**Lab 2: Image Classification (Total: 32)**

Q1: Zone 10, 2020 August 14, 1.15% **(3)**

Screenshot of NDVI over UBC campus **(3)**

Q3: Healthy vegetation absorbs red light while reflecting NIR light. The Normalized Difference Vegetation Index (NDVI) leverages this disparity by calculating the ratio of NIR to red reflectance, providing an estimate of vegetation productivity. High NDVI values correspond to dense, healthy vegetation, while low values indicate less productive or sparse vegetation. **(4)**

Q4: Water: negative NDVI values  
 Urban: 0.1-0.3  
 Veg: 0.3+ **(3)**

Q5: In supervised classification, the user provides pre-labeled training samples to the classification algorithm, which then uses these labeled samples to classify the entire image. In contrast, unsupervised classification does not rely on pre-labeled samples. Instead, it automatically groups pixels into classes based on their spectral similarities using clustering algorithms. Unsupervised classification is exploratory and can identify patterns or classes in the data without prior knowledge, but the user needs to interpret and assign meaning to the resulting classes. **(4)**

Q6: Screenshot of final classified image with legend visible **(4)**  
Water is most accurate because of very low reflectance and negative NDVI, it is easy to separate from the other classes. Agriculture and forest are the most confused because they represent vegetated surfaces, thus have similar NDVI and greenness. **(6)**

**Q7: Annotated code (5) – each code block should have at least one comment that accurately describes the process. Example code below:**

//filter and display an image

var image = ee.Image(ee.ImageCollection('LANDSAT/LC08/C02/T1\_L2')

.filterBounds(geometry)

.filterDate('2020-06-01', '2022-08-31')

.sort('CLOUD\_COVER')

.first());

print(image);

//add filtered scene to map

Map.addLayer(image, {bands: ["SR\_B4","SR\_B3","SR\_B2"], min:7000, max: 15000}, 'filteredScene');

//CALCULATE NDVI (nir-red)/(nir+red)

var red = image.select('SR\_B4');

var nir = image.select('SR\_B5');

var ndvi = nir.subtract(red).divide(nir.add(red)).rename('NDVI');

//add NDVI layer to map

Map.addLayer(ndvi, {min:0, max: 1}, 'NDVI');

//Merge feature collections representing training data

var classNames = Water.merge(Urban).merge(Agriculture).merge(Forest);

//add the NDVI layer to the image

var image = image.addBands(ndvi);

print('ClassNames', classNames);

//Select training pixels from spectral bands/NDVI layers

var bands = ['SR\_B2', 'SR\_B3', 'SR\_B4', 'SR\_B5', 'SR\_B6', 'SR\_B7', 'NDVI'];

var training = image.select(bands).sampleRegions({

collection: classNames,

properties: ['landcover'],

scale: 30

});

print(training.first())

//Train the classificaiton model using minimum distance algorithm

var classifier = ee.Classifier.minimumDistance().train({

features: training,

classProperty: 'landcover',

inputProperties: bands

});

// Apply the model to the filtered image and add to map

var classified = image.select(bands).classify(classifier);

Map.addLayer(classified,

{min: 0, max: 3, palette: ['orange', 'green', 'blue','yellow']},

'classification');