Tinkering Lab

GE107

Assignment-3

Name: Vardaan Gupta
Entry number: 2020MEB1328
Department: Mechanical

Date: April 7th , 2022

1. Objective

Use Google Earth Engine (GEE) to compute spectral indices NDVI, study their time series analysis of at least one year by plotting graphs and prepare a brief report.

2. Data to be used:

MODIS

3. Spectral Indices Used:

Normalized Difference Vegetation Index (NDVI) = $\frac{NIR-Red}{NIR+Red}$

Where NIR stands for Near Infrared.

4. Selected State for Analysis: Haryana

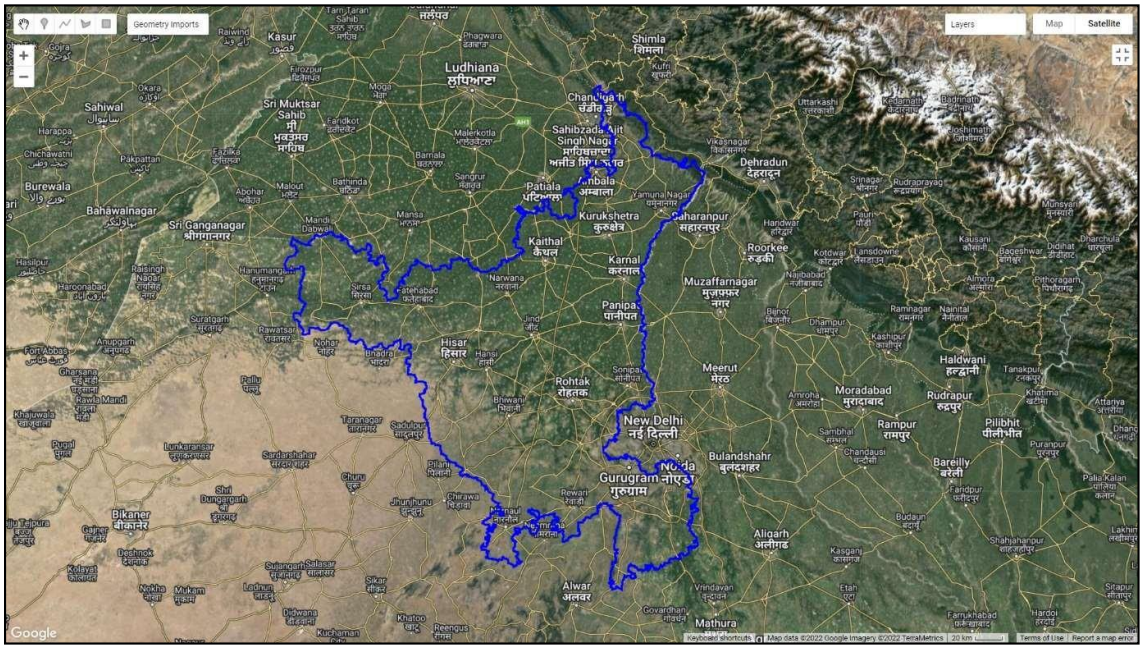

Fig1: Screenshot of Haryana from GEE

5. Time Period:

Jan. 1st 2012 to Dec 31th 2012 (1 year).

6. Procedure:

- Import MODIS from Global Administrative Unit Layers 2015, Second-Level Administrative Units.
- Select 'landcover' band with pixel values 1 which represent.
- Choose State , we choose Haryana.
- We use the Global Administrative Unit Layers (GAUL) dataset to get the state boundary.
- Then plot chart from NDVI data.

