EXPLORING SPECTRAL INDICES IN Google Earth Engine

OVERVIEW

- Spectral Indices are combinations of spectral reflectance from two or more wavelengths that indicate the relative abundance of features of interest
- There are different indices for vegetation, built-up features, water and burned areas
- In this section we will have hands-on tutorial exploring some vegetation indices in the Google Earth Engine environment.

Importing Image and Feature Collections

- Import the Landsat 8 Surface Reflectance collection as well as the Study Area

```
Imports (2 entries)
  var imageCollection: (Deprecated) ImageCollection "USGS Landsat 8 Surface Reflectance Tier 1…
  var aoi: Table users/mamponsah91/SNV_TRAINING/Scafs_Aoi
// Identify the Landsat 8 Surface Reflectance Tier 1 product

var L8Collection = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR');
var aoi = ee.FeatureCollection("users/mamponsah91/SNV_TRAINING/Scafs_Aoi");

```

- Visualize area of interest and centre visualization to your area of interest

```
// Visualize and Center to Area of Interest

//Map.addLayer(aoi, false)
Map.centerObject(aoi, 9);
```

Cloud Masking

- FMASK implements a rule-based algorithm
- Initial spectral test to identify cloud pixels
- Determine cloud probability based on brightness and temperature
- Identify potential cloud shadows using a flood-fill test based on the NIR band
- Estimate cloud height (based on temperature) and identify the shadow associated with each cloud object

Cloud Masking

- Set up the Cloud Masking Function

```
// Set  up the Cloud Masking Function

function maskL8srClouds(image) {
  // Bits 3 and 5 are cloud shadow and cloud, respectively.
  var cloudShadowBitMask = (1 << 3);
  var cloudsBitMask = (1 << 5);
  // Get the pixel QA band.
  var qa = image.select('pixel_qa');
  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
              .and(qa.bitwiseAnd(cloudsBitMask).eq(0));
  return image.updateMask(mask);
}
```

Filtering an ImageCollection

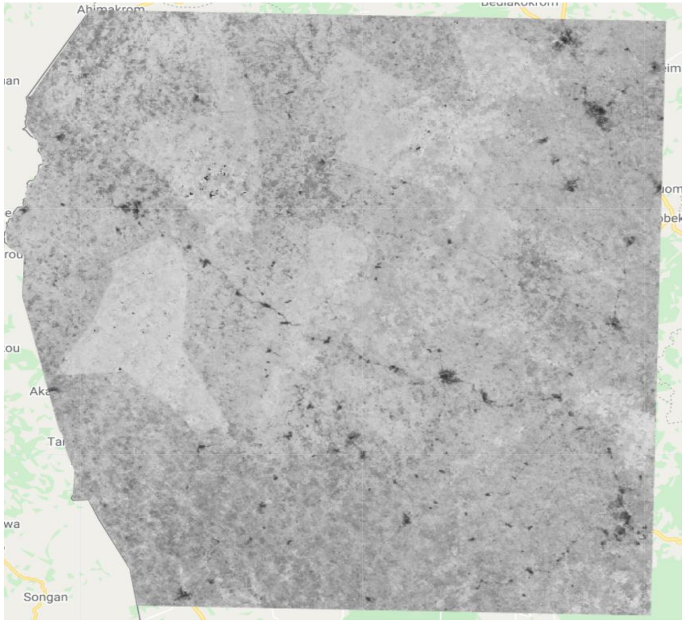
- Build image composite with;
- filterBounds()
- filterDate()
- filterMetadata(cloud cover)
- Map to the cloud mask
- Reducer, median()
- Clip to area of interest
- Visualization of the image composite in false colour.

```
// Building L8 Composite
var L8composite = L8Collection.filterBounds(aoi)
                  .filterDate('2018-01-01', '2019-12-31')
                  .filterMetadata('CLOUD_COVER','less_than',50)
                  .map(maskL8srClouds)
                  .median()
                  .clip(aoi);

Map.addLayer(L8composite, {bands: ['B7', 'B6', 'B4'], min: 0, max: 3000}, 'L8Composite');
```

Calculating NDVI

- Calculate the NDVI using the normalizedDifference() method.
- The normalized difference is computed as (NIR – Red) / (NIR + Red)
- NIR is band 5, and red is band 4.

```
// Compute the Normalized Difference Vegetation Index (NDVI)
var ndvi = L8composite.normalizedDifference(['B5','B4'])
Map.addLayer(ndvi,{min: 0 , max:1},'ndvi', false)

// Other method for calculating NDVI
var nir = L8composite.select('B5');
var red = L8composite.select('B4');
var ndvi = nir.subtract(red).divide(nir.add(red)).rename('NDVI');
Map.addLayer(ndvi,{min: 0 , max:1},'ndvi2', false)
```



