

// Geometry

var aoi = geometry;

// Planetscope median composite

var image = ee.ImageCollection("COPERNICUS/S2").filterBounds(aoi)

.filterDate('2022-01-01', '2023-03-31') // Adjust date range if needed

.map(cloudMask)

.median()

.clip(aoi)

.divide(10000);

Map.addLayer(image, {bands: ['B4', 'B3', 'B2'], min: 0, max: 0.3}, 'Sentinel-2');

function cloudMask(image){

return image.updateMask(image.select('QA60').not());

}

// Seeds

var seeds = ee.Algorithms.Image.Segmentation.seedGrid(15, 'hex');

// SNIC segmentation

var segment = ee.Algorithms.Image.Segmentation.SNIC({

image: image,

compactness: 0,

seeds: seeds,

//tileScale: 5 // Adjust this value to manage memory usage

}).reproject('EPSG:4326', null , 100);

Map.addLayer(segment.select('clusters').randomVisualizer(), {}, 'Clusters', false);

Map.addLayer(segment, {bands: ['B4\_mean', 'B3\_mean', 'B2\_mean'], min: 0, max: 0.2}, 'Segment');

// Image for classification

var imageObject = segment.select(['B.\*']);

var bandsName = imageObject.bandNames();

// Sample

var sample = Housing.merge(sea).merge(river).merge(sea).merge(RoadIntersection).merge(grassland).merge(road)

.merge(industrial);

// Trained

var trained = imageObject.sampleRegions({

collection: sample,

scale: 150,

properties: ['class'],

tileScale: 15 // Adjust this value to match the segmentation tileScale

});

// Classifier

var classifier = ee.Classifier.smileRandomForest(50).train(trained, 'class', bandsName);

// Land cover properties

var value = [1, 2, 3, 4, 5, 6, 7, 8, 9];

var classPalette = ['3366FF', '66FF66', 'FFC0CB', 'ADFF2F', '87CEFA','FFCC66', 'FF3366', '996633', 'FFFF66']; // Adjust as needed

// Classify

var landCover = imageObject.classify(classifier).rename('LULC')

.set('LULC\_class\_palette', classPalette, 'LULC\_class\_values', value);

Map.addLayer(landCover, {}, 'LULC');