7. AOI Layer with Farm Location Layer

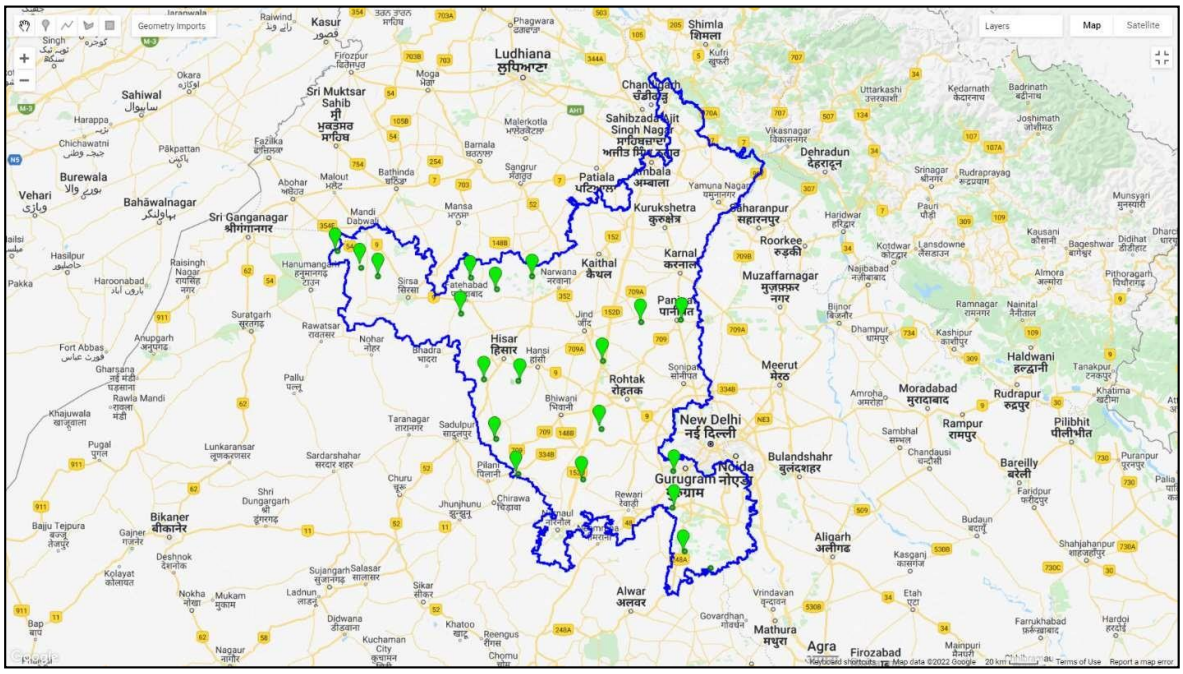

Fig4: AOI and Farm Location Layer

Where green points denote the wheats and rice crops.

