Module 2.2 Continuous Degradation Detection (CODED)

Eric Bullock

Version 1

Background

Forest Degradation

There is no internationally accepted definition of forest degradation, but it can generally be defined as a disturbance in a forest that reduces biomass, ecological productivity, and/or canopy cover but does not result in a land cover conversion.

In the context of national greenhouse gas inventories, forest degradation represents a source of carbon loss in a similar manner to deforestation. Examples of activities that cause forest degradation include selective logging, non-natural forest fires, and fuelwood collection for household use.

While deforestation has been the subject of intense debate on a global scale, forest degradation also contributes significantly to greenhouse gas emissions and reduces the ability of forests to provide essential products and services and mitigate climate change. However, monitoring forest degradation at the national to regional scale has proven to be challenging. Consequently, it is frequently omitted from national greenhouse gas inventories and represents a significantly underestimated source of carbon emissions and ecological damage.

This tutorial will introduce a methodology for monitoring forest degradation called Continuous Degradation Detection (CODED). CODED was created for the specific goal of detecting forest degradation which, as will be described in the following sections, has proven challenging using traditional approaches to change detection. Application of CODED is performed on Google Earth Engine and demonstrated for case studies in Colombia, Mozambique, and Cambodia.

Learning Objectives

At the end of this module, you will be able to:

- Identify common drivers of forest degradation in moderate-to-high resolution satellite imagery.
- Perform spectral mixture analysis with optical imagery and calculate the Normalized Difference Fraction Index (NDFI).

- Parameterize CODED to target different disturbance dynamics, forest types, and date availability.
- Use CODED to create maps of degradation and deforestation.

Advanced usage will also include:

- Post-processing of results from CODED to increase change detection accuracy.
- Application of a modified version of CODED to use Sentinel-1 radar imagery.

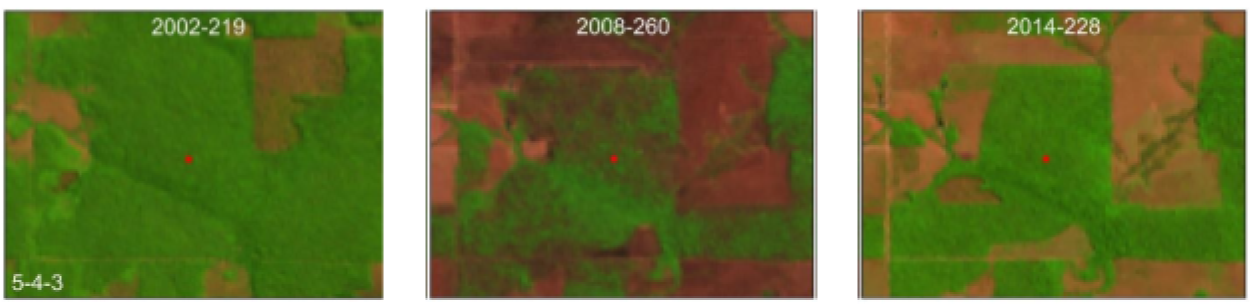
Continuous Degradation Detection (CODED)

Motivation

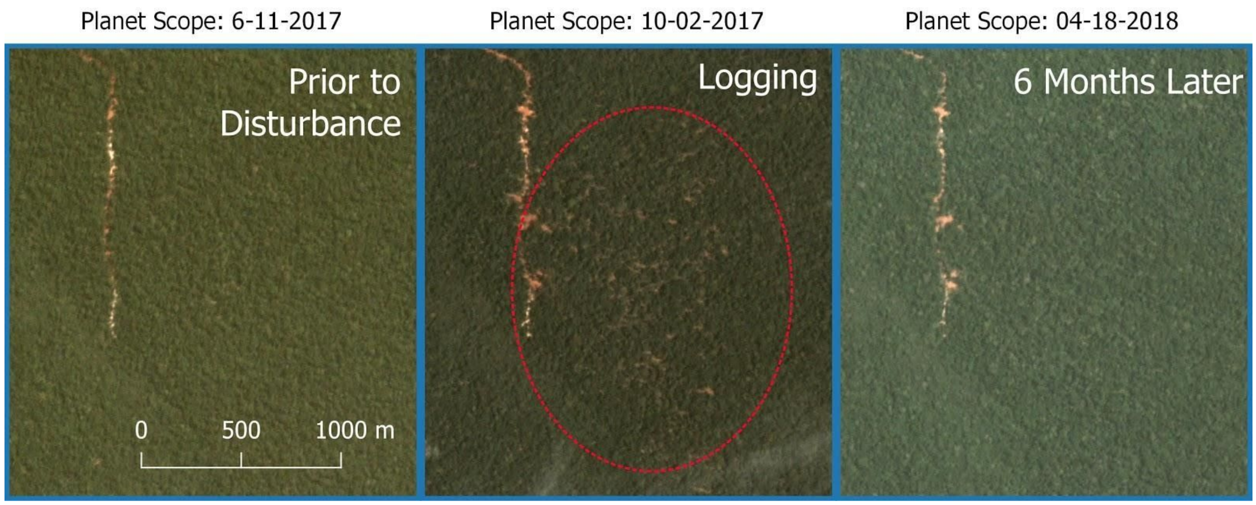
In the absence of robust field inventory, the only feasible way to do so is to use earth observation imagery, but the form degradation occurs on the landscape brings technical considerations that must be addressed. To start, any automated approach to classification of satellite imagery will inevitably contain errors. These errors can come from many factors, including unmasked clouds and cloud shadows, missing data (for example, due to the scan line corrector issue of Landsat 7), and classification errors (for example mis-classification of land cover or change classes). We also know that the humid tropics are especially susceptible to missing or contaminated data due to clouds.

While this issue is not unique to mapping forest degradation, it is important to consider this context when attempting to automate an approach for large-area monitoring. Below, we can see a few years' worth of Landsat imagery in a logged forest in the Central African Republic. The goal here is to extract meaningful information about changes in the forest, despite nearly every image being partially or entirely covered by clouds or missing data. Any effective approach to mapping disturbances in this landscape must accommodate noisy observations while also being sensitive to subtle changes in the forest canopy.

There are numerous approaches to dealing with noisy observations, such as filtering, compositing, and atmospheric correction. However, approaches based on filtering or smoothing of input data are not well suited for detecting degradation, since degradation is often visible for a short amount of time and over small areas. Note the example below of a fire in the Brazilian Amazon. The pixel highlighted in red is in a forest that burned around 2008. Signs of this fire are not visible much after the event due to regeneration.



Looking again at the example of Central Africa Republic, we can further see how ephemeral damage from degradation can be. Here, we can see signs of logging in high resolution imagery from Planet. However, just a few months later the forest has regenerated. The ephemeral nature of degradation implies that continuous monitoring is necessary rather than single-date analysis or compositing.



Furthermore, forest degradation often occurs at a spatial scale below the nominal scale of satellite imagery, meaning that spatial smoothing for the sake of noise reduction can further mix stable and disturbed forest signals. For example, below is an example of selective logging in Fiji. While there are obvious logging roads visible in the high resolution imagery, the majority of the landscape remains forest in the moderate resolution imagery.

These examples demonstrate how the very nature of forest degradation represents a significant challenge for mapping using traditional approaches to remote sensing analysis.

Land Change Monitoring

In recent years, the land change monitoring community has increasingly adopted the notion that time series analysis can alleviate many of the issues that arise in single-date analysis. Time series approaches are able to monitor trends through time, which helps to distinguish subtle land changes from noisy and characterize both abrupt and gradual changes. While change monitoring has a long history in fields such as econometrics, signal processing, pattern recognition, and environmental modeling and forecasting, application in the remote sensing domain is relatively new.

Time series approaches to change monitoring have greatly benefited from a few developments in the remote sensing communities, including:

- Free and open data access from agencies such as the USGS, NASA, ESA, and JAXA.
- Consistent and repeated data archives spanning multiple years or decades.
- High performance cloud computing environments such as Google Earth Engine and Amazon Web Services.
- Novel change detection methodologies based on dense time series data.
- Standardized pre-processing in support of data collections.

Notable early applications of time series analysis for land change monitoring include LandTrendr (Kennedy et al ####), BFAST (Verbesselt et al. ####), Vegetation Change Tracker (Huang et al., 2010), Exponential Weighted Moving Average (Brooks et al., ####), and Continuous Change Detection and Classification (Zhu and Woodcock, 2014). These approaches have their unique advantages and helped to lay the groundwork for the CODED.

Algorithm Overview

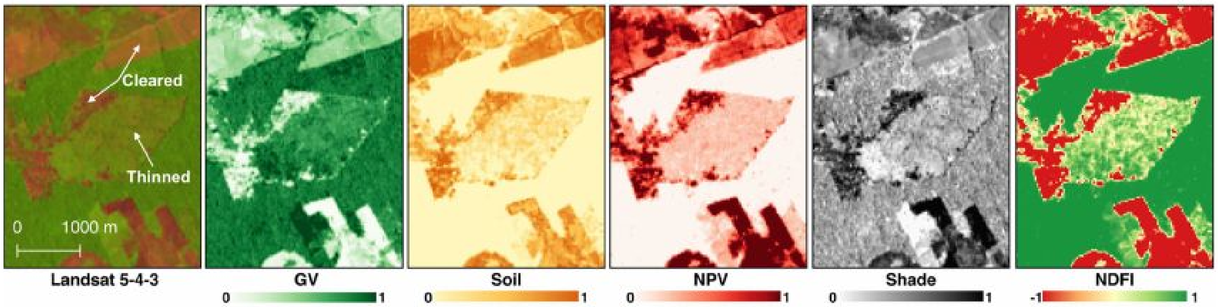
The CODED methodology has three primary components:

1. Pre-processing of optical imagery to mask clouds and transform reflectance into spectral endmember fraction images.
2. Change detection using regression-based break detection.
3. Change attribution into deforestation or degradation using and training data a machine learning classifier.

