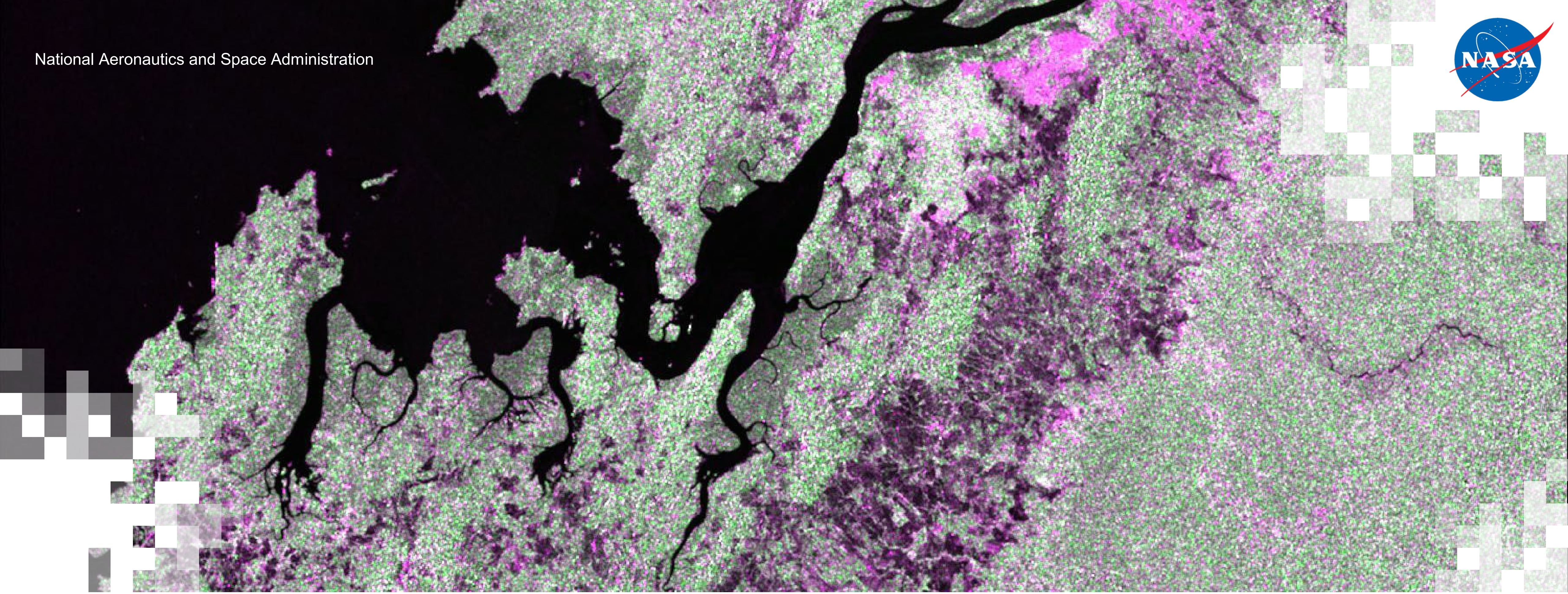
National Aeronautics and Space Administration



Forest Mapping and Monitoring with SAR Data: Land Cover Classification with Radar and Optical Data

