

Mangroves II - Change Mapping (A3.3)

Authors

Celio de Sousa, David Lagomasino, and Lola Fatoyinbo

Overview

The purpose of this chapter is to present two methods of land cover extent change detection: map-to-map and map-to-image using anomaly data. In a map-to-map approach, changes are extracted by subtracting a two-date pair of land cover extent maps: that is, time 2 (T2) extent minus time 1 (T1) extent. By comparison, a map-to-image approach uses a baseline extent map within which changes are calculated based on a T2 image where the change classes are defined by threshold values. In this lab, we will perform a map-to-map change detection between two mangrove extent maps from 2000 and 2020, and also perform a vegetation index anomaly analysis to detect changes within a mangrove extent map from the year 2000 in Guinea, West Africa.

Learning Outcomes

- Performing change detection by contrasting two preexistent mangrove extent maps.
- Harmonizing across Landsat generations to create consistent long-term series
- Calculating the anomaly of a given vegetation index based on its long-term average value.
- Performing change detection by interpreting vegetation index anomalies

Helps if you know how to:

- Recognize similarities and differences among satellite spectral bands (Part F1, Part F2, Part F3).
- Perform basic image analysis: select bands, compute indices, create masks (Part F2).
- Use `normalizedDifference` to calculate vegetation indices (Chap. F2.0).
- Perform a supervised image classification (Chap. F2.1).
- Work with array images (Chap. F3.1, Chap. F4.6).
- Use expressions to perform calculations on image bands (Chap. F3.1).
- Summarize an image with `reduceRegion` (Chap. F3.1).
- Write a function and `map` it over an `ImageCollection` (Chap. F4.0).
- Mask cloud, cloud shadow, snow/ice, and other undesired pixels (Chap. F4.3).

- Perform a two-period change detection (Chap. F4.4).

Introduction to Theory

Mangrove forests are among the most productive ecosystems on Earth, providing a wide range of critical economic, ecological, and societal services. However, mangroves worldwide have undergone intense conversion, with human-related disturbances being one of the main causes of global mangrove loss in recent decades (Goldberg 2020). Thus, regular monitoring of the dynamics of mangrove ecosystems worldwide is key for establishing better coastal management policies and mangrove conservation initiatives.

Readily available, low-cost, and repeated-coverage remote sensing data provides numerous advantages for monitoring mangrove forests and detecting changes in the landscape over time. In this context, the Landsat data archive, which covers more than 35 years, has been widely used for mapping land cover dynamics worldwide. Most notably, the Landsat mission coupled with Google Earth Engine's computing infrastructure have been leveraged to develop widely used global datasets at 30 m spatial resolution, such as the global forest cover (Hansen et al. 2013) and global mangrove extent (Bunting et al. 2018) datasets.

However, as extensively explored in the literature, change detection using remotely sensed data is challenged by several factors, including (but not limited to) the following: spatial, spectral, thematic, and temporal constraints, atmospheric conditions, high cloud-coverage, and differences in sensors' spectral characteristics. These differences may have a direct effect on the ability to accurately detect and monitor changes in the landscape, including mangrove forests. As explored by Roy et al. (2016), the Landsat TM/ETM+ and OLI sensors present small but potentially significant differences between their spectral characteristics, with the greatest differences in the near-infrared (NIR) and the shortwave infrared (SWIR) bands—the most important spectral bands for vegetation studies.

In this chapter, we take advantage of the statistical functions presented in Roy et al. (2016) to transform between the comparable TM/ETM+ and OLI bands in order to ensure inter-sensor harmonized spectral information and temporal continuity. We will use this harmonized Landsat TM/ETM+/OLI collection to derive anomaly-based changes in mangrove extent over 21 years (2000–2020) in Guinea, West Africa.

Practicum

Section 1. Map-to-Map Change Detection

If you have not already done so, you can add the book's code repository to the Code Editor by entering

https://code.earthengine.google.com/?accept_repo=projects/gee-edu/book (or the short URL bit.ly/EEFA-repo) into your browser. The book's scripts will then be available in the script manager panel to view, run, or modify. If you have trouble finding the repo, you can visit bit.ly/EEFA-repo-help for help.