NDVI output of Study Area

Tasseled Cap Transformation

- Tasseled Cap Transformation is a method to transform spectral information of satellite data into spectral indicators, useful for vegetation analysis
- Provides an analytical way to detect and compare changes in vegetation, soil, and man-made features over time
- Coefficients used in the linear equation of Tasseled Cap transformation are sensor specific
- The first three bands created are generally held to represent
  - <u>Brightness</u>: measured value for the ground
  - <u>Greenness</u>: measured value for vegetation
  - <u>Wetness</u>: measured value for canopy moisture

Tasseled Cap Transformation

- Derivation of the three levels of information for Landsat is a weighted sum of the bands (without the thermal channel 6), where each band is multiplied by its specific coefficient:

- Brightness = 0.3037 (band 2) + 0.2793 (band 3) + 0.4743 (band 4) + 0.5585 (band 5) + 0.5082 (band 6) + 0.1863 (band 7)
- Greenness = −0.2848 (band 2) − 0.2435 (band 3) − 0.5436 (band 4) + 0.7243 (band 5) + 0.0840 (band 6) − 0.1800 (band 7)
- Wetness = 0.1509 (band 2) + 0.1973 (band 3) + 0.3279 (band 4) + 0.3406 (band 5) − 0.7112 (band 6) − 0.4572 (band 7)

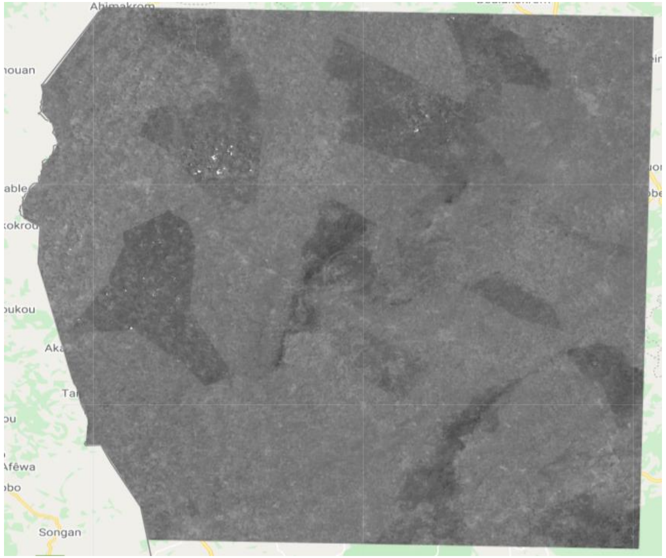
References:

1. Benefoh, D. T., et al (2018). Assessing land-use typologies and change intensities in a structurally complex Ghanaian cocoa landscape. Applied Geography, 99, 109-119
2. Crist, E. P., & Kauth, R. J. (1986). The Tasseled Cap de-mystified.[transformations of MSS and TM data

Tasseled Cap Brightness (TC-B)

- Generating the three tasseled cap bands the same way as done in Benefoh et al., 2018 using the bands; Blue, Green, Red, Near Infrared and the Shortwave Infrared bands of Landsat

```
// Compute the Tasselled Cap Brightness
var TCB = L8composite.expression(
    '(0.3029*BLUE)+(0.2786*GREEN)+(0.4733*RED)+(0.5599*NIR)+(0.508*SWIR1)+(0.1872*SWIR2)', {
      'BLUE':L8composite.select('B2'),
      'GREEN':L8composite.select('B3'),
      'RED':L8composite.select('B4'),
      'NIR':L8composite.select('B5'),
      'SWIR1':L8composite.select('B6'),
      'SWIR2':L8composite.select('B7')
    });
print('TCB', TCB);
Map.addLayer(TCB,{min: 0, max: 7500},'TCB', true)
```

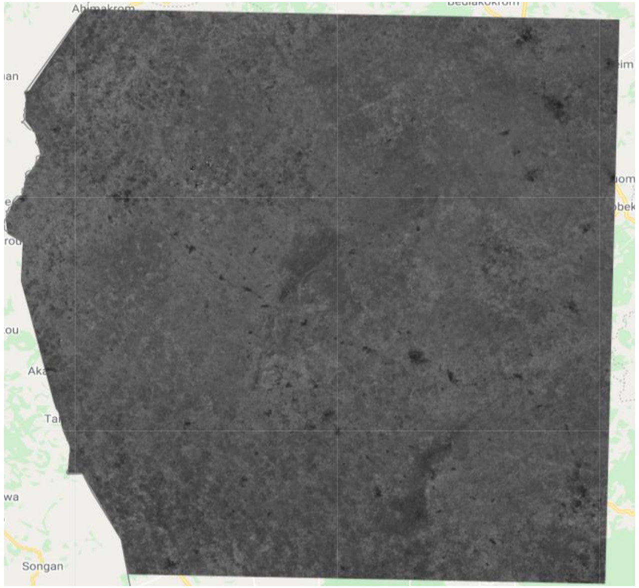


Tasseled Cap Brightness output of study area

Tasselled Cap Greenness (TC-G)