Erika Podest, Amber McCullum, Juan Luis Torres Perez, Sean McCartney

May 14, 2020

APPLIED REMOTE SENSING TRAINING PROGRAM

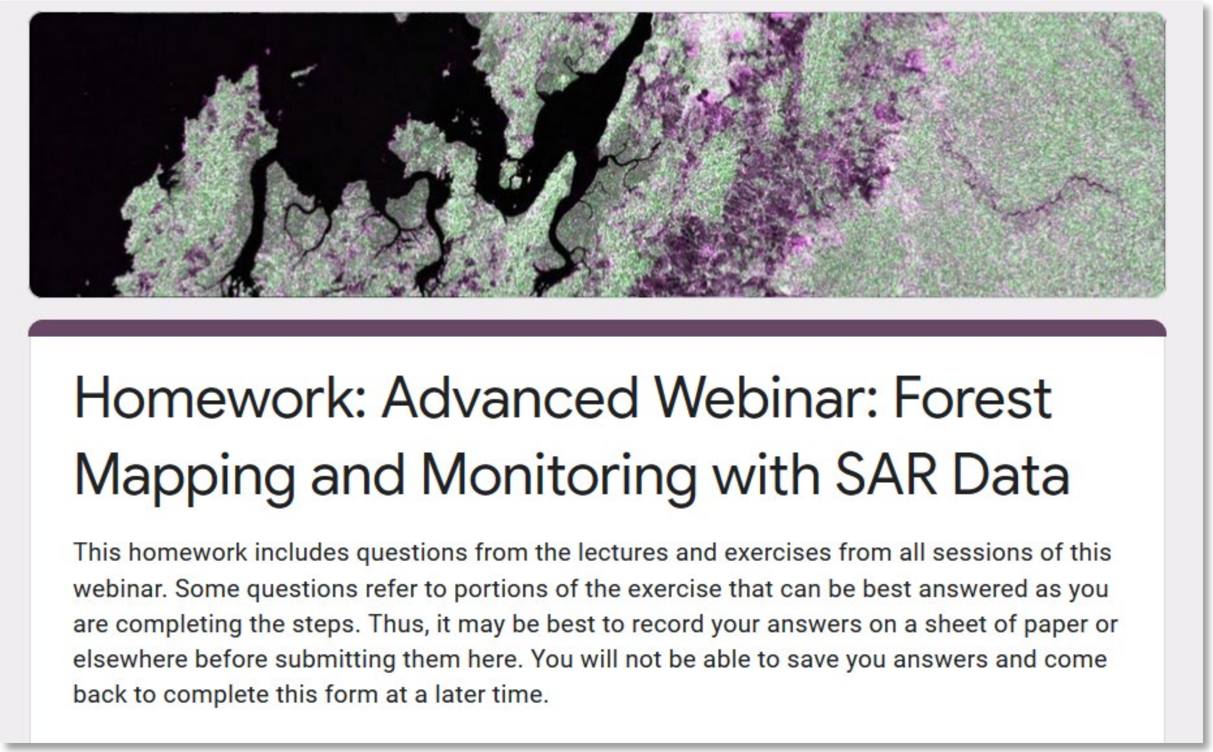
TRAIN - EMPOWER - ADVANCE

Course Structure

- Four, 2-hour sessions on **May 12, 14, 19,** and **21**
- There will be 2 sessions per day presenting the same material in
  - English (11:00-13:00 EST)
  - Spanish (14:00-16:00 EST)
  - **Please only sign up for and attend one session per day.**
- Webinar recordings, PowerPoint presentations, and the homework assignment can be found after each session at:
  - https://arset.gsfc.nasa.gov/land/webinars/forest-mapping-sar
  - Q&A: Following each lecture and/or by email
    - erika.podest@jpl.nasa.gov
    - amberjean.mccullum@nasa.gov
    - juan.l.torresperez@nasa.gov