Spectral Mixture Analysis

Pre-processing of input data is transformed into fractional images using spectral mixture analysis (SMA). Spectral unmixing has been shown in a variety of environments to be sensitive to sub-pixel land changes, making it ideal for the detection of small scale degradation events. Application of SMA for mapping degradation was demonstrated in the Brazilian Amazon in the work of Dr. Carlos Souza Jr.. Souza et al., (2004,5) which introduced the mixture model used by default in CODED.

The SMA model used in CODED is used to transform reflectance into the pixel-scale proportion of green vegetation (GV), soil, non-photosynthetic vegetation (NPV), shade, and cloud. These layers can then be used to calculate the Normalized Difference Fraction Index (NDFI), which is a fractional-image transform first introduced in Souza et al. (####) and demonstrated to be sensitive to sub-pixel damage from logging or fires.



Try it yourself

Before diving into the more complex usage of CODED, it is worth looking at the fractional image results from SMA on individual images. The following Javascript code can be used in Google Earth Engine. The full code for SMA can be found in the Open-MRV repo in the file 'TrainingData/Unmix_L8'.

First, load a Landsat 8 image for a study region in Colombia and add it to the map. The study area is defined as a feature collection in the 'studyArea' variable.

```
var studyArea = ee.FeatureCollection('users/ramblingrek/ColombiaRectangle')

Map.addLayer(studyArea)
Map.centerObject(studyArea)
```

Next we can load the Landsat 8 collection and apply the cloud mask using a defined function 'maskL8sr'.

```
var landsat8 = ee.ImageCollection("LANDSAT/LC08/C01/T1_SR").filterBounds(studyArea)


function maskL8sr(image) {
  var bandList = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7']
  var nameList = ['BLUE', 'GREEN', 'RED', 'NIR', 'SWIR1', 'SWIR2']

  var cloudShadowBitMask = 1 << 3;
  var cloudsBitMask = 1 << 5;
  var qa = image.select('pixel_qa');
  var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
          .and(qa.bitwiseAnd(cloudsBitMask).eq(0));
```

```
  return image.updateMask(mask).divide(10000)
            .select(bandList).rename(nameList)
            .copyProperties(image, ["system:time_start",'CLOUD_COVER']);
}

var l8Masked = landsat8.map(maskL8sr)
```

To find a relatively cloud-free image to apply SMA, we can sort the collection by the 'CLOUD_COVER' metadata property and use the first image in the collection (i.e. the image with the lowest value for 'CLOUD_COVER'.

```
var l8Image = l8Masked.sort('CLOUD_COVER',true).first()

Map.addLayer(l8Image.select(['RED','GREEN','BLUE']), {min: 0, max: .14}, 'RGB Image')
```

Note that the image only partially overlaps our study area. For this exercise that is okay, because the purpose is just to look at any single fraction image. To do so, we can calculate fraction images using the spectral endmembers defined in Souza et al., 2005 and add each fraction image to the map.

```
// Define the spectral endmembers in units of reflectance
var endmembers = {
  gv: [.0500, .0900, .0400, .6100, .3000, .1000],
  shade: [0, 0, 0, 0, 0, 0],
  npv: [.1400, .1700, .2200, .3000, .5500, .3000],
  soil: [.2000, .3000, .3400, .5800, .6000, .5800],
  cloud: [.9000, .9600, .8000, .7800, .7200, .6500],
}

// Define color palettes for visualization

var palettes = {
  gv: ['#ffffcc','#c2e699','#78c679','#31a354','#006837'],
  shade: ['#f7f7f7','#cccccc','#969696','#636363','#252525'],
  npv: ['#ffffb2','#fecc5c','#fd8d3c','#f03b20','#bd0026'],
  soil: ['#ffffd4','#fed98e','#fe9929','#d95f0e','#993404'],
  cloud: ['#ffffcc','#a1dab4','#41b6c4','#2c7fb8','#253494'],
  ndfi: ['#d73027','#fc8d59','#fee08b','#ffffbf','#d9ef8b','#91cf60','#1a9850']
}

// Perform unmixing on Landsat 8 image.
var unmixedImage = ee.Image(l8Image).unmix({
  endmembers: [endmembers.gv, endmembers.shade, endmembers.npv, endmembers.soil,
endmembers.cloud],
  sumToOne: true,
  nonNegative: true
```

```
}).rename(['GV', 'Shade', 'NPV','Soil','Cloud'])

// Add each fractional image to the map
Map.addLayer(unmixedImage.select('GV'), {min: 0, max: .6, palette: palettes.gv}, 'GV')
Map.addLayer(unmixedImage.select('Shade'), {min: 0, max: .6, palette: palettes.shade}, 'Shade')
Map.addLayer(unmixedImage.select('NPV'), {min: 0, max: .6, palette: palettes.npv}, 'NPV')
Map.addLayer(unmixedImage.select('Soil'), {min: 0, max: .6, palette: palettes.soil}, 'Soil')
```

Finally, we can calculate NDFI on the endmember fraction images and add the NDFI data to the map.

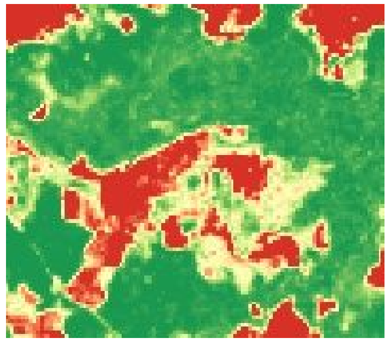
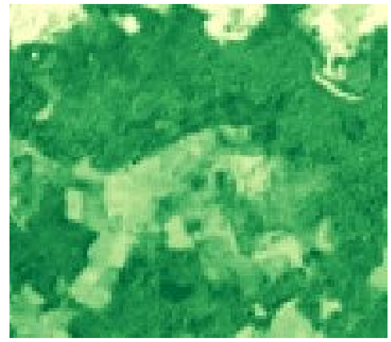
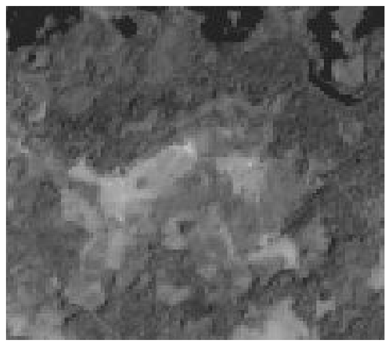
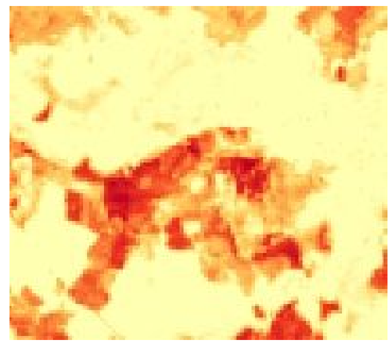
```
var ndfi = unmixedImage.expression(
  '((GV / (1 - SHADE)) - (NPV + SOIL)) / ((GV / (1 - SHADE)) + NPV + SOIL)',
  {'GV': unmixedImage.select('GV'),
  'SHADE': unmixedImage.select('Shade'),
  'NPV': unmixedImage.select('NPV'),
  'SOIL': unmixedImage.select('Soil')}
)

Map.addLayer(ndfi, {min: 0, max: 1, palette: palettes.ndfi}, 'NDFI')
```

The example below is a zoom-in the results that can be navigated to by defining the image boundary as a geometry:

```
var geo = ee.Geometry.Polygon(
        [[[-73.84809238658222, 2.093131559229626],
        [-73.84809238658222, 2.047928319636153],
        [-73.8027737830666, 2.047928319636153],
        [-73.8027737830666, 2.093131559229626]]])

Map.centerObject(geo)
```

Note the disturbed patch of land in the middle of the geometry. The clearly non-forest pixels have a high NDFI (red), while the damaged forest pixels on the right have a moderate NDFI (yellow), and dense forest has a high NDFI (green). These examples are labeled in red, yellow, and green on the Google Earth Image below the SMA images.

