IIT ROPAR

GE107: TINKERING LAB

ASSIGNMENT 3

GOOGLE EARTH ENGINE

NAME: **SEHAJPREET SINGH**

DATE: 7/04/2022

ENTRY NO: 2020MEB1314

1. OBJECTIVE:

Use Google Earth Engine (GEE) to compute spectral indices such as NDVI, NDWI, NDSI, NDSII, NDGI, NDBI, and other indices for any area, analyse their time series analysis for at least one year by plotting graphs, and write a concise report.

2. PROJECT:

For the project I chose to do normalised difference built-up index analysis (NDBI) on the map of Delhi.

$$\mathbf{NDBI} = \frac{(SWIR - NIR)}{(SWIR + NIR)}$$

Short-wave infrared rays (wavelength = 0.9 to 1.7 um) and Near-infrared rays (wavelength = 0.7 to 1 um) are the two types of infrared rays.

We have to develop a GEE algorithm first to get the spectral maps and time series graph. For the project, we used Landsat data and GHSL data extracted from Landsat data. We got a variety of maps and graphs after running it.

3. GEE CODE:

```
var modis = ee.ImageCollection("MODIS/MOD09A1"),
    gaul = ee.FeatureCollection("FAO/GAUL/2015/level1"),
    gfsad = ee.Image("USGS/GFSAD1000_V0");

Map.setCenter = (75.379257,29.988245,15);
// Select 'landcover' band with pixel values 1 which represent
//Rice and Wheat Rainfed crops
var wheatrice = gfsad.select('landcover').eq(1)
// punjab
// We use the Global Administrative Unit Layers (GAUL) dataset to get the state boundary
var punjab = gaul.filter(ee.Filter.eq('ADM1_NAME', 'Punjab'))
// 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:100, region:punjab, geometries: true} )
// We need a unique id for each point. We take the feature id and set it as
// 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: punjab,
  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 = '2015-01-01'
var endDate = '2015-12-31'
```

```
// 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)
```

4. 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.