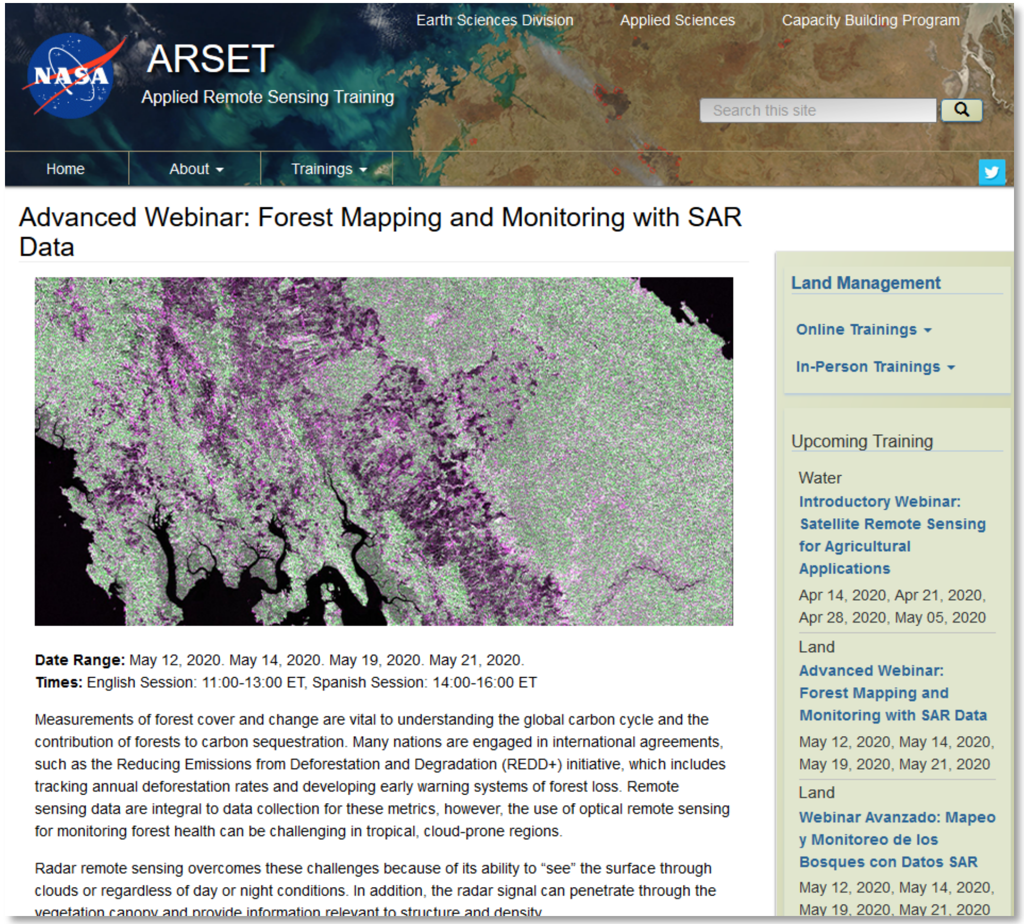
Homework and Certificates

- **Homework:**
  - One homework assignment
  - Answers must be submitted via Google Forms
- **Certificate of Completion:**
  - Attend all three live webinars
  - Complete the homework assignment by **Thursday, June 4th** (access from ARSET website)
  - You will receive certificates approximately two months after completion of the course from: marines.martins@ssaihq.com



Prerequisites and Course Materials

- **Prerequisites:**
  - Please complete these two courses or have equivalent experience
    - [Introduction to Synthetic Aperture Radar]
    - [Advanced Webinar: SAR for Landcover Applications]
  - Set-up a Google Earth Engine Account (free) here:
  - https://earthengine.google.com
- **Course Materials:**
  - https://arset.gsfc.nasa.gov/land/webinars/forest-mapping-sar