RGB	NDFI	GV

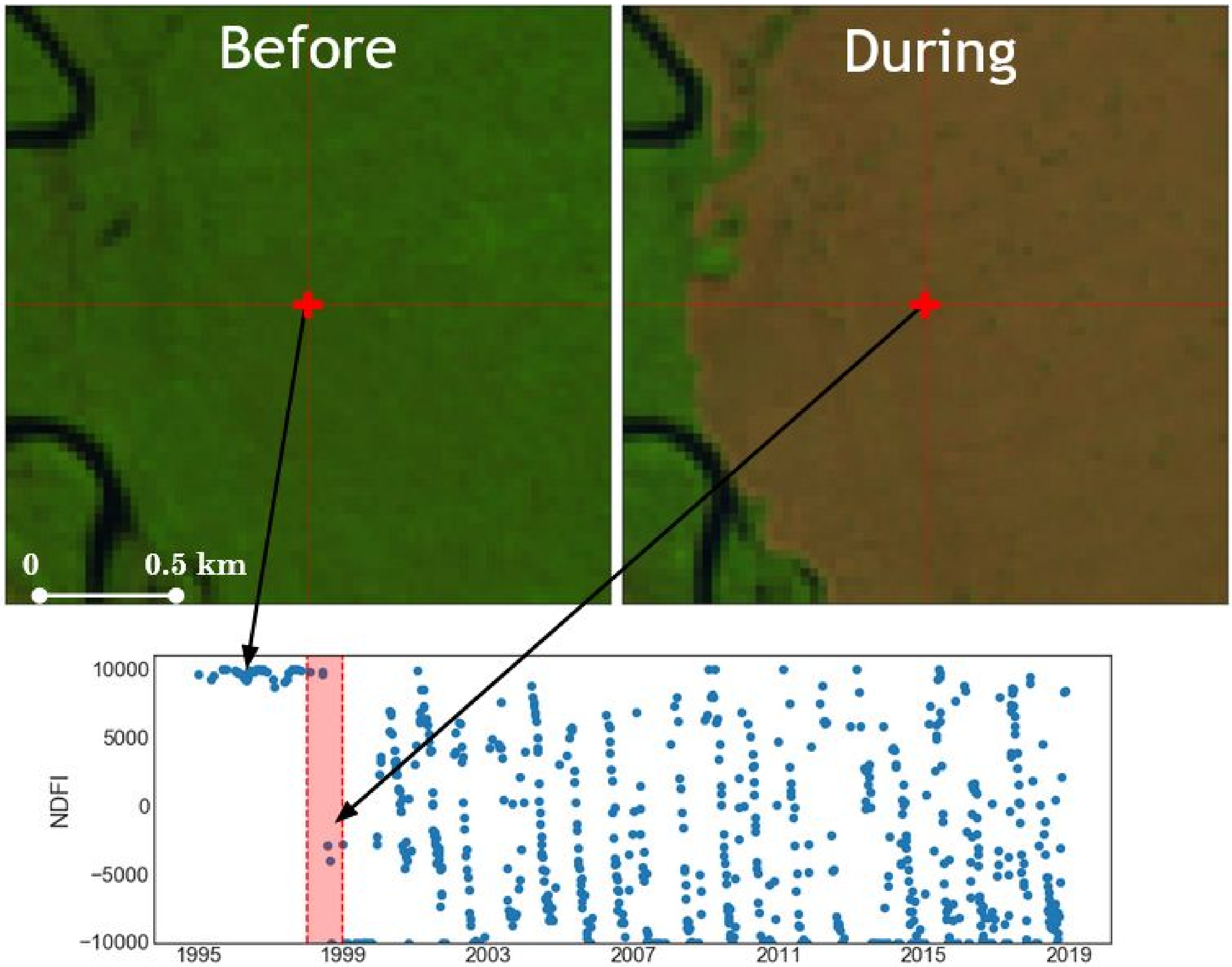
		
Soil 	Shade 	NPV 

Google Earth image with labels:

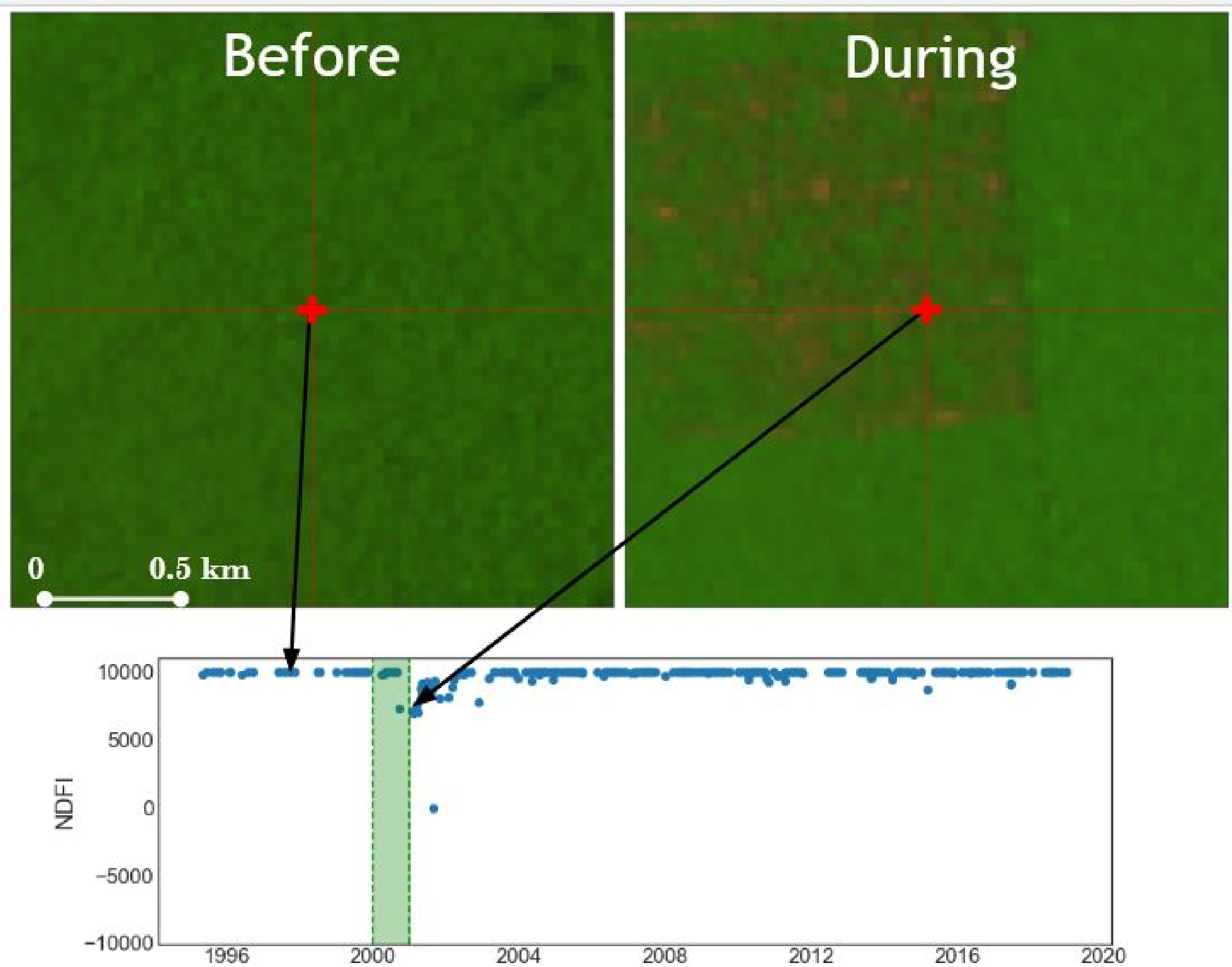
Temporal trajectories

CODED performs SMA for every available Landsat image in the study region. It then performs change detection using the time series NDFI trajectories. Let’s look at a few examples of what the time series of NDFI looks like for degradation and deforestation.

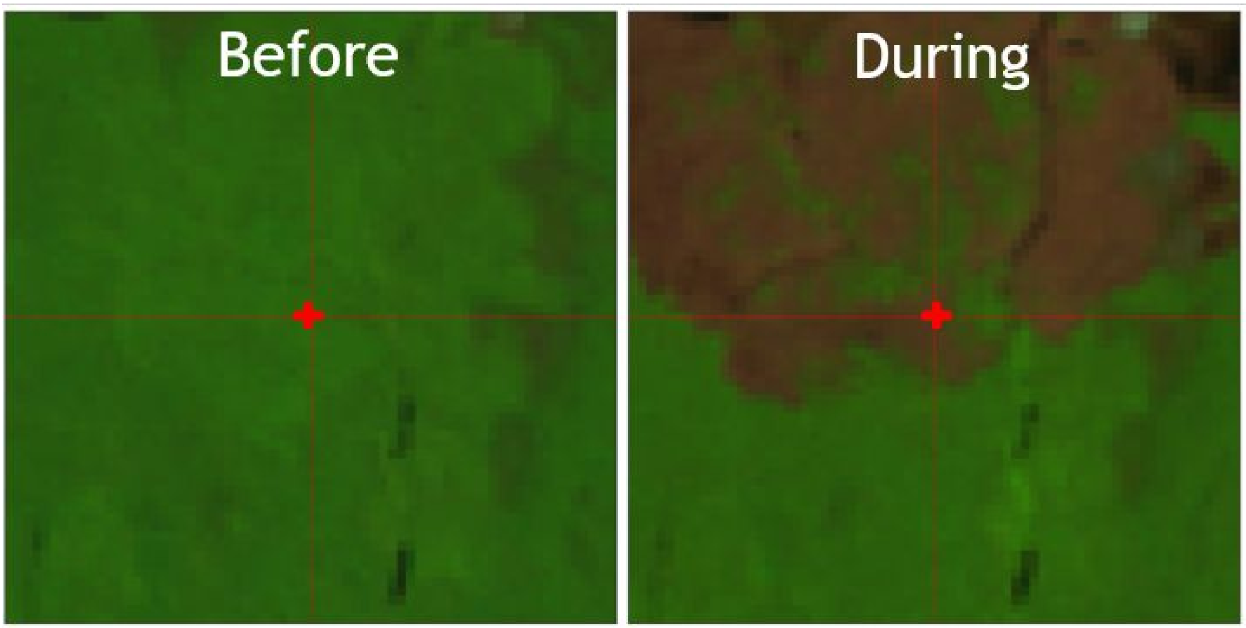
NDFI is high (~1) for dense forests, lower (0-1) for open or degraded forests, and lowest (-1 - 0) for non-vegetative landscapes. The following figure shows an NDFI trajectory for a deforestation event. In each of the following examples, the red and green vertical lines on the plot represent the year of the deforestation or degradation event, respectively. Note that before the deforestation event, NDFI is around 1 because it is a dense and stable forest. After the forest is converted to a pasture, NDFI drops suddenly and the oscillates between -1 and 1 seasonally.

