Handbook on Agriculture Monitoring Using Google Earth Engine - part 3

Aniruddha Ghosh (anighosh@ucdavis.edu)

Contents

0.1 Sentinel-1 for crop mapping

In the exercise, we show a thresholding based approach to map rice/crop using Sentinel-1SAR C-band Synthetic Aperture Radar data. For learning basics of radar visit EO College or European Space tutorials. The images were acquired as ground range products with equidistant pixel spacing. To learn more about the data that is available through Google Earth Engine read the data description page.

0.1.1 Background

First we will try to find the areas under crop and then attempt to find the rice areas. Usually during the transplanting phase, flooded fields cause lower backscatter. Backscatter signal continues to increase during the vegetative growth period and then decreases as the crop reaches towards maturity (harvesting). We use the differences in backscattering values during different stages of rice growing seasons to differentiate rice from other crops. This is a relatively simple approach and may not produce very accurate results. The results are sensitive some of the user provided input that requires local knowledge of crop growth.

0.1.2 Data import parameters

At the begining of the script it is a good practice to include the parameters that depends on the user, for example, custom dataset, date range for analysis, user provided model parameters. This practice helps to quickly change the parameters.

```
// Import boundaries from asset
var aoi = ee.FeatureCollection('users/juliusadewopo/MzeTargetRegion_alt_dslv2');

// Set map center to the aoi for making sure we have the correct study area
Map.centerObject(aoi, 11)

// Define period of analysis
var start = '2018-07-01';
var end = '2018-10-31';

var season = ee.Filter.date(start,end);
print(season);
```

Next we use the input parameters to find necessary satellite data for the analysis.

```
// Import Sentinel-1 collection
var sentinel1 =  ee.ImageCollection('COPERNICUS/S1_GRD');


// Filter Sentinel-1 collection for study area, date ranges and polarization components
var sCollection =  sentinel1
                    //filter by aoi and time
                    .filterBounds(aoi)
                    .filter(season)
                    // Filter to get images with VV and VH dual polarization
                    .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
                    .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
                    // Filter to get images collected in interferometric wide swath mode.
                    .filter(ee.Filter.eq('instrumentMode', 'IW'));

// Also filter based on the orbit: descending or ascending mode
var desc = sCollection.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'));
var asc = sCollection.filter(ee.Filter.eq('orbitProperties_pass', 'ASCENDING'));

// Inspect number of tiles returned after the search; we will use the one with more tiles
print("descending tiles ",desc.size());
print("ascending tiles ",asc.size());

// Also Inspect one file
print(asc.first());
```

So far we have no idea how the Sentinel-1 data looks like. Next we show how to create a 3-band RGB like visualization using Sentinel-1 composites.

```
// Create a composite from means at different polarizations and look angles.
var composite = ee.Image.cat([
  asc.select('VH').mean(),
  asc.select('VV').mean(),
  desc.select('VH').mean()
]).focal_median();

// Display as a composite of polarization and backscattering characteristics.
Map.addLayer(composite, {min: [-25, -20, -25], max: [0, 10, 0]}, 'composite');
```

All the tiles returned by the search are not mosaiced or stacked. We first combine (reduce) the tiles to create a single composite image representing one or multiple statistics (e.g. mean, median, standard deviation, percentiles) of the observations.

```
var collection = ee.ImageCollection(asc.select('VV').merge(desc.select('VV')))

// Mean
var mean = collection.reduce(ee.Reducer.mean());

//alternate: var mean = collection.mean();

// Median
var med = collection.reduce(ee.Reducer.median());
```

```
//alternate: var med = collection.median();

// Standard Deviation
var sd = collection.reduce(ee.Reducer.stdDev());

// Percentile
// Lower values---possibily wet
var p10 = collection.reduce(ee.Reducer.percentile([10]));
// Higher values---possibily dry/bright
var p90 = collection.reduce(ee.Reducer.percentile([90]));
```

Now we explore the differences in percentile composites to map areas under crop/paddy rice. If we know locations of few rice fields, we can use the inspector tab to learn the range of values.

```
// compute difference of p90 and p10

var nd_diff = p90.subtract(p10).focal_mean();
// Plot the difference layer to use with inspector
Map.addLayer(nd_diff, {}, 'diff', false);

// Threshold difference value from the inspector from some known location
var th = 7; // Selecting this value is tricky;

var crop_mask = nd_diff.gt(th);
var crop = crop_mask.updateMask(crop_mask);

var cropViz = {palette:"yellow"};
Map.addLayer(crop, cropViz, 'Potential crop areas');

//Focus on some speific plot
var point = ee.Geometry.Point([37.340164603171615, -0.7207780378819623]);
Map.centerObject(point, 13)
```

Export result of the crop areas to Google Drive.

```
var CRS = 'EPSG:4326'; // Only if you want export in specific reference system

Export.image.toDrive({
  image: crop,
  description: 'exporting-crop-to-drive',
  fileNamePrefix: 'rcmrd_crop',
  folder: 'GEE_export', // Name of the Google Drive folder
  scale: 10,
  region: aoi ,
  maxPixels: 1e13,
  crs: CRS
});
```

Another option is to export result of the crop areas to Earth Engine Asset. Having the results as Earth Engine Asset helps in quick visualization and analysis for future use.

```
// Export the image to an Earth Engine asset.

Export.image.toAsset({
  image: crop,
  description: 'exporting-crop-to-Assest',
  assetId: 'users/anighosh/rcmrd/crop',
  scale: 10,
  region: aoi,
  pyramidingPolicy: {
    '.default': 'sample',
  },
  maxPixels: 1e13
});
```

Exercise Repeat the rice mapping exercise for a different year with `VH` polarization data.