Previous chapters demonstrated how to perform a general supervised classification, as well as how to apply those classifiers and functions to create a mangrove extent map (Chap. A3.2). In this section, we will build upon those techniques to perform a map-to-map change analysis. Paste the code below into a new script, which will access pre-created assets for two time periods: 2000 and 2020.

```
var area0fstudy = ee.FeatureCollection(  
    'projects/gee-book/assets/A3-3/Border5km');  
var mangrove2000 = ee.Image(  
    'projects/gee-book/assets/A3-3/MangroveGuinea2000_v2');  
var mangrove2020 = ee.Image(  
    'projects/gee-book/assets/A3-3/MangroveGuinea2020_v2');
```

Start by setting the map center around Conakry, Guinea, and adding your 2000 and 2020 mangrove extent to the map using a color of your choice and naming them '['Mangrove Extent 2000'](#)' and '['Mangrove Extent 2020'](#)':

```
Map.setCenter(-13.6007, 9.6295, 10);  
// Sets the map center to Conakry, Guinea  
Map.addLayer(area0fstudy, {}, 'Area of Study');  
Map.addLayer(mangrove2000, {  
    palette: '#16a596'  
}, 'Mangrove Extent 2000');  
Map.addLayer(mangrove2020, {  
    palette: '#9ad3bc'  
}, 'Mangrove Extent 2020');
```

Because the assets have the value of 1 (one) assigned to mangrove pixels and 0 (zero) to everything else, you can derive losses and gains from both mangrove extent maps

with a simple subtraction of T1 (2000) from T2 (2020). The value of 1 has been assigned to mangrove pixels and everything else has been masked. For the mathematical operation between the two layers to work, every pixel has to have a value assigned to it. In this case, you can unmask previously masked pixels and assign the value 0 to them using `unmask(0)`. Finally, subtract T1 from T2 into a new variable `change`:

```
var mang2020 = mangrove2020.unmask(0);
var mang2000 = mangrove2000.unmask(0);
var change = mang2020.subtract(mang2000)
  .clip(areaOfstudy);
```

Pixels in the raster `change` will have values of -1 , 0 , or 1 , which represent loss/conversion, no change, and gains, respectively:

- -1 = no mangroves in 2020, mangroves in 2000 ($0 - 1 = -1$);
- 0 = mangroves in 2020, mangroves in 2000 ($1 - 1 = 0$);
- 1 = mangroves in 2020, no mangroves in 2000 ($1 - 0 = 1$)

Finally, add `change` to the map:

```
var paletteCHANGE = [
  'red', // Loss/conversion
  'white', // No Change
  'green', // Gain/Expansion
];

Map.addLayer(change, {
  min: -1,
  max: 1,
  palette: paletteCHANGE
}, 'Changes 2000-2020');
```

You can calculate the area of expansion/conversion by isolating the pixels of gain/loss from the `change` into `gain` and `loss`. Then, calculate the area of each pixel using `ee.Image.pixelArea` and multiplying by the count of pixels in `gain` and `loss` using `multiply`. The default unit is square meters (m^2). You can use `divide` to transform into square kilometers (`divide(1000000)`) or hectares (`divide(10000)`). Finally, use