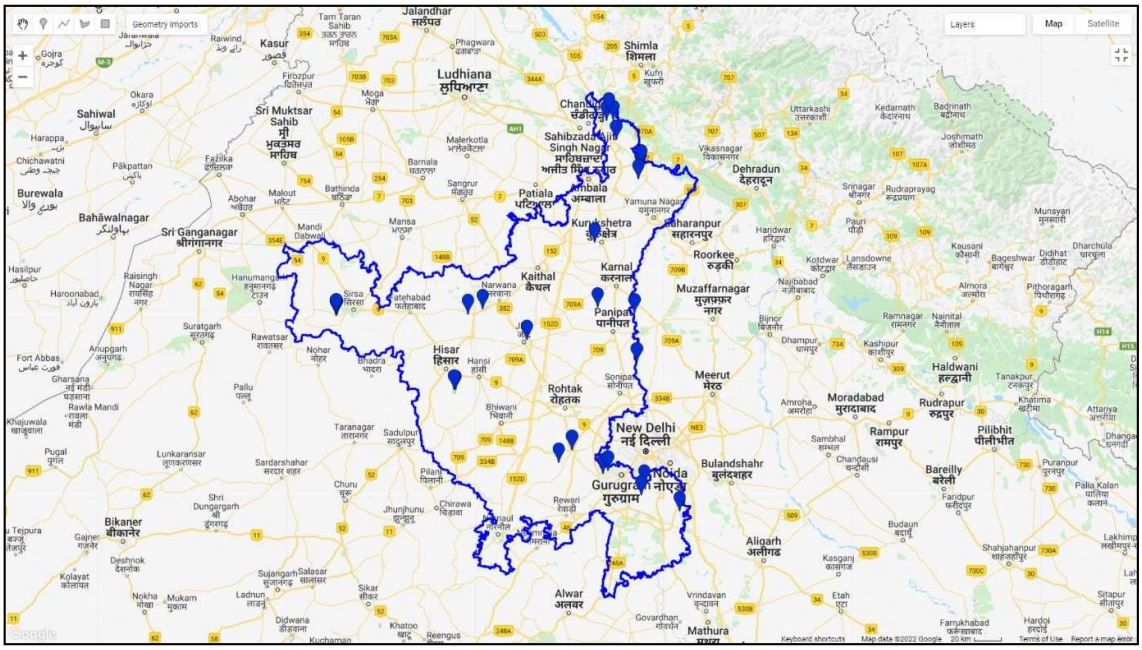

Fig5: Blue denotes the water bodies

8. MODIS Composition Layer

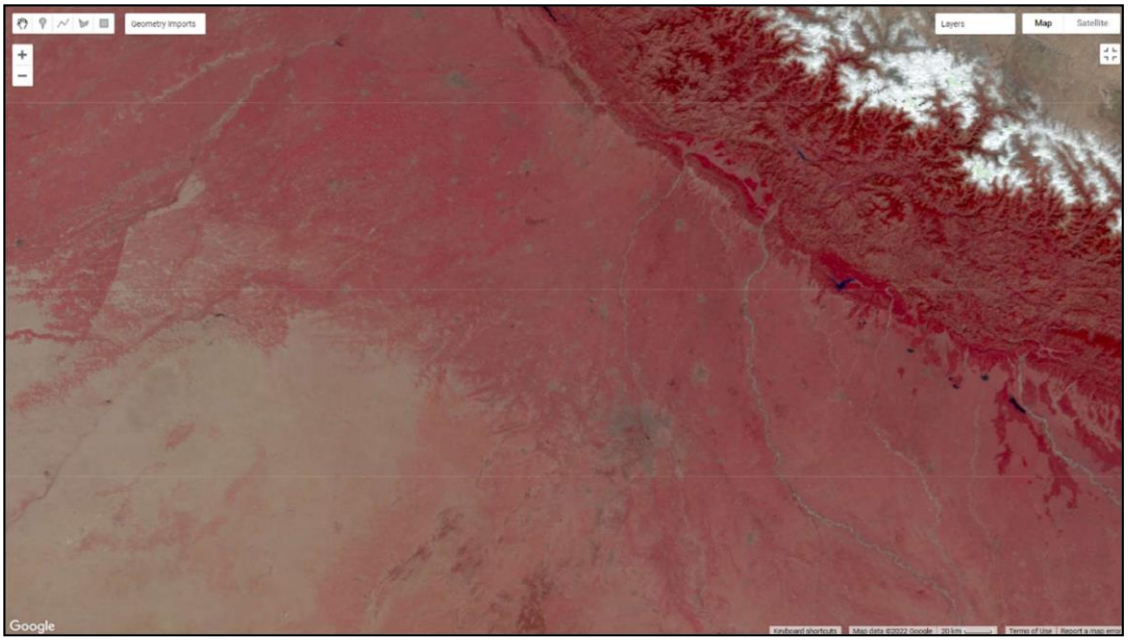

Fig6: MODIS composition Layer

9. All four layer(NDVI):

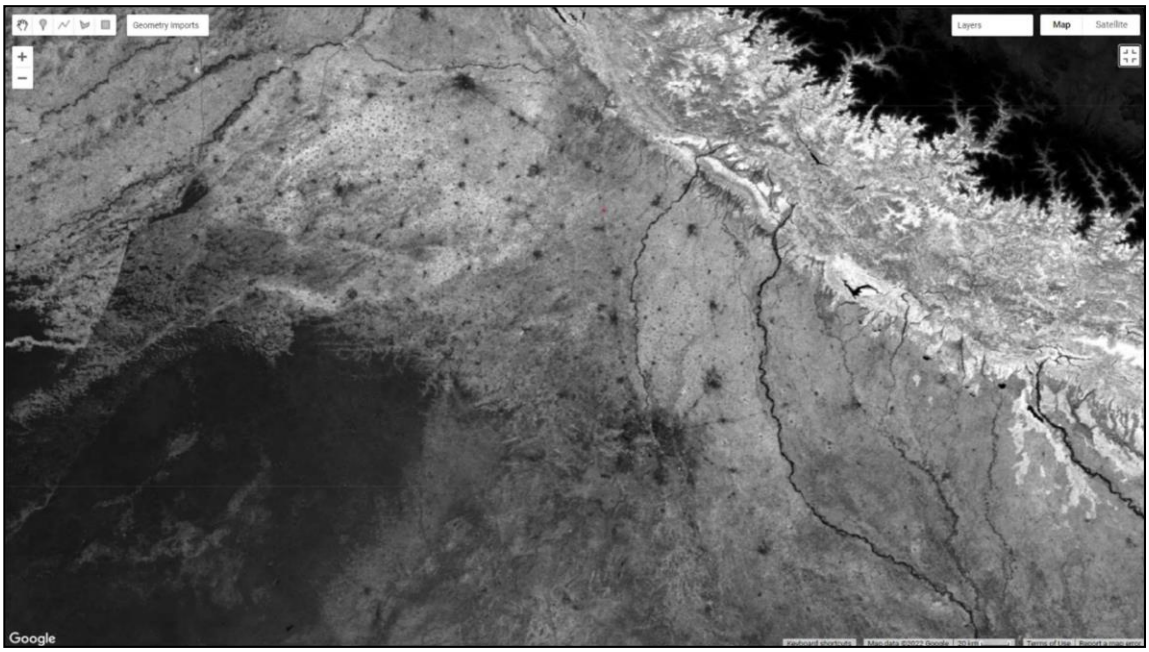

Fig7: All layer together

10. Time Series Graphs:

- **Band values in different regions:**