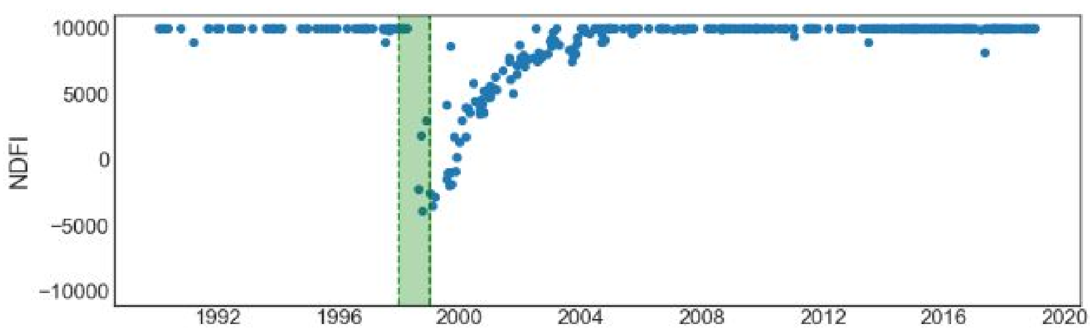


Now contrast this with an example of degradation due to selective logging. Note that the change in NDFI due to the disturbance is relatively small, and there is a clear recovery signal due to forest regeneration.



Finally, we see in the following figure an example of a forest affected by fire. Note that here the canopy damage was rather dramatic, as evident in the large reduction in NDFI after the disturbance. However, there is once again a clear regeneration signal and it is clear from the time series that there was no land cover conversion.





Try it yourself

It is relatively straightforward to plot the temporal trajectories of NDFI on Google Earth Engine. This can be a useful exercise to understand the land use history of a location in addition to the spectral response to landscape changes. To do so, it is necessary to first calculate NDFI for an entire image collection, and then define a function for plotting NDFI at location selected on the map.

```
// Define a function for SMA and calculate or NDFI
var unmixAndNDFI = function(image) {
  var unmixedImage = ee.Image(image).unmix({
        endmembers: [endmembers.gv, endmembers.shade, endmembers.npv, endmembers.soil,
endmembers.cloud],
        sumToOne: true,
        nonNegative: true
  }).rename(['GV', 'Shade', 'NPV','Soil','Cloud'])
```

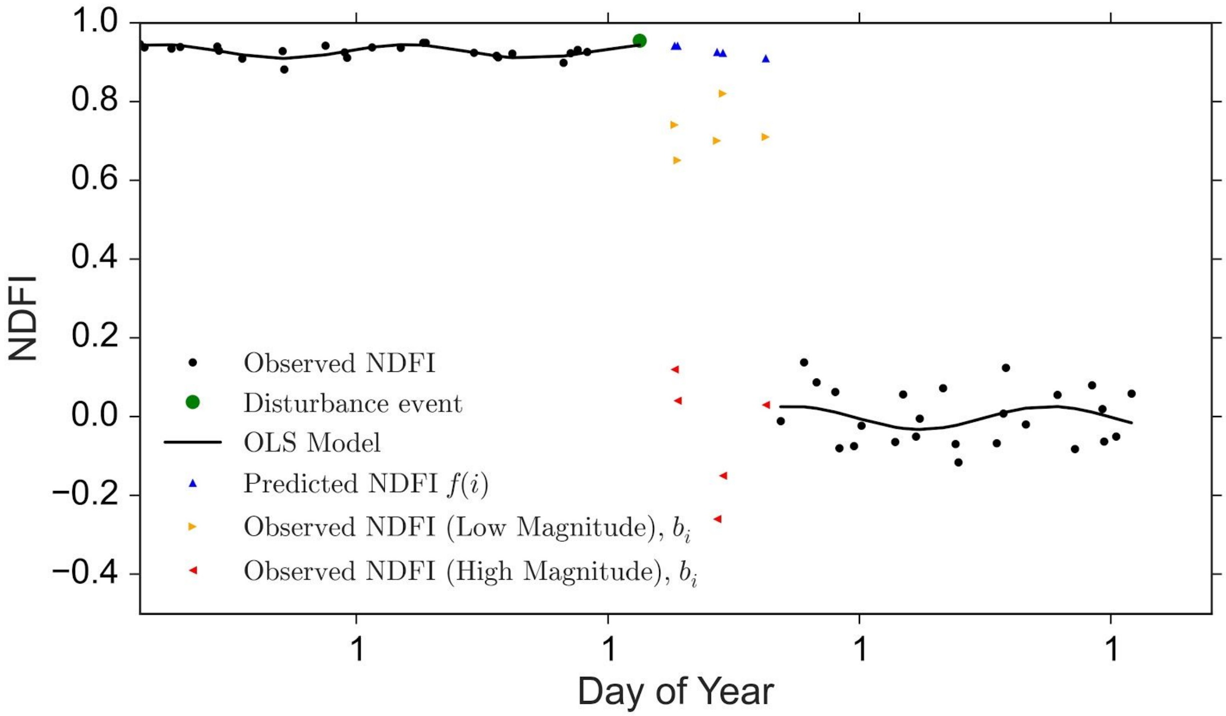
```
  var ndfi = unmixedImage.expression(
        '((GV / (1 - SHADE)) - (NPV + SOIL)) / ((GV / (1 - SHADE)) + NPV + SOIL)',
        {'GV': unmixedImage.select('GV'),
        'SHADE': unmixedImage.select('Shade'),
        'NPV': unmixedImage.select('NPV'),
        'SOIL': unmixedImage.select('Soil')}
  ).rename('NDFI')
  return image.addBands([unmixedImage, ndfi])
}

// Map over the Landsat 8 collection and calculate NDFI
var ndfiCollection = l8Masked.map(unmixAndNDFI).select('NDFI')
```

The following code demonstrates how you can create a geometry from the location clicked on the map (“var point”), and plot the NDFI time series (“var chart”) for all data intersecting that location. The plots have a callback function that will load the corresponding image to the map when an observation is selected on the plot.

```
// Function to make an NDFI plot at the clicked location
var makeImagePlot = function(col, region){
  var chart = ui.Chart.image.series(col, region, ee.Reducer.mean(), 30)
        .setOptions({
        lineWidth: 0,
        pointSize: 6,
        });

  // Callback function to add clicked image to map.
  chart.onClick(function(x, y, sName) {
        if(x) {
        var im = l8Masked.filterMetadata('system:time_start','equals',x)
        Map.addLayer(im.select(['RED','GREEN','BLUE']), {min: 0, max: .14}, 'Image ' + x)
        }

  });
  print(chart)
}

// Callback function when clicking on the map.
Map.onClick(function(coords) {
  var point = ee.Geometry.Point([coords.lon, coords.lat])
  makeImagePlot(ndfiCollection, point)
  Map.addLayer(point)
})
```

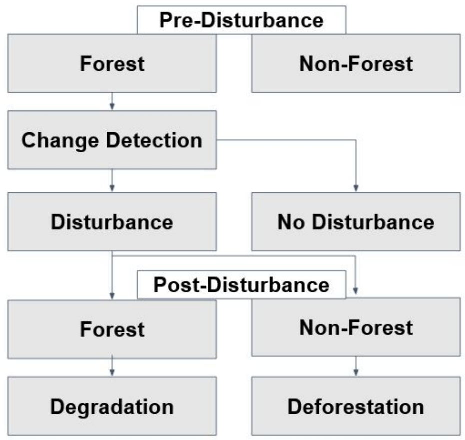
Change Detection

The regression models are used to predict observations within a moving window. If the residuals of the observations in the window exceed a critical value, then a change is detected. Note below the triangles are the observations in the moving window. The blue triangles are the predicted observations, while the yellow and red represent low and high magnitude changes, respectively. The process then repeats with a new regression fit to subsequent observations.



Change Attribution

Disturbances are attributed as deforestation or degradation based on the land cover after the disturbance. If there is a conversion from forest to non-forest (e.g. pasture, settlements, or agriculture), the disturbance is considered deforestation. If regeneration begins after the disturbance and there is no land cover conversion then the disturbance is labeled degradation. This attribution process can generally be described according the following flowchart:



A schematic diagram of the change attribution process.

Running CODED with a Graphical User Interface

CODED can be run using a graphic user interface (GUI) named the 'Forest Change Mapping GUI', which can be found in the [CODED GEE repository](). This section provides an overview of the application and its functionality, and the following section demonstrates examples of using the application.

After adding the repo, you should see a "coded" folder in your "Scripts" panel under "Reader":

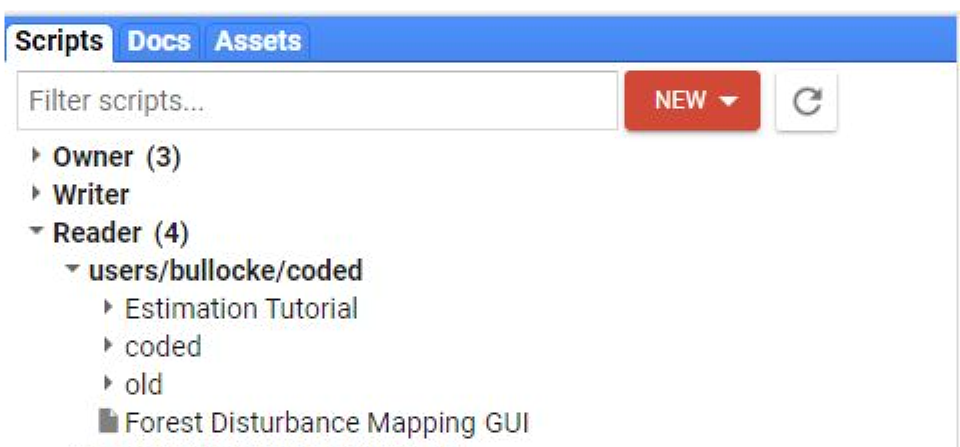
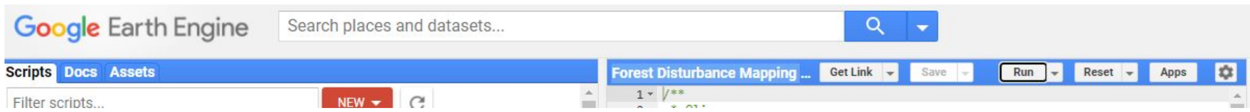


Figure 1 is an image of what your screen should look like. If you do not see this, navigate to the scripts tab in the upper left hand corner of the GEE interface. Expand the Reader tab, then click on the script called **Forest Disturbance Mapping GUI**.