`ee.Reducer.sum` to sum all area values for both `gain` and `loss` and print them to the **Console** (Fig. A3.3.1).

```
// Calculate the area of each pixel
var gain = change.eq(1);
var loss = change.eq(-1);

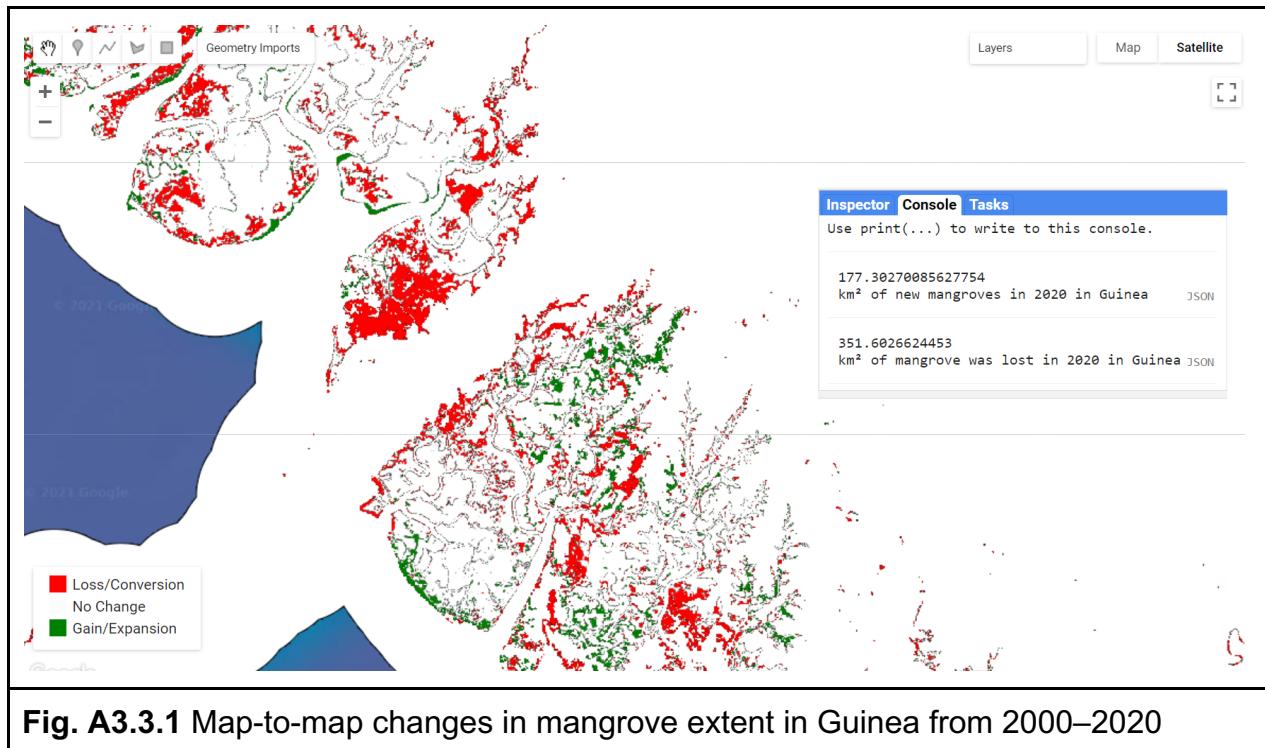
var gainArea = gain.multiply(ee.Image.pixelArea().divide(1000000));
var lossArea = loss.multiply(ee.Image.pixelArea().divide(1000000));

// Sum all the areas
var statsgain = gainArea.reduceRegion({
  reducer: ee.Reducer.sum(),
  scale: 30,
  maxPixels: 1e14
});

var statsloss = lossArea.reduceRegion({
  reducer: ee.Reducer.sum(),
  scale: 30,
  maxPixels: 1e14
});

print(statsgain.get('classification'),
  'km2 of new mangroves in 2020');
print(statsloss.get('classification'),
  'of mangrove was lost in 2020');

Map.addLayer(gain.selfMask(), {
  palette: 'green'
}, 'Gains');
Map.addLayer(loss.selfMask(), {
  palette: 'red'
}, 'Loss');
```



Code Checkpoint A33a. The book’s repository contains a script that shows what your code should look like at this point.

Question 1. What are some of the issues that may arise when using a map-to-map change detection? Explain how these different dates may affect the classification output and, consequently, the change output.

Section 2. Map-to-Image Change Detection

Using the mangrove extent you created in the previous section, we will look at another way to detect changes without having to classify an image. In this case, change classes are defined by threshold values of a specific metric, such as a vegetation index or a spectral image band. In this section, we will take advantage of the Landsat archive to create an average reference value of a given vegetation index at an earlier date T1 and see how it compares to its later value at T2.