5. OBSERVATIONS:

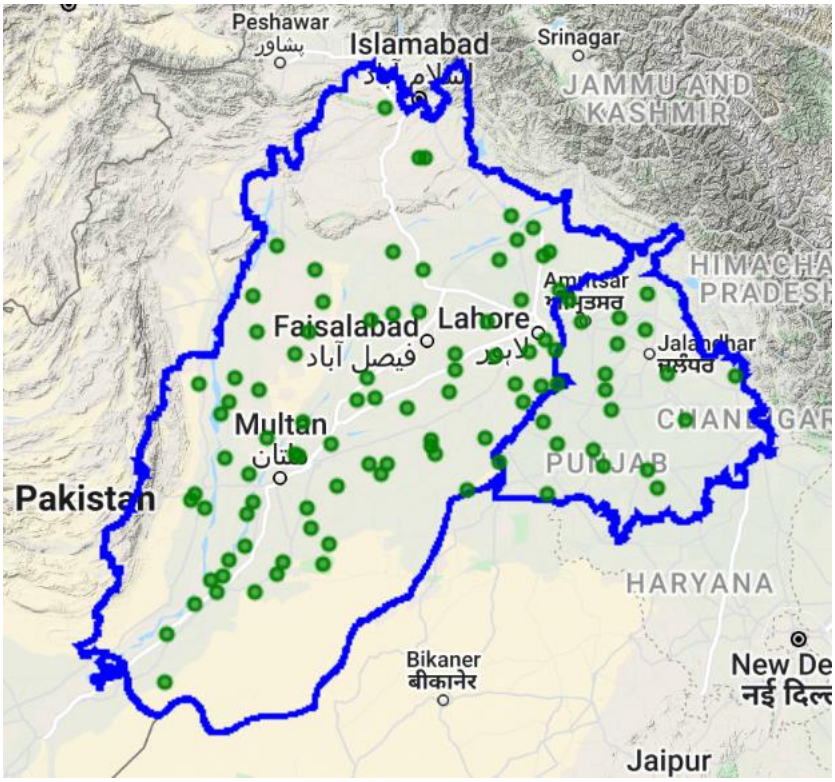


Figure 1: AOI and Farm Location Layer

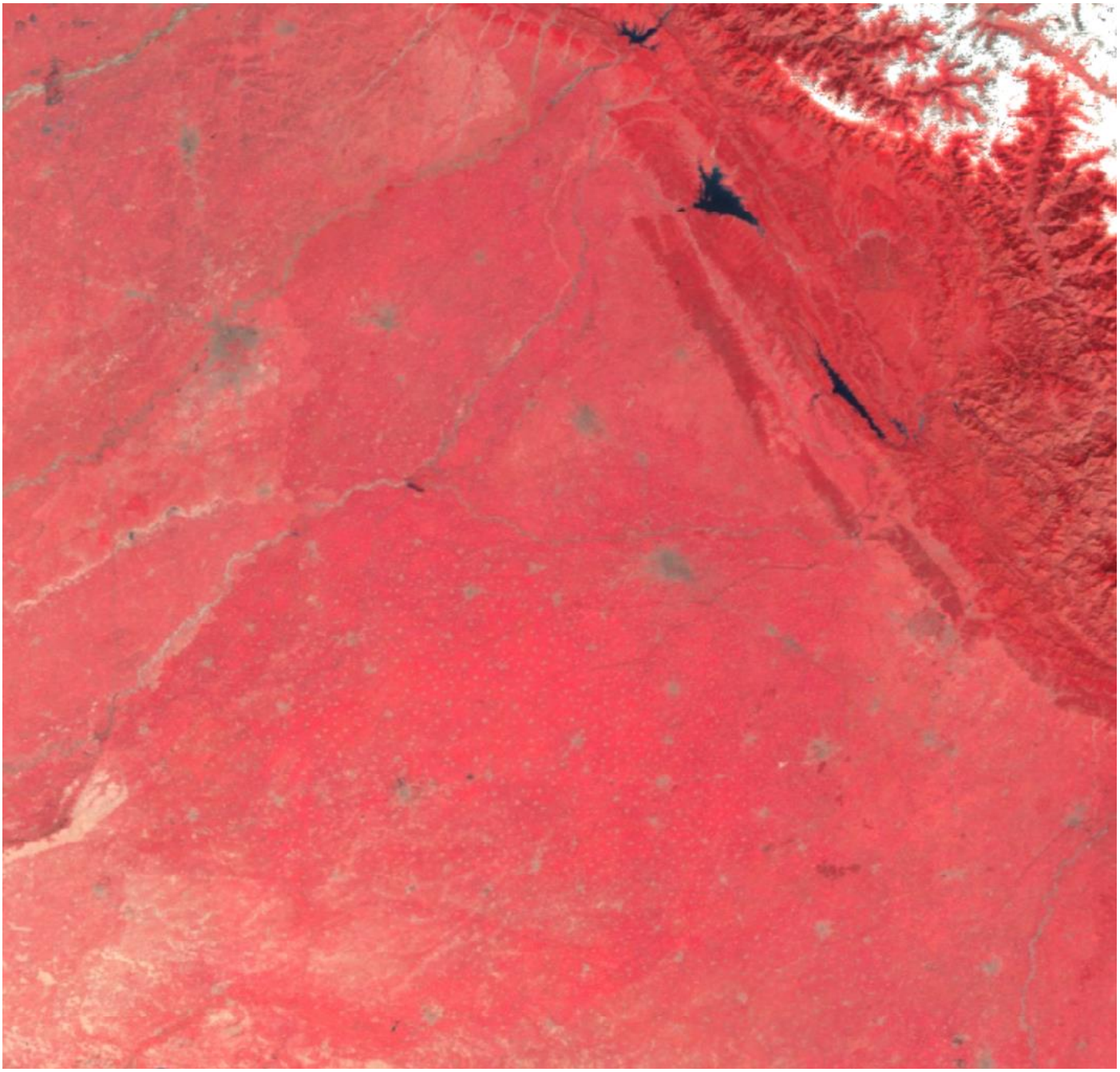


Figure 2: MODIS composition Layer