You may then need to Click on Run to initiate the script to load in the browser. Note, it may take a few moments for the GUI to appear in the browser.



Then it will prompt you to choose your language. After the language has been set, your browser should look similar to the following Figure, however B, D, and E will not display until after running CODED.

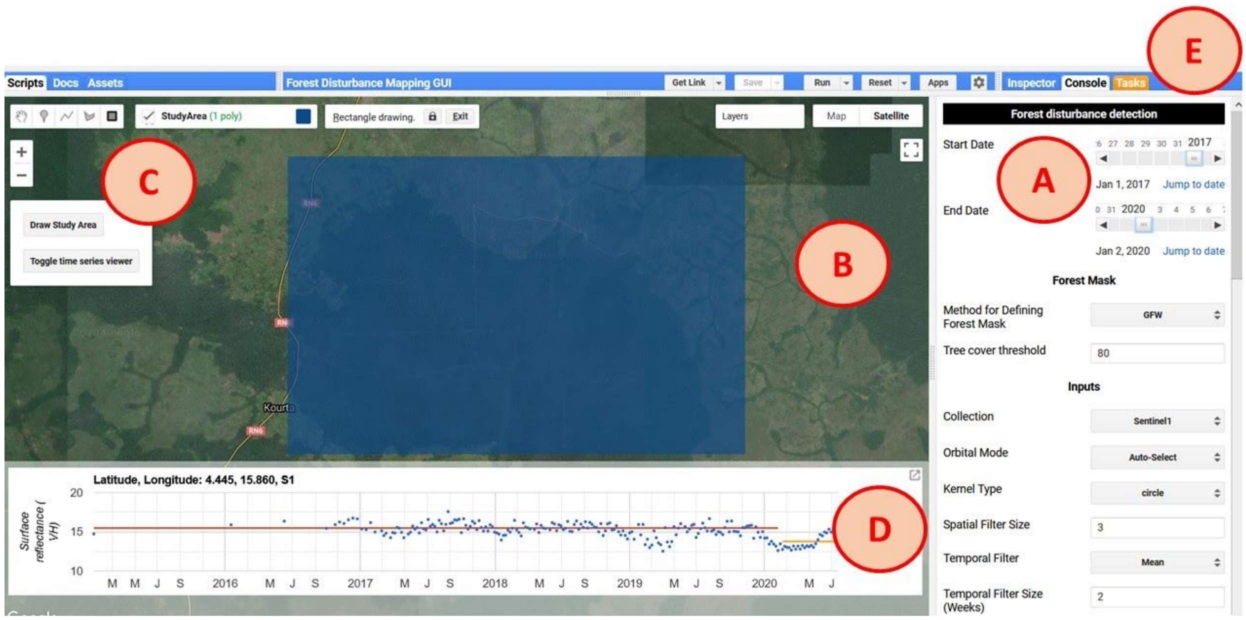


Figure 1. The Forest Change Mapping GUI interface. The components are labeled according to their descriptions below.

As can be seen in Figure 1, the application contains various components that are labeled according to the letters in parenthesis below:

Parameter panel (A)

- On the right of the screen is the parameter panel. This panel contains widgets that control all aspects of the mapping process including the input data, pre-processing, change detection parameters, post-processing, visualization, and exporting.

Map panel (B)

- The map displays the results of the change detection and can be used interactively to define the study area using the map controls and display historical trajectories of clicked locations.

Map controls (C)

- The map controls contain two buttons: ‘Draw Study Area’ and ‘Toggle time series viewer’. After clicking the first button, a geometry import is created named ‘StudyArea’. Use this box to define the study area of your analysis. The study area can be reset by clicking the button. The second button is used to display or remove the time series panel from the map.

Time series panel (D)

- The time series panel displays the temporal trajectory of the collection defined in the parameter panel. The band that is visualized can be defined under the visualization parameters. Each point in the scatter plot represents one observation from the location clicked on the map. When clicking on a point on the scatter plot, the corresponding image will be loaded to the map. Band combination and stretch options for the displayed image can be changed in the visualization section of the parameter panel.

Task tab (E)

- A default feature of the GEE web interface, the task tab is used to submit jobs for processing. Tasks can be used to export images and feature collections as assets or to external storage such as Google Drive.

Parameterization

Forest Mask

A forest mask can be used that determines the valid pixels used for mapping forest disturbances. There are currently two approaches to defining a forest/non-forest (FNF) mask, or you can choose not to use a mask and assume all pixels in the study region can potentially contain a disturbance.

Asset

- The first option allows you to use an existing image asset as a forest/non-forest mask Specify the path to the image in the textbox that appears after selecting ‘Asset’ in the ‘Forest Mask’ dropdown widget.
- The FNF mask asset must have values of 1 indicating valid (forest) pixels, and zero or null indicating invalid (non-forest) pixels.
- By default, this option uses the image geometry as the study region. The study area can be manually overridden using the study area selection widget.

Global Forest Watch (GFW)

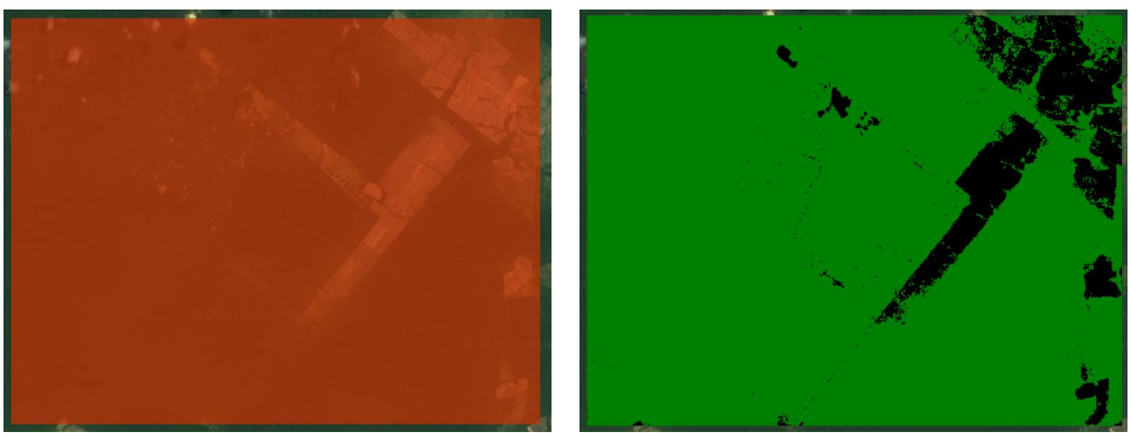
- The second option uses the 2000 tree cover percentage layer of the Hansen et al (2013)/Global Forest Watch dataset. This option requires you to manually define the study area (next section).

- The GFW layer is in units of percent tree cover for the year 2000. After selecting 'GFW' in the dropdown menu, specify the minimum tree cover to be labeled as forest. All pixels not meeting this threshold are assigned to the non-forest class.
- There is an option to 'Mask Prior to Start Date'. This masking is performed using the GFW tree cover loss and year layers. This option will convert all changes in the GFW layer prior to your start date to non-forest.

Defining a Study Area and Time Period

- The study area is set by pressing the button with the text 'Draw Study Area' on the left side of the map (Figure 1). Once pressed, a geometry titled 'StudyArea' appears in the Geometry Imports. Use the map to draw a bounding box, which will be the extent of the analysis.

Tip: Using too big of a study area might cause the computation to time out. If that happens, try exporting the results by submitting a task.



An example study area as defined by the geometry import layer 'StudyArea' (left) and the corresponding FNF mask using the GFW option with a tree cover threshold of 80% (right).

- In the Parameter panel (Figure 1, A), you can specify the start and end dates for the analysis. These dates will filter the data used for change detection, and therefore define the change monitoring period. The 'Jump to date' link allows for date selection on a calendar.

Inputs

- Currently, the application provides support for using either Landsat, Sentinel 2, or Sentinel-1 data as the basis for change detection and attribution. The appropriate inputs to use for your study depends on the specifics of your study and the data available in your study area. Generally speaking, Landsat will be more sensitive to subtle disturbances in areas with sufficient data but is subject to masked or faulty data due to clouds.
- Each input collection can be filtered by day of year (DOY). For example, a 'Start DOY' of 152 and 'End DOY' of 244 will use all imagery between June 1 and September 1 and within the years of the study time period.
- The collections can also be temporally "smoothed" using an ee.Reducer. For example, a 2 week mean temporal filter would create 14-day composites by calculating the mean for each pixel for all data within the 14-day period.

Change Detection

CODED can be modified using three parameters:

Parameter	Description
chiSquareProbability	The chi-square probability of detecting a change [0-1].
Consecutive Obs	# of consecutive observations beyond change probability to flag a change
Change Band	Band to use in change detection test. Currently, only NDFI is supported.