The first assumption of this approach is that changes will happen within a buffer zone from the baseline extent. Start by setting the baseline extent and buffer zone using `focal_max`:

```
var buffer = 1000; // In meters
var extentBuffer = mangrove2000.focal_max(buffer, 'circle', 'meters');
```

```

Map.addLayer(mangrove2000, {
  palette: '#000000'
}, 'Baseline', false);
Map.addLayer(extentBuffer, {
  palette: '#0e49b5',
  opacity: 0.3
}, 'Mangrove Buffer', false);

```

Harmonizing Landsat 5/7/8 Image Collections

As described in the beginning of this chapter, the Landsat TM/ETM+ and OLI sensors present differences between their spectral characteristics. Thus, to ensure inter-sensor harmonized spectral information and temporal continuity, we will harmonize the entire Landsat image archive using the statistical functions presented in Roy et al. (2016). For that, start by defining the temporal parameters for the harmonization:

```

var startYear = 1984;
var endyear = 2020;
var startDay = '01-01';
var endDay = '12-31';

```

Next, we will create several functions for the harmonization of the Landsat archive:

Harmonization Function: `harmonizationRoy` uses the regression coefficients (slopes and intercepts) retrieved from Roy et al. (2016) and performs a linear transformation of ETM+ spectral space to OLI spectral space:

```

var harmonizationRoy = function(oli) {
  var slopes = ee.Image.constant([0.9785, 0.9542, 0.9825,
    1.0073, 1.0171, 0.9949
  ]);
  var itcp = ee.Image.constant([-0.0095, -0.0016, -0.0022,
    -0.0021, -0.0030, 0.0029
  ]);
  var y = oli.select(['B2', 'B3', 'B4', 'B5', 'B6', 'B7'], [
    'B1', 'B2', 'B3', 'B4', 'B5', 'B7'
  ])
    .resample('bicubic')
    .subtract(itcp.multiply(10000)).divide(slopes)
    .set('system:time_start', oli.get('system:time_start'));
  return y.toShort();
}

```

```
};
```

Retrieve a Particular Sensor Function: `getSRcollection` will be used to retrieve individual sensor collections based on the temporal parameters and the harmonization function above. Additionally, this function will mask cloud, cloud shadow, and snow based on the Landsat quality assessment bands:

```
var getSRcollection = function(year, startDay, endYear, endDay,
    sensor) {
  var srCollection = ee.ImageCollection('LANDSAT/' + sensor +
    '/C01/T1_SR')
    .filterDate(year + '-' + startDay, endYear + '-' + endDay)
    .map(function(img) {
      var dat;
      if (sensor == 'LC08') {
        dat = harmonizationRoy(img.unmask());
      } else {
        dat = img.select(['B1', 'B2', 'B3', 'B4',
          'B5', 'B7']
        ).unmask()
        .resample('bicubic')
        .set('system:time_start', img.get(
          'system:time_start'));
      }
      // Cloud, cloud shadow and snow mask.
      var qa = img.select('pixel_qa');
      var mask = qa.bitwiseAnd(8).eq(0).and(
        qa.bitwiseAnd(16).eq(0)).and(
        qa.bitwiseAnd(32).eq(0));
      return dat.mask(mask);
    });
  return srCollection;
};
```

Combining the Collections Function: `getCombinedSRcollection` will merge all the individual L5/L7/L8 collections into one:

```
var getCombinedSRcollection = function(year, startDay, endYear,
    endDay) {
```

```

var lt5 = getSRcollection(year, startDay, endYear, endDay,
    'LT05');
var le7 = getSRcollection(year, startDay, endYear, endDay,
    'LE07');
var lc8 = getSRcollection(year, startDay, endYear, endDay,
    'LC08');
var mergedCollection = ee.ImageCollection(le7.merge(lc8)
    .merge(lt5));
return mergedCollection;
};

```

Vegetation Indices: addIndices calculates several vegetation/spectral indices based on the harmonized Landsat bands. In this example, we are including the Normalized Difference Vegetation Index (NDVI), the Enhanced Vegetation Index (EVI), the Soil Adjusted Vegetation Index (SAVI), the Normalized Difference Mangrove Index (NDMI), the Normalized Difference Water index (NDWI), and the Modified Normalized Difference Water Index (MNDWI). Here is where you can include your own vegetation indices:

```

var addIndices = function(image) {
    var ndvi = image.normalizedDifference(['B4', 'B3']).rename(
        'NDVI');
    var evi = image.expression(
        '2.5 * ((NIR - RED) / (NIR + 6 * RED - 7.5 * BLUE + 1))', {
            'NIR': image.select('B4'),
            'RED': image.select('B3'),
            'BLUE': image.select('B1')
        }).rename('EVI');
    var savi = image.expression(
        '((NIR - RED) / (NIR + RED + 0.5) * (0.5 + 1))', {
            'NIR': image.select('B4'),
            'RED': image.select('B3'),
            'BLUE': image.select('B1')
        }).rename('SAVI');
    var ndmi = image.normalizedDifference(['B7', 'B2']).rename(
        'NDMI');
    var ndwi = image.normalizedDifference(['B5', 'B4']).rename(
        'NDWI');
    var mndwi = image.normalizedDifference(['B2', 'B5']).rename(
        'MNDWI');
    return image.addBands(ndvi)
        .addBands(evi)