Course Outline

Part 1: Time Series Analysis of Forest Change

Part 2: Land Cover Classification with Radar and Optical Data

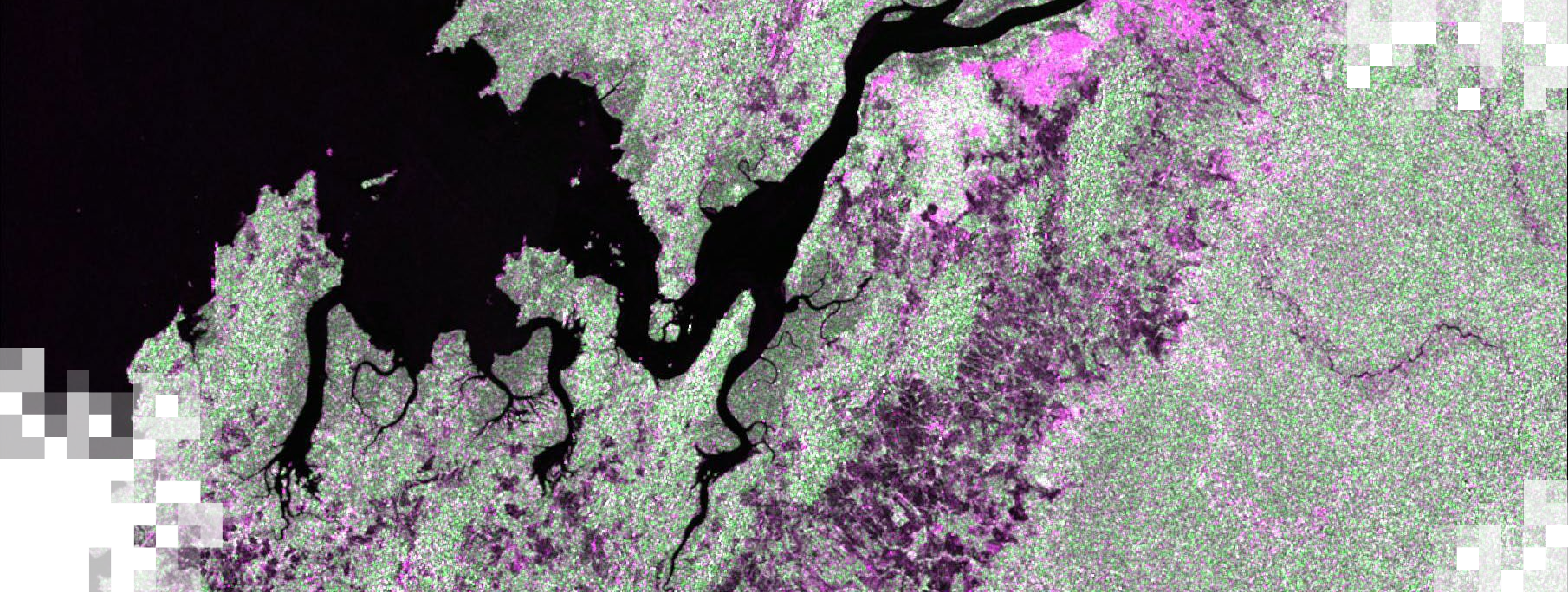
Part 3: Mangrove Mapping

Part 4: Forest Stand Height

Learning Objectives

By the end of this presentation, you will be able to:

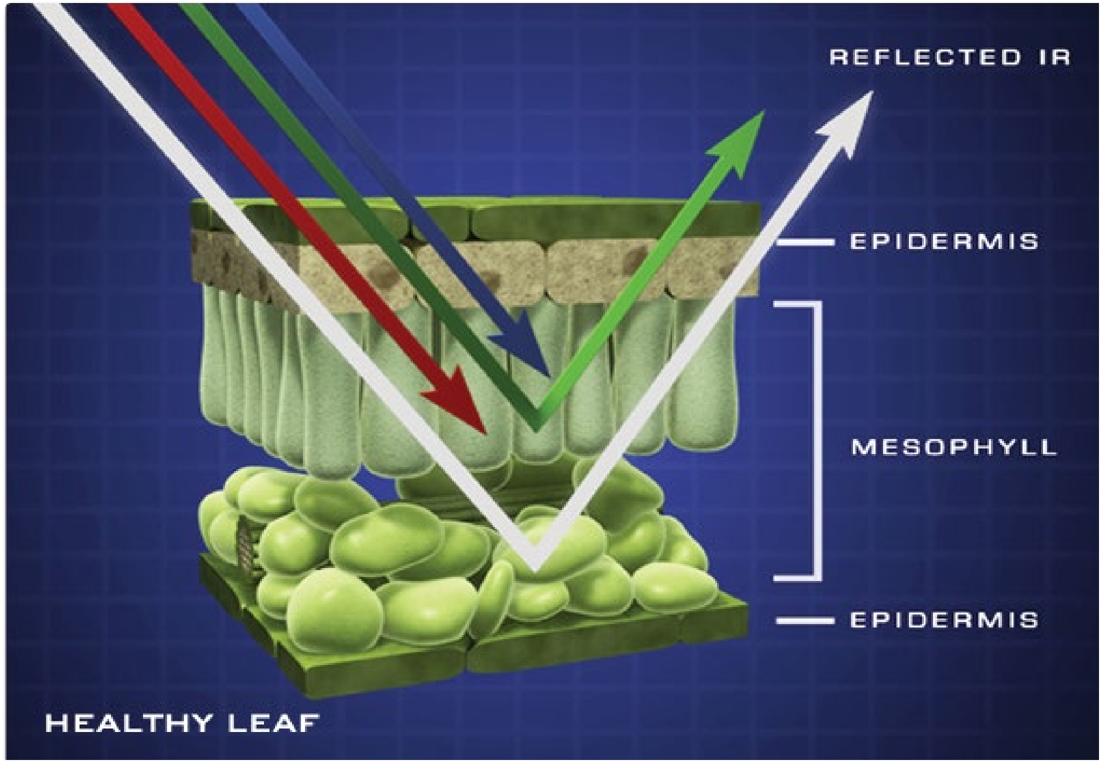
- Identify the unique attributes of radar and optical data
- Explain the benefits and limitations to radar and optical data for forest mapping
- Understand the basics of land cover classification using both radar and optical data
- Conduct a land cover classification using Landsat and Sentinel-1 data in Google Earth Engine