Change Attribution

Sampling

- With this method, the FNF mask is sampled and used as training data to classify the time segment after the first disturbance.
- If the time segment after the disturbance is forest, the disturbance is labeled as degradation. If it is non-forest, then that is evidence of a land cover conversion and the disturbance is labeled as deforestation. If the segment cannot be classified due to insufficient data then it is labeled as an unknown disturbance.
- There are only two parameters that need to be specified for sampling. The first is the number of training points to allocate to each class (forest and non-forest), and the second is the year to train the classifier on. The year must be within the study period. Pixels that undergo a change will not be used for training.

Training data

- With this method, the users can manually specify training data to classify the post-disturbance land cover.
- This option requires the user to specify a path to the training data as a ee.FeatureCollection of points with land cover labels. There can be an unlimited number of classes, but exactly one must be forest.
- The training year represents the year that the training locations correspond to the associated land cover label.

Post-Processing

- Post-processing of the map results is a good way to remove spurious changes due to noise (such as unmasked clouds) or real changes that are not due to disturbance.
- Currently, the only post-processing step implemented is the use of a threshold on change magnitude. Change magnitude relates to change in data space. For example, a disturbance that causes a large change in NDFI will have a larger change magnitude than a subtle change that results in a minor change in NDFI. The units of change magnitude are the residuals during the model change window normalized by the model root-mean-squared error.
- The units of change magnitude can be hard to wrap your head around, but in general a larger threshold will remove more changes, and a threshold of 0 will remove no changes.

Visualization

Add Forest Mask Layer

- Add the forest/non-forest mask layer to the map. Forest pixels will show in green, while non-forest will be black.

Add Break Date

- Add the date of the first detected change as a layer to the map.

Add Stratification

- Add stratification to the map.

Center Zoom

- Center the map zoom on the study area.

Mask Results

- Apply the forest/non-forest mask to the results. All areas labeled as non-forest will be masked in the results.

Reset Map

- Reset the map layers.

Verbose

- Print messages to the console.

Asset path

- Path to an Earth Engine Table or Image that can be loaded to the map for reference.

Load asset

- Load the asset specified in the ‘Asset path’ widget to the map.

Exporting

Raw (Change) Output

- This represents the raw outputs exactly how they appear from the change detection algorithm (e.g. CCDC). No post-processing is performed nor is any sort of land cover classification or change attribution.

Forest Change Information

- This represents multiple bands related to pixels identified as forest disturbance. The bands contain the disturbance date (band 1) and magnitude of NDFI (CODED), and VH (Sentinel-1) (band 2).

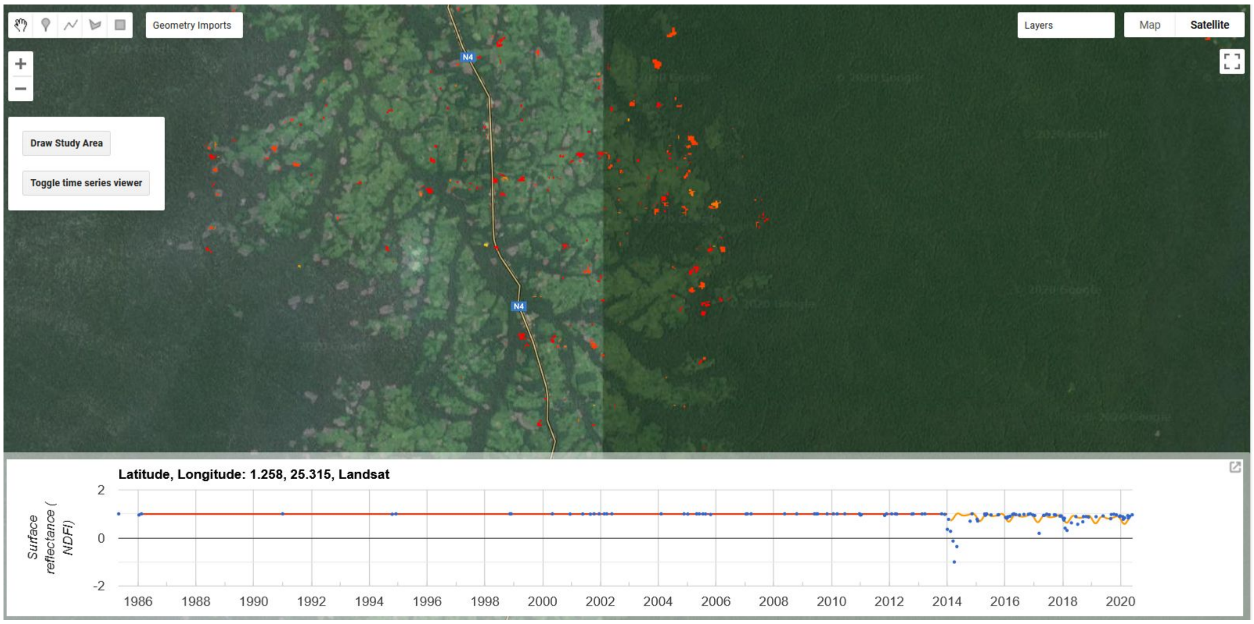
Stratification

- This layer represents a single-band stratification of the study area according to disturbance and land cover history over the study period. The pixel values correspond to stable forest (1), non-forest (2), deforestation (3), degradation (4), and unknown disturbance (5).

Interpreting the Results

Time Series Viewer

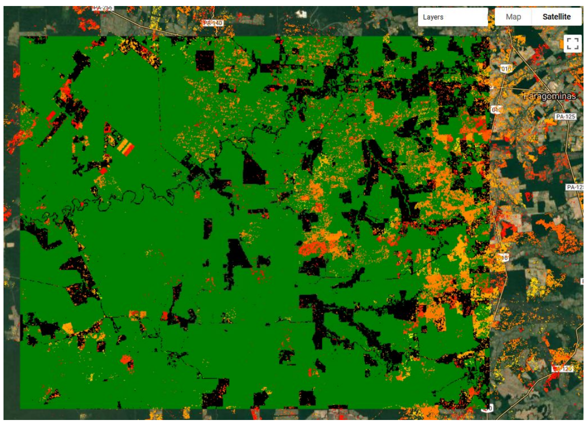
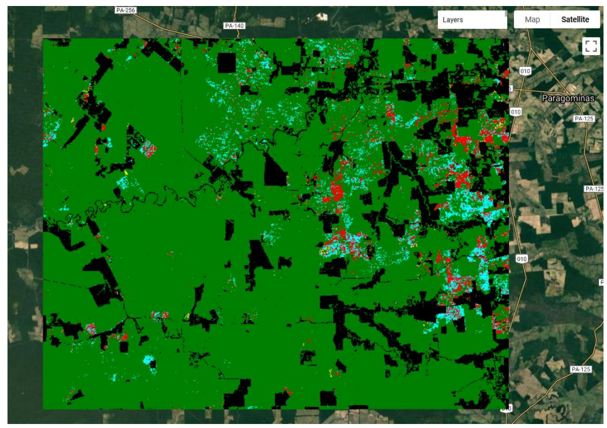
- The second button on the map control panel (Figure 1, C) will add and remove the time series viewer from the map.



- The time series with the model fit can be displayed for any location by clicking on the map. In the example above, the change date layer from Landsat/CODED is shown on the map for a location in the Democratic Republic of the Congo.
- To view the image for a specific data value on the chart on the map, simply click on it in the time series viewer.

Interpreting the Map

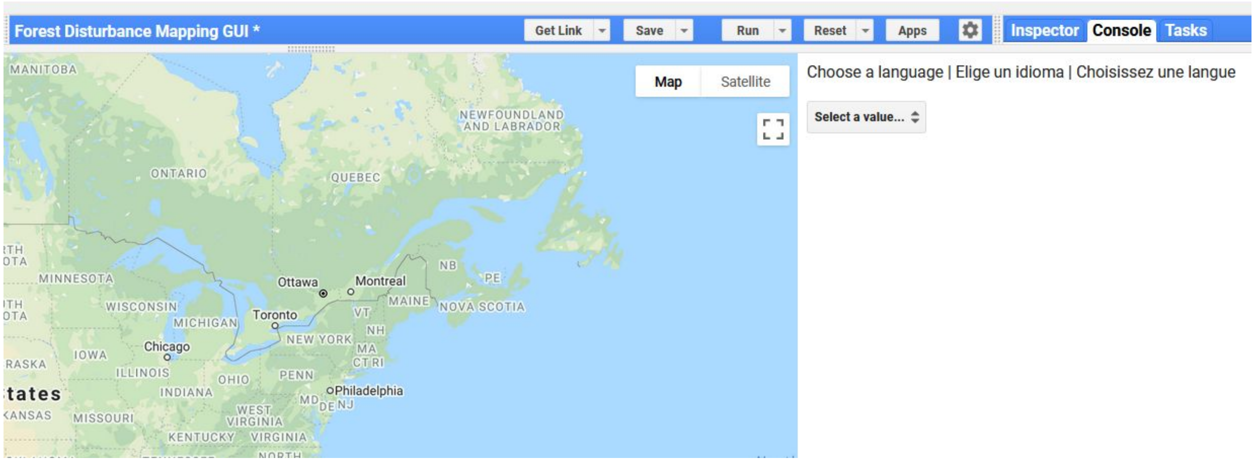
- The layers selected in the Visualization parameters will be added to the map after clicking the ‘Run!’ button.