```

```
.addBands(savi)
.addBands(ndmi)
.addBands(ndwi)
.addBands(mndwi);
};
```

Finally, `collectionSR_wIndex` will include the final harmonized collection with all the sensors based on the temporal parameters defined previously, with all spectral bands and vegetation/spectral indices:

```
var collectionSR_wIndex = getCombinedSRcollection(startYear, startDay,
endyear, endDay).map(addIndices);
```

Filter this collection by the bounds of the area of study:

```
var collectionL5L7L8 = collectionSR_wIndex.filterBounds(area0fstudy);
```

Question 2. Based on your knowledge and the functions described above, what are the main fundamental differences between Landsat TM, ETM+, and OLI?

Question 3. What are the issues that may arise when using the same function for calculating vegetation indices using surface reflectance data acquired by ETM+ and OLI sensors?

Vegetation Index Anomaly

By definition, anomaly is anything that deviates from what is standard, normal, or expected. Usually, anomalies are calculated by subtracting a long-term average of a variable from the actual value of that variable at a given time. For example, if X = actual value of average NDVI for mangroves in 2020, and Y = long-term average NDVI of mangroves (an average over many years), then the anomaly = X - Y. If the anomaly values are zero (or very close to zero), it means that NDVI remained relatively stable in that period, which indicates that there has not been any significant disturbance in that area. On the other hand, a positive anomaly means that the NDVI signal is greater than its long-term average, which indicates that vegetation has shown growth in that area;

similarly, a negative anomaly means that the NDVI signal is weaker than its long-term average, indicating a potential loss in the area.

To calculate the anomaly, start by defining the index you want to compute the anomaly for and the reference period to get the average value. In this example, we are going to use the 16 years before the mangrove extent baseline in 2000:

```
var index = 'NDVI';
var ref_start = '1984-01-01'; // Start of the period
var ref_end = '1999-12-31'; // End of the period
```

Next, create the reference collection using collectionL5L7L8 and the parameters above. You can print the size of this reference collection to the **Console** using `print` and `size`:

```
var reference = collectionL5L7L8
  .filterDate(ref_start, ref_end)
  .select(index)
  .sort('system:time_start', true);
print('Number of images in Reference Collection', reference.size());
```

You can now calculate the mean value (and other statistics) for the reference collection reference. Mask the results by the baseline mangrove extent using extentBuffer:

```
var mean = reference.mean().mask(extentBuffer);
var median = reference.median().mask(extentBuffer);
var max = reference.max().mask(extentBuffer);
var min = reference.min().mask(extentBuffer);
```

Now that we have our long-term reference metrics, you can define the period for which you want to compute the gains and losses. In this example, we will use the full period of 2000–2020. However, any combination of years is possible depending on what period you are interested in. Then, an anomaly function can be created to subtract the metric from the average of your period of interest:

```
var period_start = '2000-01-01'; // Full period
var period_end = '2020-12-31';
```

```

var anomalyfunction = function(image) {
  return image.subtract(mean)
    .set('system:time_start', image.get('system:time_start'));
};

```

Finally, map the `anomalyfunction` to the Landsat collection filtered by your `period_start` and `period_end`:

```

var series = collectionL5L7L8.filterDate(period_start, period_end)
  .map(anomalyfunction);

```

The object `series` will have all the spectral bands and vegetation/spectral indices for the time period defined above. Their values, however, will be different from the original collection since we subtracted the average value of the reference period. The next step is to sum all the values for the index from `series` and divide by the number of images available:

```

var seriesSum = series.select(index).sum().mask(extentBuffer);
var numImages = series.select(index).count().mask(extentBuffer);
var anomaly = seriesSum.divide(numImages);