Optical Data Review

Spectral Signatures

- Every surface on Earth reflects and absorbs energy in different ways.
- Spectral signature is the unique way a surface reflects energy.
- We typically characterize spectral signatures in a graph:
  - Percent reflectance on the y-axis
  - Wavelength on the x-axis
- Example: Healthy, green vegetation <u>absorbs</u> **Blue** and **Red** wavelengths (used by chlorophyll for photosynthesis) and <u>reflects</u> **Green** and **Infrared**.



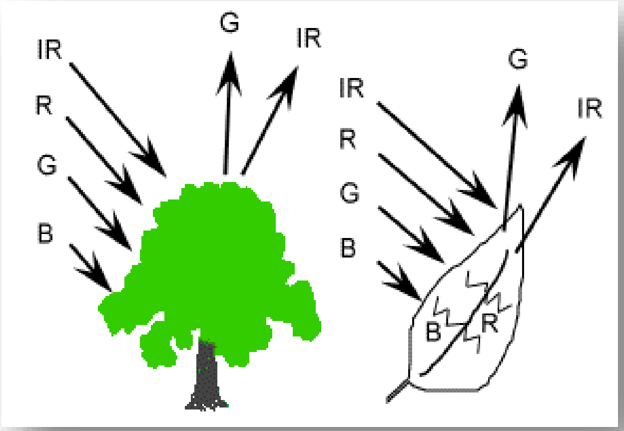
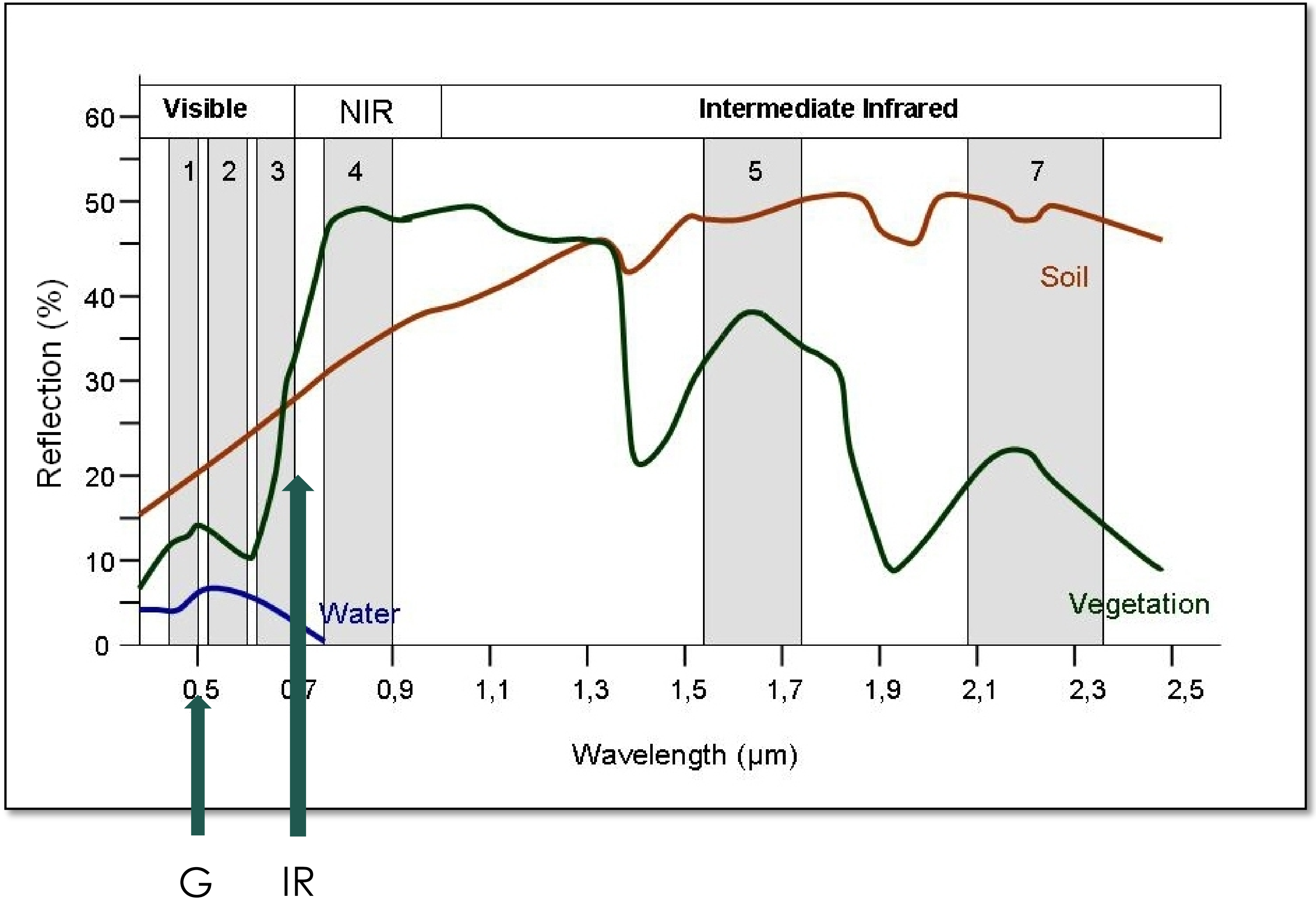


Image Credits: NASA/Jeff Carns & Ginger Butcher

Spectral Signatures

- Different surfaces have different spectral signatures.
- In this example you can see the differences between Water, Vegetation, and Soil signatures.