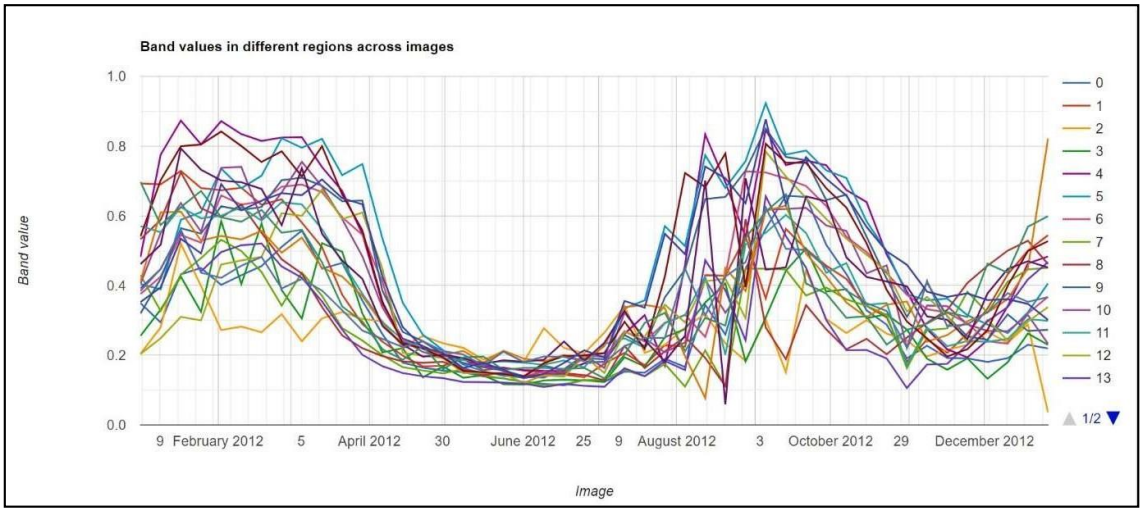

Fig8: Band Values in Different regions

- **NDVI over time at a Single Location**

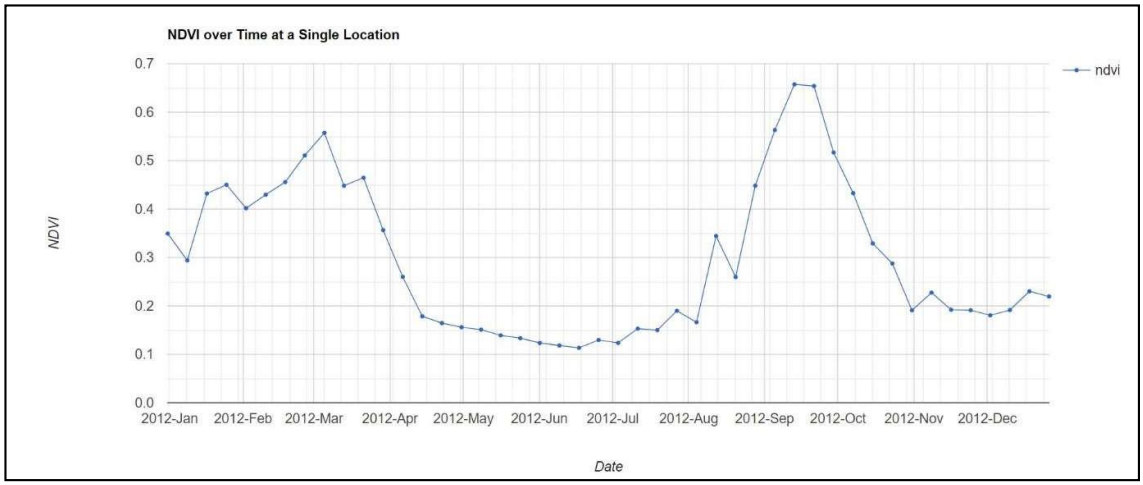

Fig9: Band Values in Different regions

CSV file links:
https://drive.google.com/drive/u/0/folders/1Rn9aUzNNPF6eCydFHizPyluiWUpiNNux