```

Add the anomaly layer to the map using a color ramp of your choice (Fig. A3.3.2):

```

var visAnon = {
  min: -0.20,
  max: 0.20,
  palette: ['#481567FF', '#482677FF', '#453781FF', '#404788FF',
    '#39568CFF', '#33638DFF', '#2D708EFF', '#287D8EFF',
    '#238A8DFF',
    '#1F968BFF', '#20A387FF', '#29AF7FFF', '#3CBB75FF',
    '#55C667FF',
    '#73D055FF', '#95D840FF', '#B8DE29FF', '#DCE319FF',
    '#FDE725FF'
  ]
};

Map.addLayer(anomaly, visAnon, index + ' anomaly');

```

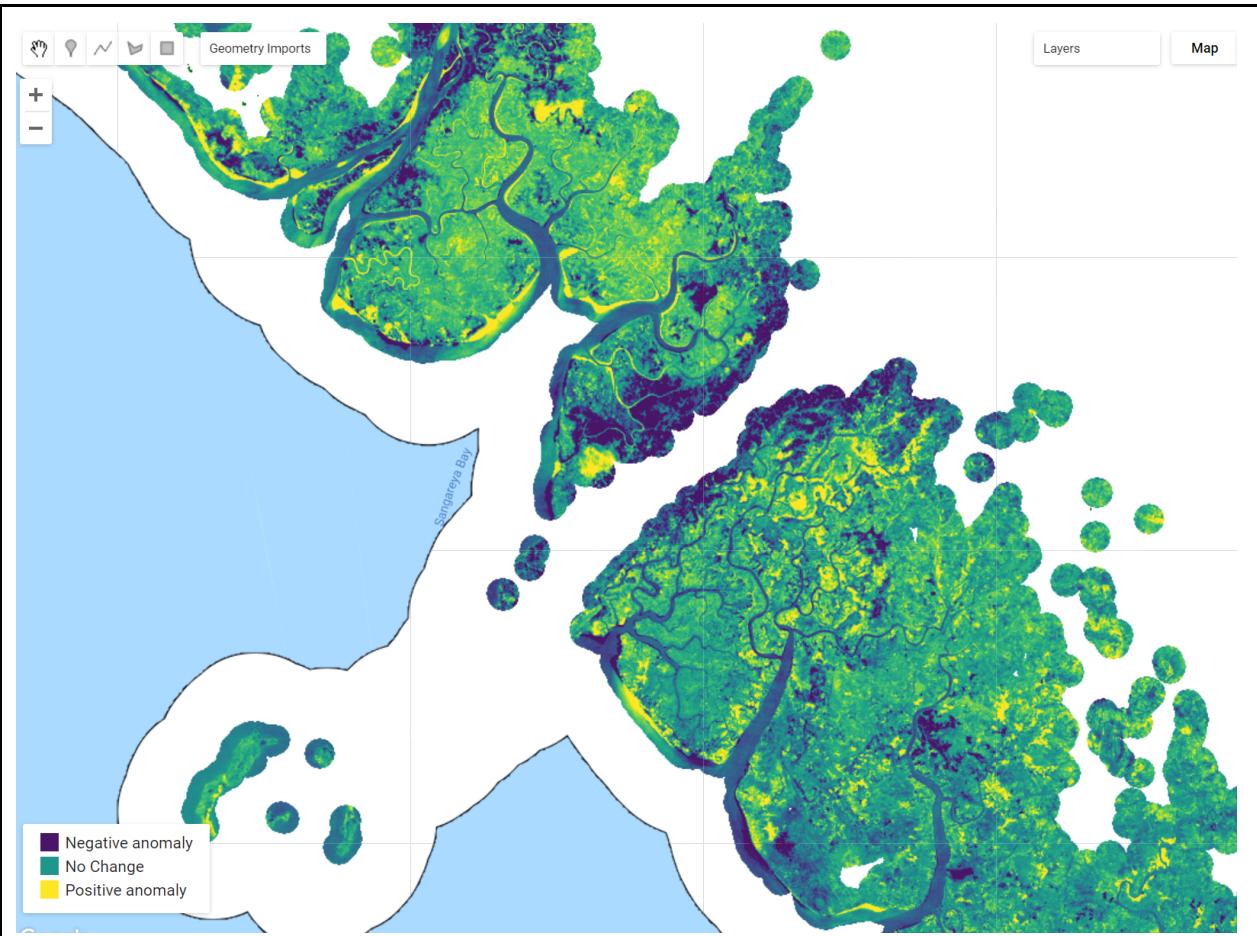


Fig. A3.3.2 NDVI anomaly for the period of 2000–2000

You can then extract loss areas by selecting a value threshold on the anomaly (Fig. A3.3.3):

```

var thresholdLoss = -0.05;
var lossfromndvi = anomaly.lte(thresholdLoss)
  .selfMask()
  .updateMask(
    mangrove2000
); // Only show the losses within the mangrove extent of year 2000

Map.addLayer(lossfromndvi, {
  palette: ['orange']
}, 'Loss from Anomaly 00-20');

var thresholdGain = 0.20;
var gainfromndvi = anomaly.gte(thresholdGain)
  
```

```

    .selfMask()
    .updateMask(
      extentBuffer
    ); // Only show the gains within the mangrove extent buffer of year
2000

Map.addLayer(gainfromndvi, {
  palette: ['blue']
}, 'Gain from Anomaly 00-20');