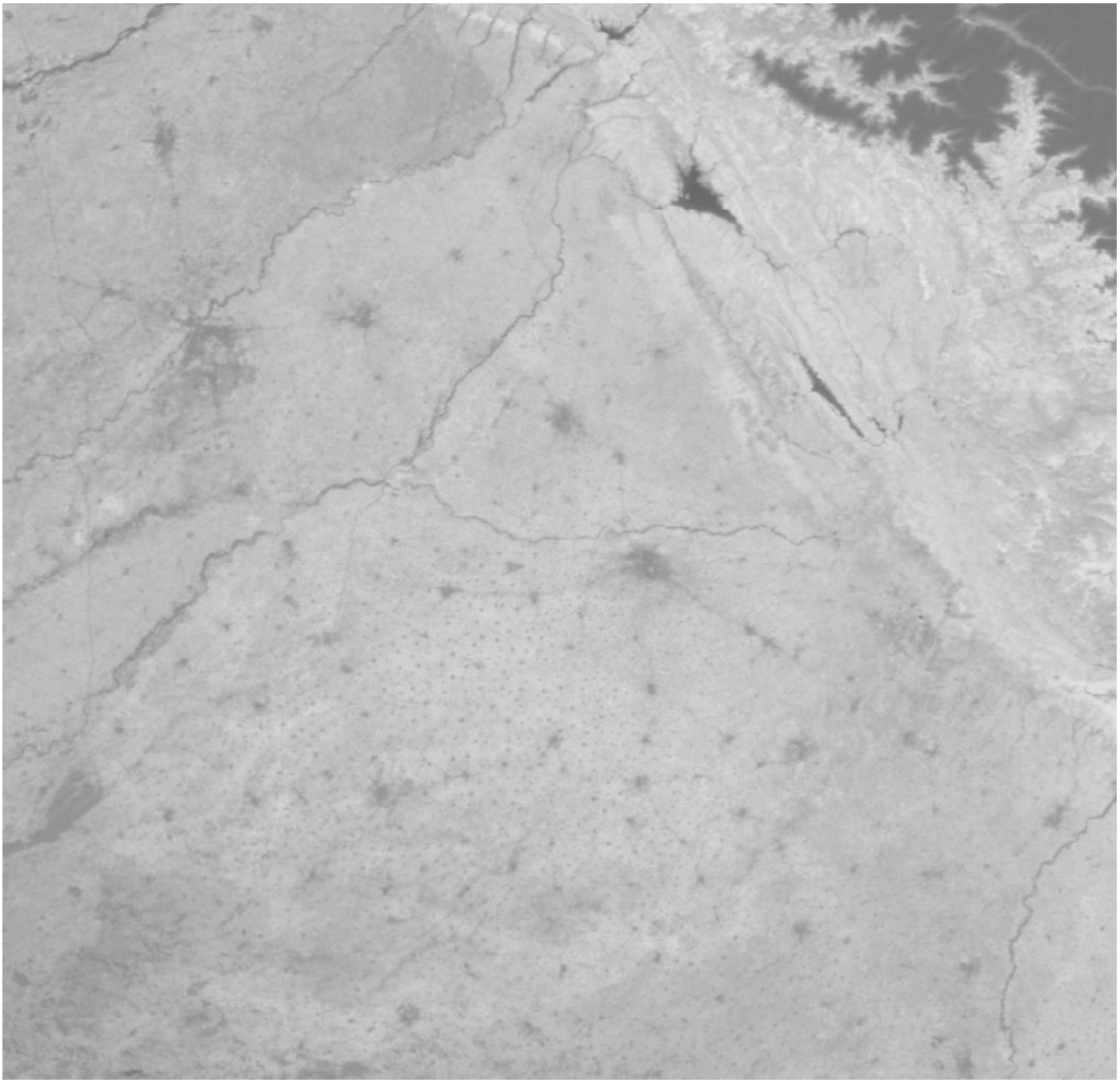
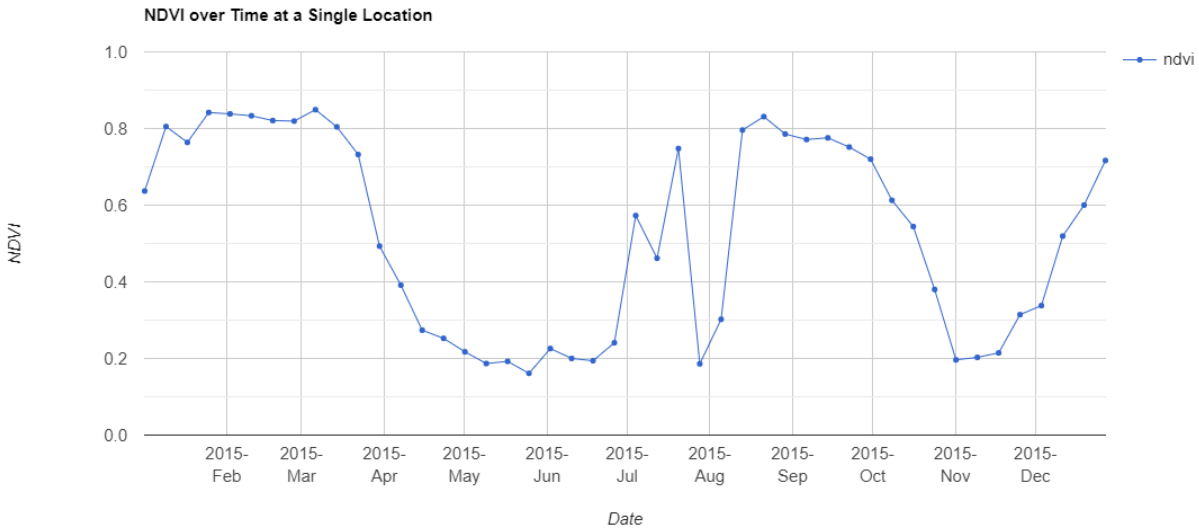
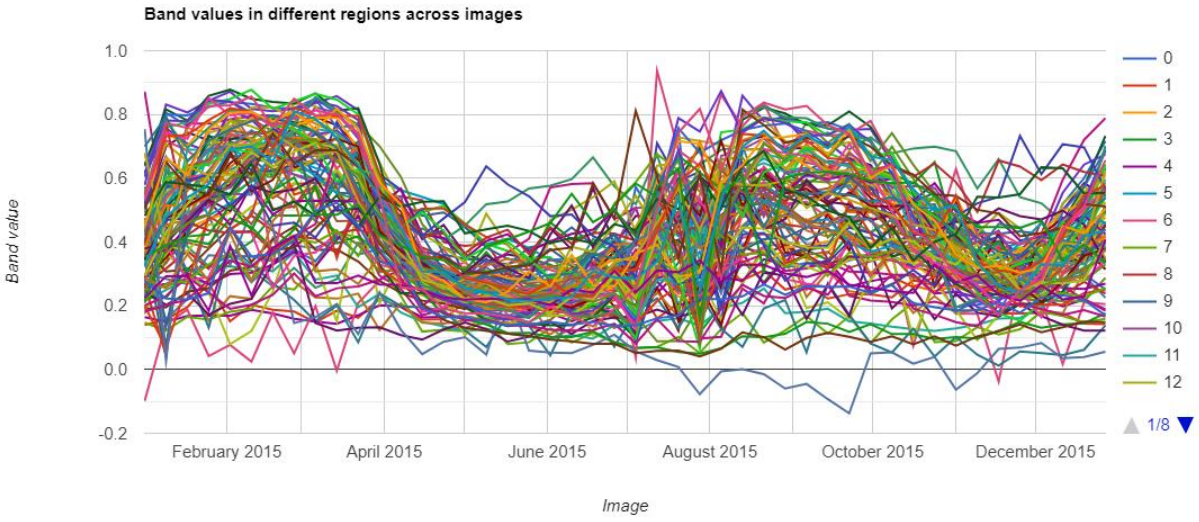


Figure 3 All Layers together

We outlined the boundary of Punjab (I don’t know why it is taking Pakistan with it !).





Link: https://code.earthengine.google.co.in/b6fa93d76059295a253e79689fd2bb28