11. GEE Code

Link:
https://code.earthengine.google.com/8259e3cf864050e663a4b48a5a7df936?accept_repo=user%20%20s%2Ftylere%2Fg4g18-ee101


```
Map.setCenter = (76.77,30.73, 11);

// Select 'landcover' band with pixel values 1 which represent

//Rice and Wheat Rainfed crops

var wheatrice = gfsad.select('landcover').eq(1)

// State :Haryana

// We use the Global Administrative Unit Layers (GAUL) dataset to get the state boundary

var haryana = gaul.filter(ee.Filter.eq('ADM1_NAME', 'Haryana'))

// wheatrice image contains 1 and 0 pixels. We want to generate points

// only in the pixels that are 1 (representing crop areas)

// selfMask() masks the pixels with 0 value.
```

```
var points = wheatrice.selfMask().stratifiedSample({numPoints:20, region:haryana,
geometries: true})
// We need a unique id for each point. We take the feature id and set it s
// a property so we can refer to each point easily
var points = points.map(function(feature) {
  return ee.Feature(feature.geometry(), {'id': feature.id()})
})
// Show the state polygon with a blue outline
var outline = ee.Image().byte().paint({
featureCollection: haryana,
  color: 1,
  width: 3
});
Map.addLayer(outline, {palette: ['blue']}, 'AOI')
// Show the farm locations in green
Map.addLayer(points, {color: 'green'}, 'Farm Locations')
//define the time period
var startDate = '2012-01-01'
var endDate = '2012-12-31'
// bands
var modisBands =
['sur_refl_b03','sur_refl_b04','sur_refl_b01','sur_refl_b02','sur_refl_b06','sur_refl_b07'];
var lsBands = ['blue','green','red','nir','swir1','swir2'];
// helper function to extract the QA bits
function getQABits(image, start, end, newName) {
    // Compute the bits we need to extract.
    var pattern = 0;
    for (var i = start; i <= end; i++) {
      pattern += Math.pow(2, i);
    }
    // Return a single band image of the extracted QA bits, giving the band a new name.
    return image.select([0], [newName])
```

```
                .bitwiseAnd(pattern)
                .rightShift(start);
}
// A function to mask out cloudy pixels.
function maskQuality(image) {
  // Select the QA band.
  var QA = image.select('StateQA');
  // Get the internal_cloud_algorithm_flag bit.
  var internalQuality = getQABits(QA,8, 13, 'internal_quality_flag');
  // Return an image masking out cloudy areas.
  return image.updateMask(internalQuality.eq(0));
}

// create cloud free composite
var noCloud = modis.filterDate(startDate,endDate)
                        .map(maskQuality)
                        .select(modisBands,lsBands)
                        .filter(ee.Filter.bounds(points))
// vis parameters
var visParams = {bands:['nir','red','green'],min:0,max:3000,gamma:1.3};
// add the cloud free composite
Map.addLayer(noCloud.median(),visParams,'MODIS Composite');
// Adding a NDVI band
function addNDVI(noCloud){
  var ndvi = noCloud.normalizedDifference(['sur_refl_b02', 'sur_refl_b01']).rename('ndvi')
  return noCloud.addBands([ndvi])
 }
var collection = modis.filterDate(startDate, endDate)
    .map(addNDVI)
    //.filter(ee.Filter.bounds(points))
// View the median composite
```

```
var vizParams = {bands: ['ndvi'], min: -1, max: 1}
Map.addLayer(collection.median(), vizParams, 'collection')

var testPoint = ee.Feature(points.first())
Map.centerObject(testPoint, 10)
var chart = ui.Chart.image.series({
   imageCollection: collection.select('ndvi'),
   region: testPoint.geometry()
   }).setOptions({
     interpolateNulls: true,
     lineWidth: 1,
     pointSize: 3,
     title: 'NDVI over Time at a Single Location',
     vAxis: {title: 'NDVI'},
     hAxis: {title: 'Date', format: 'YYYY-MMM', gridlines: {count: 12}}
   })
print(chart)

var chart = ui.Chart.image.seriesByRegion({
   imageCollection: collection.select('ndvi'),
   regions: points,
   reducer: ee.Reducer.mean()
})
print(chart)
```