How Optical Satellites Collect Data

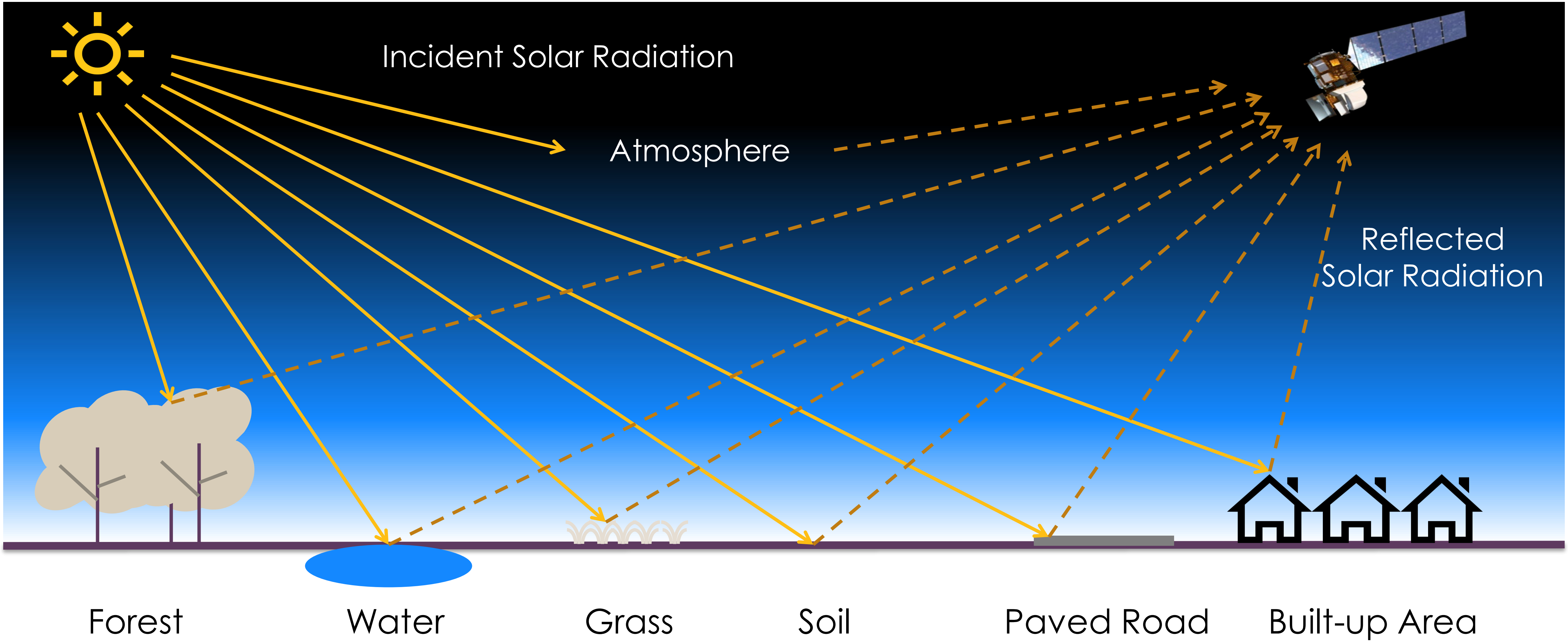
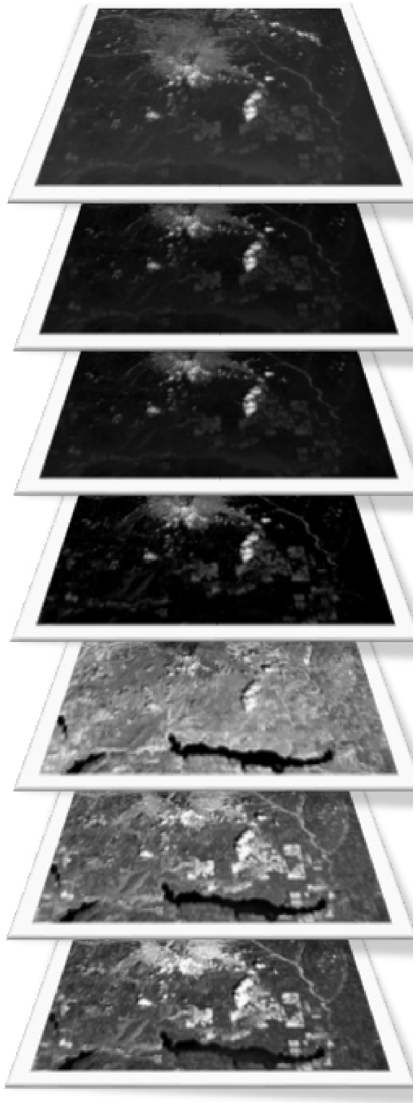