Break Date Yellow (Start Date) -> Red (End Date)	**Strata** Green: Forest Black: Non-Forest Red: Deforestation Blue: Degradation Yellow: Unclassified Disturbance
	

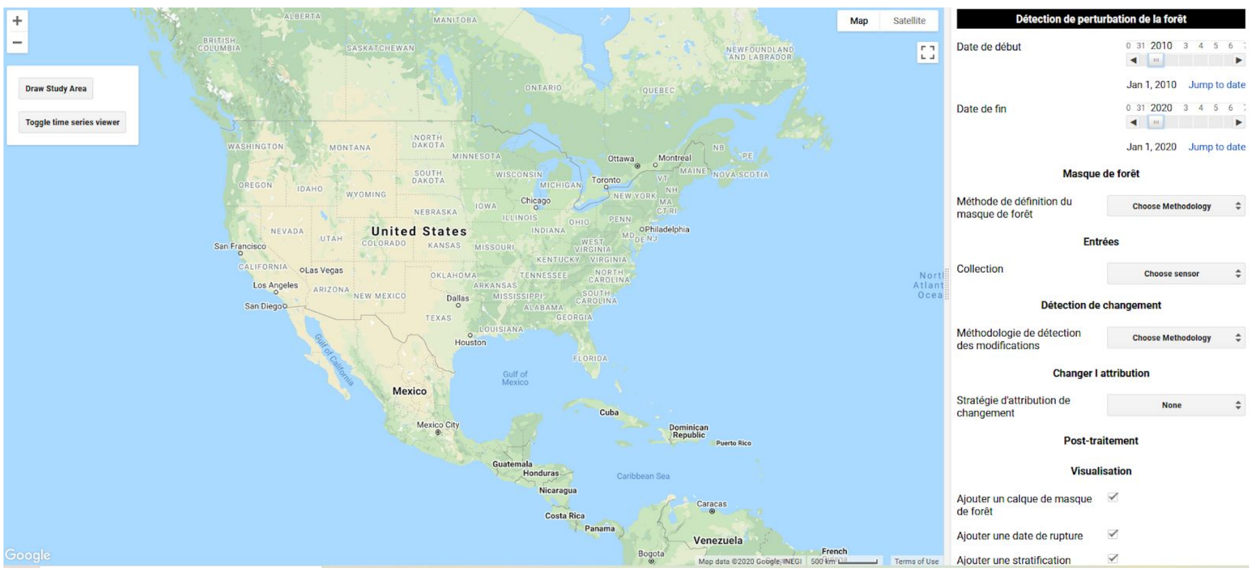
Tutorial: Colombia

Running the application

1. If you have not done so already, add the CODED repository on Google Earth Engine https://code.earthengine.google.com/?accept_repo=users/bullocke/coded
2. Navigate to the script titled ‘Forest Disturbance Mapping GUI’ in the CODED repository.
3. Click ‘Run’.



4. The panel that appears next to the map allows for the selection of a language. Choose your preferred language. Currently, the only options are English and French.



Parameterization

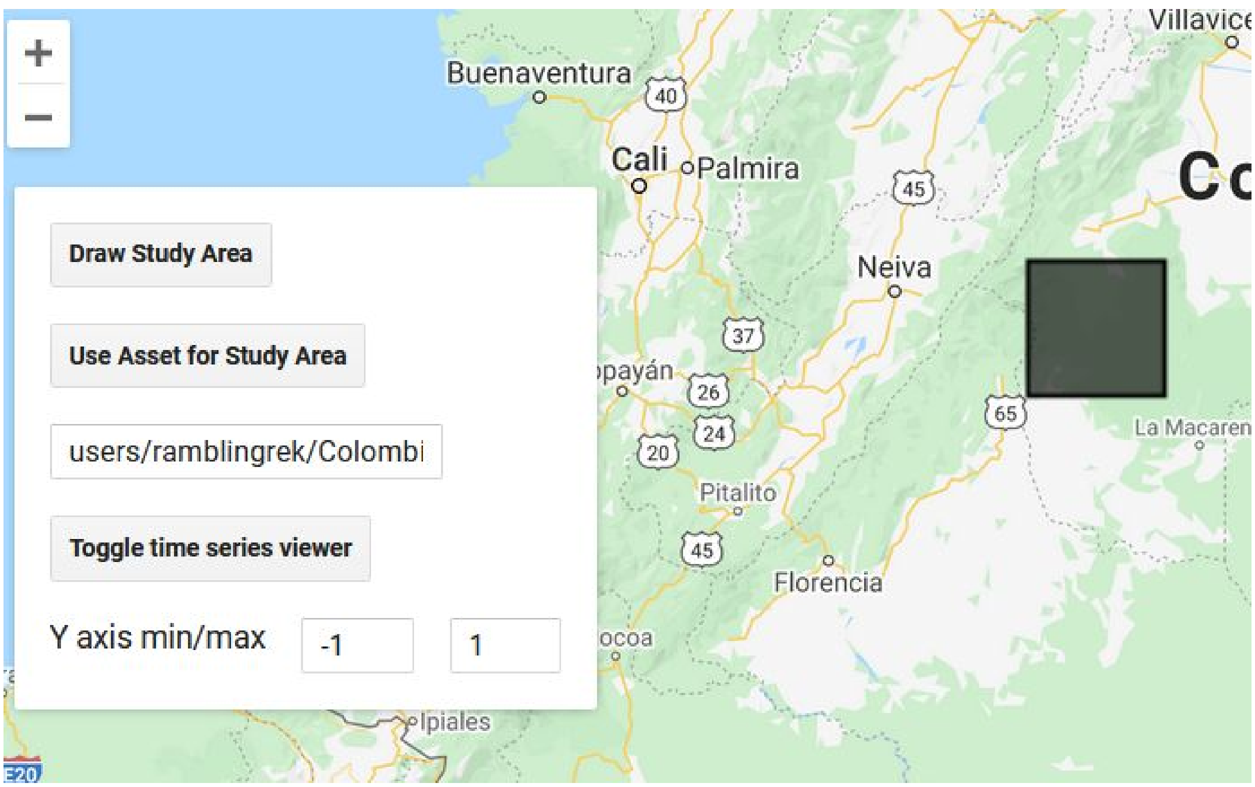
The following are a recommendation of parameters to use based on previous experience. Everything can and should be tuned for local analysis, and details on parameters can be found in the Application Guidebook.

1. **Start and End date:** Change the start and end dates to ‘2000-01-01’ and ‘2020-01-01, respectively.
2. **Forest mask**: Global Forest Watch (GFW): This uses the Hansen et al. (2013) global tree canopy cover layer for 2000. Everything in the GFW layer greater than the tree cover threshold layer will be classified as forest.
    a. Tree cover threshold: *80.* This defines every pixel with over 80% tree cover to be treated as forest.

3. **Inputs:** Use *Landsat* with all of the default values.
4. **Change Detection Method:** Currently, the only supported change detection methodology with Landsat is *CODED*. The default parameters are fine for initial testing.
5. **Change Attribution:** *Sample* will automatically sample the forest/non-forest map for change attribution. After a change is detected, the subsequent land covers classified as non-forest will be labeled *Deforestation*, while a disturbance in a forest that remains forest will be labeled *Degradation*
6. **Post-Processing:** Use the default post-processing parameters, which performs no post-processing.
7. **Visualization:** Enable 'Verbose" to read information in the Console.
8. **Export:** Export the default ('Stratification').

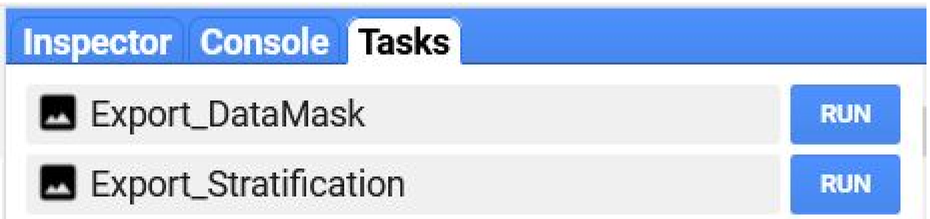
Study Area

1. You can specify a path to an Earth Engine Table, the boundary of which will be used as the study area.
2. Click the button that says "Use Asset for Study Area".
3. Enter the path to the table in the box right below the textbox under the button: users/ramblingrek/ColombiaRectangle
4. After entering the path to the Table, it should load on the map after clicking on the map or pressing 'Enter'.