```

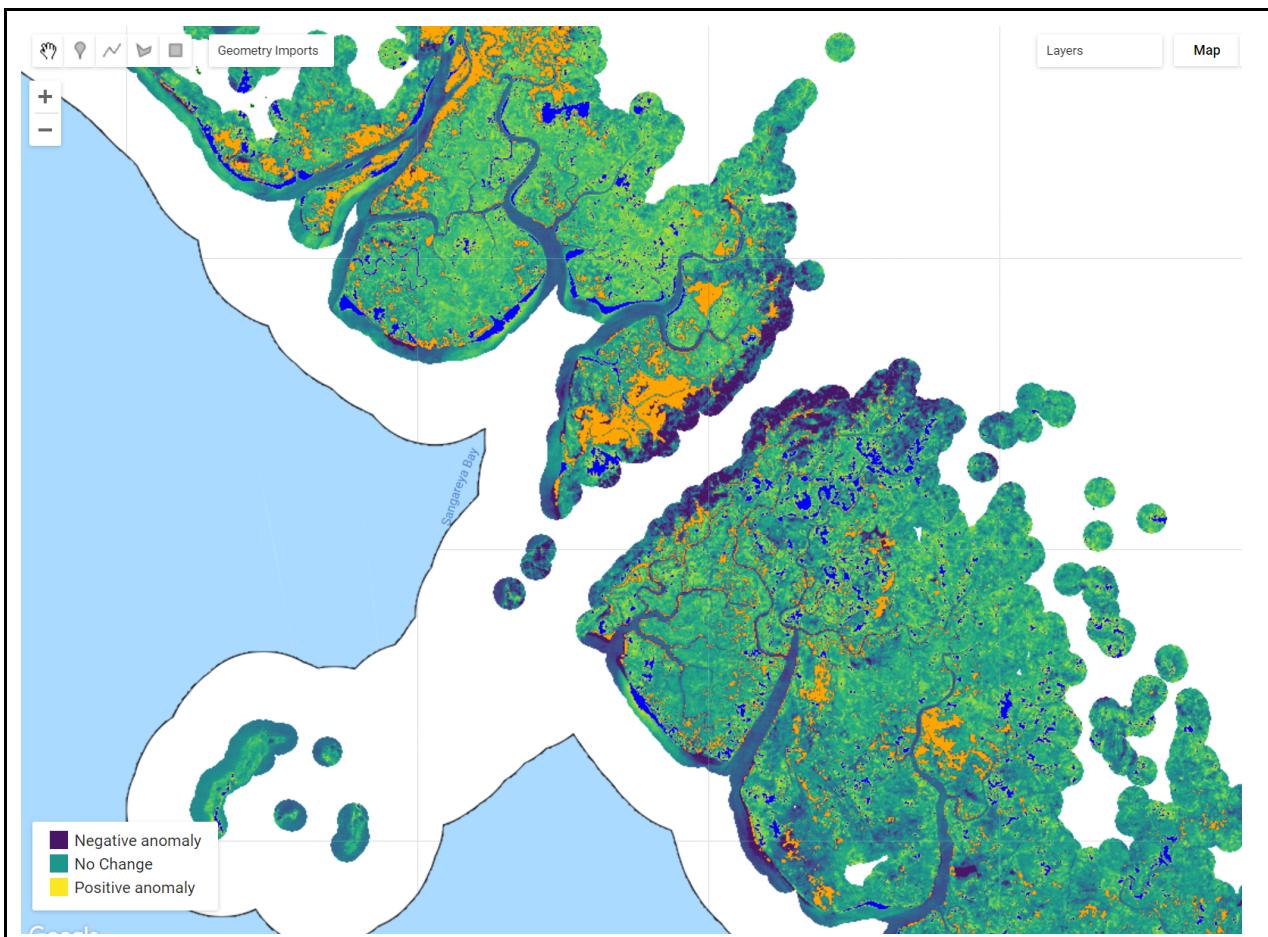


Fig. A3.3.3 NDVI anomaly losses (orange) and gains (blue) based on change threshold values

Code Checkpoint A33b. The book's repository contains a script that shows what your code should look like at this point.

Question 4. What are the main challenges of a map-to-image change detection approach? Think from a perspective of baseline extent, spectral index, and metric used (e.g., mean versus median).

Synthesis

With the content covered in this chapter, you will be able to reproduce a map-to-map and a map-to-image change detection to your own area of interest. Additionally, the anomaly analysis can be used to detect changes in other land cover types, including (but not limited to) forests (forest cover loss) and agricultural land (crop harvest).

Assignment 1. Practice your land cover classification skills by using the code in script **A33s1 - Supplemental** in the book's repository. In this code, we show how to create a mangrove extent map for Guinea using manually and automatically selected samples. The code covers masking techniques and using data from Google Earth Engine Catalog to assist in the classification workflow. Using the techniques you have learned, create two 30 m Landsat-based mangrove extent maps for the year of 2000 and 2020 for Guinea, West Africa.

Assignment 2. Using what you learned in this chapter, compile the areas (in km²) of mangrove change in Guinea derived from the anomaly analysis using: (a) vegetation spectral index versus a water-based spectral index; (b) mean versus median; and (c) five-year intervals (2000–2005, 2005–2010, 2010–2015, and 2015–2020).

Assignment 3. Practice the classification and compare the map-to-map and map-to-image change approaches for another land cover type/area of your choice, such as the following:

- Agricultural land expansion in the Nile Delta, Egypt.
- Forest burn/loss near Mount Hood, Oregon, United States.

Conclusion

Google Earth Engine's computing infrastructure has been revolutionizing time-consuming remote sensing processes, creating a new way for rapid land cover classification and change detection at large scales. In the case of mangrove forests—a highly dynamic ecosystem that has been under increasing anthropogenic pressure—these approaches allow for a rapid and consistent monitoring of change in extent over time. These approaches using Earth Engine may be particularly useful for developing countries where, until very recently, the high computational power and the difficulties of distributing nontrivial classification algorithms across multiple computational workstations has challenged classification and change detection at large scales.

Feedback

To review this chapter and make suggestions or note any problems, please go now to bit.ly/EEFA-review. You can find summary statistics from past reviews at bit.ly/EEFA-reviews-stats.

References

Bunting P, Rosenqvist A, Lucas RM, et al (2018) The global mangrove watch - A new 2010 global baseline of mangrove extent. *Remote Sens* 10:1669. <https://doi.org/10.3390/rs10101669>

Goldberg L, Lagomasino D, Thomas N, Fatoyinbo T (2020) Global declines in human-driven mangrove loss. *Glob Chang Biol* 26:5844–5855. <https://doi.org/10.1111/gcb.15275>

Hansen MC, Potapov PV, Moore R, et al (2013) High-resolution global maps of 21st-century forest cover change. *Science* 342:850–853. <https://doi.org/science.1244693>

Roy DP, Kovalskyy V, Zhang HK, et al (2016) Characterization of Landsat-7 to Landsat-8 reflective wavelength and normalized difference vegetation index continuity. *Remote Sens Environ* 185:57–70. <https://doi.org/10.1016/j.rse.2015.12.024>