- Compute for tasseled cap greenness in GEE

```
// Compute the Taselled Cap Greenness
var TCG = L8composite.expression(
    '(-0.2941*BLUE)+(-0.243*GREEN)+(-0.5424*RED)+(0.7276*NIR)+(0.0713*SWIR1)+(0.1608*SWIR2)', {
      'BLUE':L8composite.select('B2'),
      'GREEN':L8composite.select('B3'),
      'RED':L8composite.select('B4'),
      'NIR':L8composite.select('B5'),
      'SWIR1':L8composite.select('B6'),
      'SWIR2':L8composite.select('B7')
    });
print('TCG', TCG);
Map.addLayer(TCG,{min: 0, max: 6000}, 'TCG', true)
```

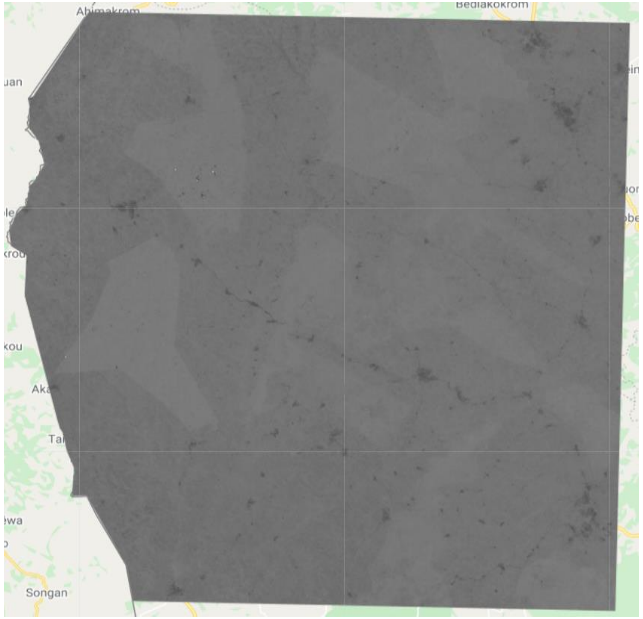


Tasseled Cap Greenness output of Study Area

Tasselled Cap Wetness (TC-W)

- Compute for tasseled cap wetness using derived coefficients for Landsat data

```
// Compute the Tasselled Cap Wetness
 var TCW = L8composite.expression(
    '(0.1511*BLUE)+(0.1973*GREEN)+(-0.3283*RED)+(0.3407*NIR)+(-0.7117*SWIR1)+(-0.4559*SWIR2)', {
      'BLUE':L8composite.select('B2'),
      'GREEN':L8composite.select('B3'),
      'RED':L8composite.select('B4'),
      'NIR':L8composite.select('B5'),
      'SWIR1':L8composite.select('B6'),
      'SWIR2':L8composite.select('B7')
    });
print('TCW', TCW);
Map.addLayer(TCW,{min:-7000, max: 7000},'TCW', true)
```

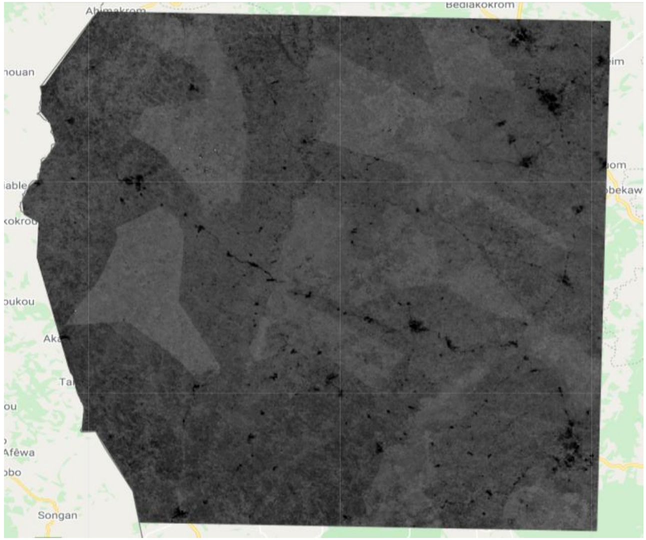


Tasseled Cap Wetness output of study area

Land Surface Water Index (LSWI)

- Calculate the LSWI using the normalizedDifference() method.
- The normalized difference is computed as (NIR – SWIR1) / (NIR + SWIR1)
- NIR is band 5
- Shortwave infrared 1 is band 6.

```
// Compute the Land Surface Water Index (LSWI)
var LSWI = L8composite.normalizedDifference(['B5','B6'])
print('LSWI', LSWI);
Map.addLayer(LSWI,{min: 0, max: 1},'LSWI', true)
```



LSWI output of study area

Link to GEE script

https://code.earthengine.google.com/0a82390c4dcc0f581bd0fbd9573057ef

Hands-on Exercises

Using Sentinel-2 MSI: Multispectral Instrument, Level-1C Image Collection, compute the following indices in Google Earth Engine. Use geometry Point(0.03695166,7.12756658) as point of interest.

- Normalized Difference Water Index (NDWI)
- Normalized Difference Built-up Index (NDBI)
- Enhanced Vegetation Index (EVI)

Thank You