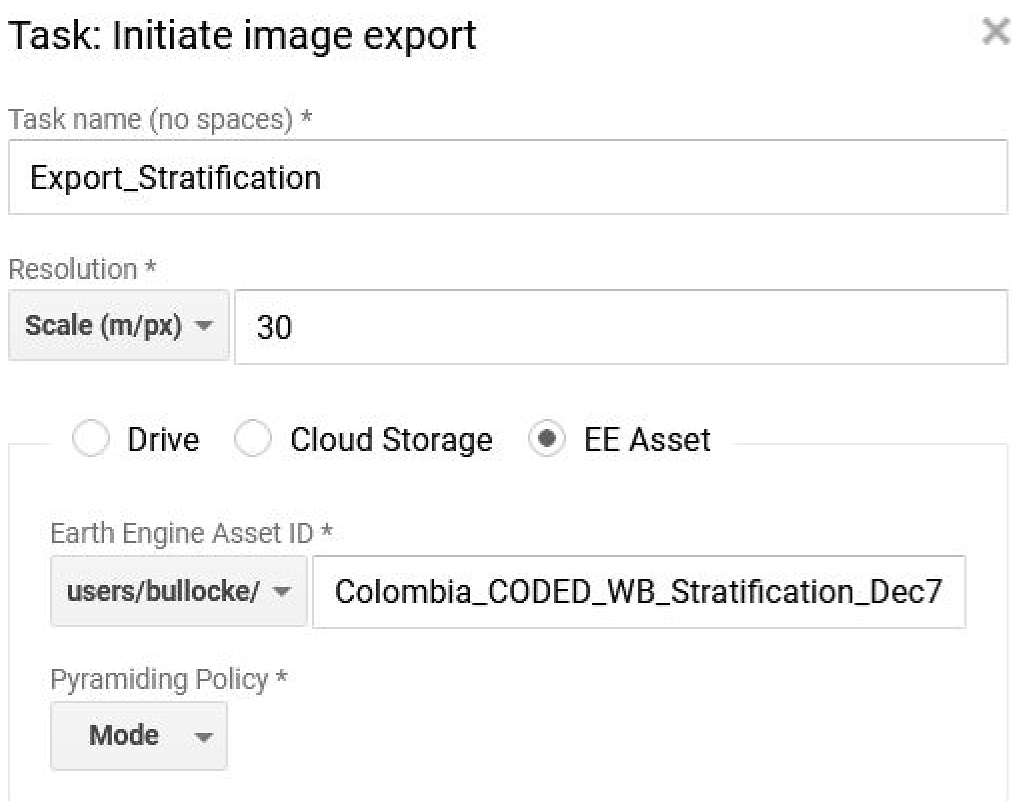


Run the script

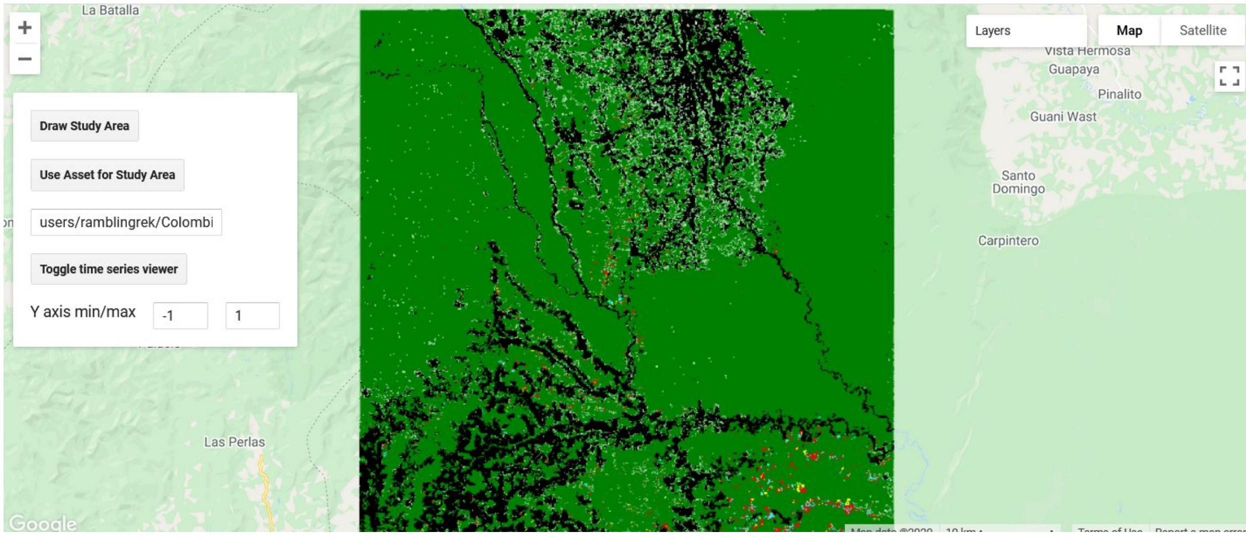
1. Execute the script to display the results. The default options will display the forest change date and stratification.
2. The change date layer is displayed from the start of the study period to the end, corresponding to yellow to red in the map.
3. The stratification has the following legend:
    a. Green: Stable forest
    b. Black: Non-forest
    c. Red: Deforestation
    d. Blue: Degradation
    e. Yellow: Unclassified disturbance
4. Exports can be submitted as a GEE task using the ‘Task’ tab:



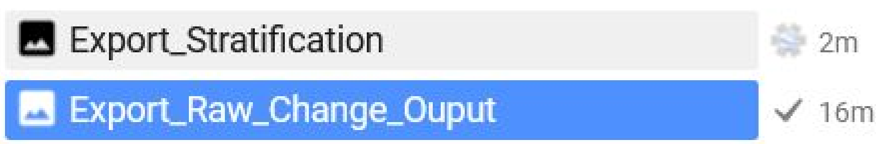
5. After clicking ‘Run’ in the ‘Tasks’ tab for the ‘Export_Stratification’ task, give the file an export name and click ‘Run’.



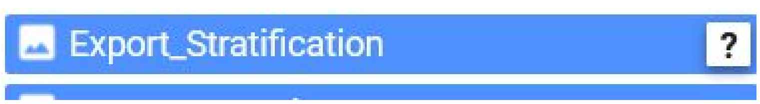
6. After a few minutes, you should see the ‘Change Date’ and ‘Stratification’ layers added to the map. The Forest Mask layer is also available in the map.



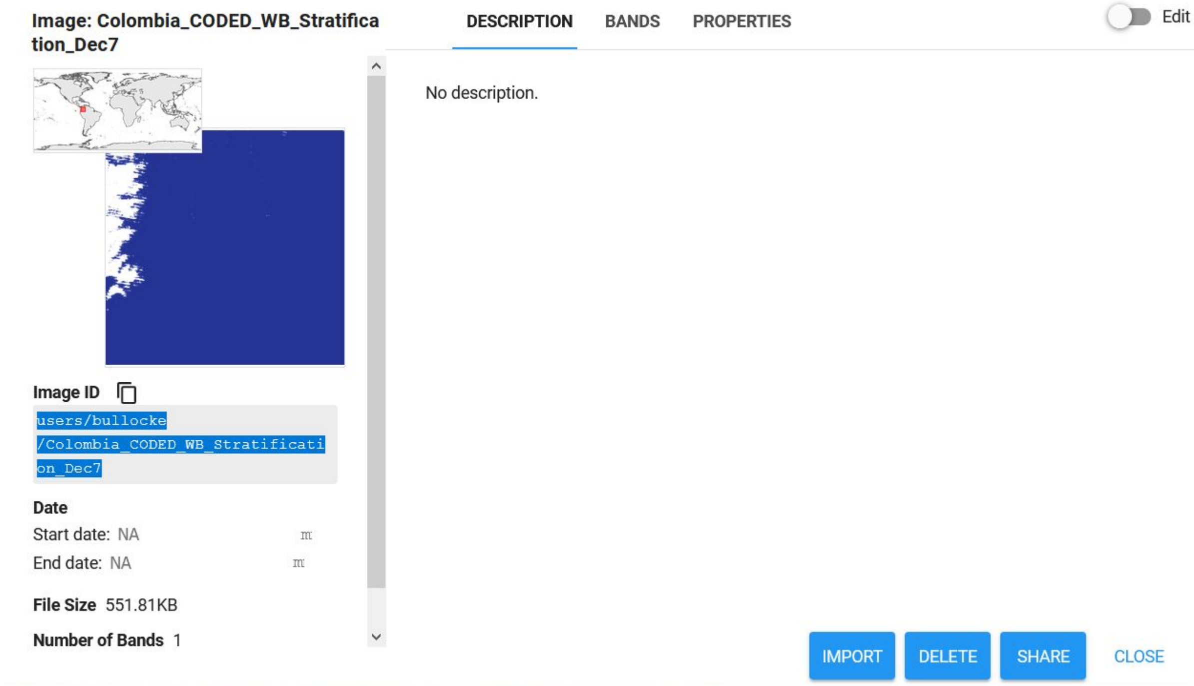
7. If you ever run into an issue with the results loading on the map, or loading too slow, you can export the ‘Stratification’ layer using a task and add it back to the map.
    a. Submit a task (‘Export_Stratification’) to create the single-layer stratification in step 5.
    b. When the task is complete it will turn blue in the ‘Task’ tab.



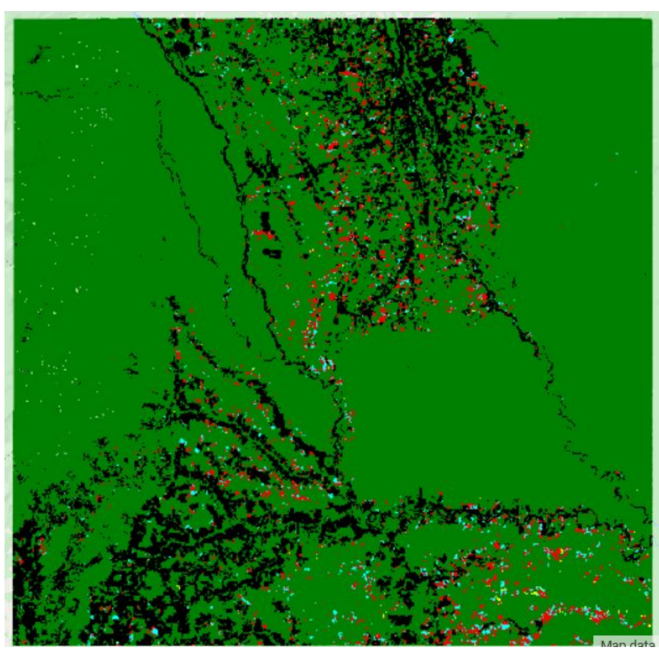
    c. Click on the question mark on the right of the blue task box and click ‘View Asset.



    d. Copy the path to the saved asset.



e. Under “Visualization”, paste the path in the box next to “Asset path” and click “Load asset”. Three layers should be added to the map: one for all pixels mapped as degradation, one for all pixels mapped as deforestation, and the stratification as styled with the palette in step 3.



f. Select the ‘Toggle time series viewer’ to enable clicking on the map and viewing the respective NDFI time series.

g. To start to evaluate the results, try clicking on the map for a pixel mapped as degradation (blue) or deforestation (red). The time series for that pixel should load on the plot at the bottom of the screen. Optionally, use the ‘Y axis min/max’ textboxes to change the y-axis. A few extra layers get added to the map including a box indicating where you clicked, the image collection used to create the plot, and the regression model coefficients.