq(0);
  return image.updateMask(mask);
}
```

Guided Lab

4. Generate and loop through every image per year and per city

```
// Generate image per year and per city

//start of first loop by city
cities.map(function(city){

  // Filter feature collection to get city geometry
  var roi = gaul2.filter(ee.Filter.eq('ADM2_NAME',
city)).geometry();

  //start of second loop by year
  years.map(function(year){

    // Generate sentinel-2 composite free imagery
    var image = s2.filterBounds(roi).filterDate(year + '-01-01',
year + '-12-31')
    .filter(ee.Filter.dayOfYear(150, 240)) // Filter only in dry
season
    .map(cloudMask) // Cloud masking
    .median()
    .divide(10000) // Scale to 0 - 1
    .clip(roi);

    // Show the image of each cities
    Map.addLayer(image, { min: [0.05, 0.035, 0], max: [0.5,
0.35, 0.2], bands: ['B8', 'B11', 'B2']}, 'S2_' + city + year,
false);
```

Guided Lab

5. Calculate EVI & Green Urban Area

Note: This set of codes are still within the 2 loops in Section 4.

```
     // Band map for index
    var bandMap = { NIR: image.select('B8'), RED: image.select('B4'),
BLUE: image.select('B2') };

    // Calculate enchanced vegetation index
    var evi = image.expression('(2.5 * (NIR - RED)) / (NIR + 6 * RED -
7.5 * BLUE + 1)', bandMap);

    // Show EVI
    Map.addLayer(evi, { min: -0.5, max: 0.5, palette: ['red', 'yellow',
'green']}, 'EVI_' + city + year, false);

    // Urban green space using a threshold of 0.35
    var green = evi.gte(0.35).selfMask();
    Map.addLayer(green, { palette: 'green' }, 'Urban green space ' +
city + year, false);

    // Area pixel in Hectare
    var areaPixel = ee.Image.pixelArea().divide(10000);

    // Calculate city area
    var area =
ee.Number(evi.mask().multiply(areaPixel).rename('area').reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: roi,
      scale: 10,
      bestEffort: true
    }).get('area'));

    // Calculate city urban green space area
    var areaGreen =
ee.Number(green.multiply(areaPixel).rename('area').reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: roi,
      scale: 10,
      bestEffort: true
    }).get('area'));

    // Percentage of green area
    var percentGreen = areaGreen.divide(area).multiply(100);
```

Guided Lab

6. Storing parameters in a featureCollection

Note: This set of codes are still within the 2 loops in Section 4.

```
 print(city, ee.String('Area: ').cat(area),
ee.String('Year:').cat(year), ee.String('Green area:
').cat(areaGreen), ee.String('Percentage:
').cat(percentGreen));

    featureGreen =
featureGreen.merge(ee.FeatureCollection(ee.Feature(null , {
City: city, Year: year, Area: area, Green_area: areaGreen,
Percentage: percentGreen})));
  }); // end of TIME LOOP

}); //end of CITY LOOP
```

Guided Lab

7. Create a table and chart of the data

```
// Create table
var tableGreen = ui.Chart.feature.byFeature(featureGreen, 'City',
['Area', 'Year','Green_area', 'Percentage'])
  .setChartType('Table')
  .setOptions({
    pageSize: 20
  });
print(tableGreen);

// Create chart of area per city
var chartGreenArea = ui.Chart.feature.groups(featureGreen,
'City', 'Green_area', 'Year')
  .setChartType('ColumnChart')
  .setOptions({
    title: 'Green Urban Area (hectare)',
    vAxis: {title: 'Green Urban Area (hectare)',titleTextStyle:
{italic: false, bold: true}},
    colors: ['604791', '1d6b99', '39a8a7', '0f8755', '76b349',
'f0af07',
    'e37d05', 'cf513e', '96356f', '724173', '9c4f97', '696969']
  });
print(chartGreenArea);


// Create chart of area percentage per city
var chartGreenPercent = ui.Chart.feature.groups(featureGreen,
'City', 'Percentage', 'Year')
  .setChartType('ColumnChart')
  .setOptions({
    title: 'Green Urban Area (%)',
    vAxis: {title: 'Green Urban Area (%)',titleTextStyle:
{italic: false, bold: true}},
    colors: ['604791', '1d6b99', '39a8a7', '0f8755', '76b349',
'f0af07',
    'e37d05', 'cf513e', '96356f', '724173', '9c4f97', '696969']
  });
print(chartGreenPercent);
```

Guided Lab

8. Explore the results

Click on 'Layers' in the map panel and explore the various maps of urban green space, EVI and Sentinel 2 images.

What do the various oolors means?
How 'accurate' are the results from eyeballing?

Under Console panel, explore the city areas and green areas calculated, the charts produced and try to extract the table out as a csv file.

Which cities show an increase/decrease in green areas?
Which cities show the least/most green areas?

Guided Lab

9. Additional Exploratory Topics

About: Enhanced Vegetation Index, https://en.wikipedia.org/wiki/Enhanced_vegetation_index

1) Using other sensors such as the Landsat series or MODIS and/or other indices such as NDVI to extract the green urban areas.

2) Mapping of public urban spaces versus private urban spaces.
(https://www.mdpi.com/2220-9964/10/4/251)
(https://www.gislounge.com/mapping-public-urban-green-spaces-open-gis-data/)

3) Explore the extent of green urban spaces with land surface temperature.
(https://developers.google.com/earth-engine/datasets/tags/surface-temperature)

4) Explore the extent of green urban spaces with air pollution and/or air quality indicators.
(https://developers.google.com/earth-engine/datasets/tags/pollution)

5) Explore the extent of green urban spaces with human factors such as mortality, life expectancy, obesity rates etc.