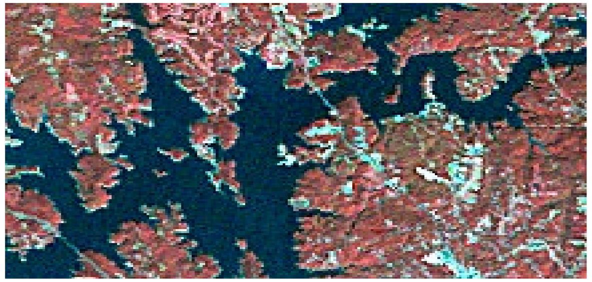
*Image recreated from Natural Resources Canada image.

Image Bands vs. Color Channels

Band 1
Band 2
Band 3
Band 4
Band 5
Band 6
Band 7



	Red Channel	Green Channel	Blue Channel
True Color	Red Band	Green Band	Blue Band
False Color	Near-Infrared Band	Red Band	Green Band
SW Infrared (vegetation contrast)	Mid-Infrared Band	Near Infrared Band	Green Band



Turning Data Into Information

Optical Image Classification

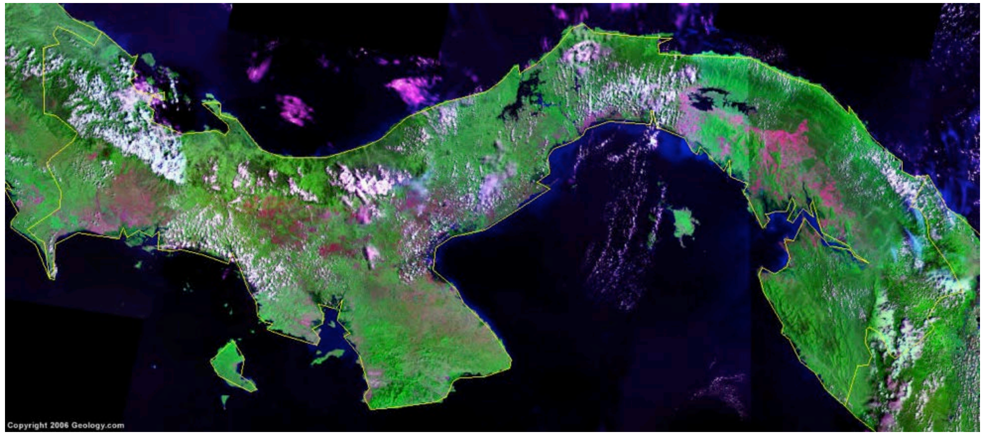
Spectral Classes

- Groups of pixels that are uniform with respect to their pixel values in several spectral bands.

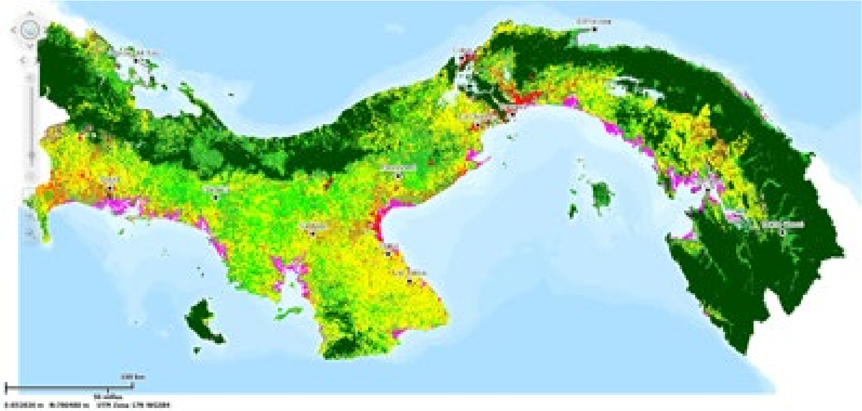
Informational Classes

- Categories of interest to users of the data (i.e. water, forest, urban, agriculture, etc.).

Image classification is the process of grouping spectral classes and assigning them informational class names.



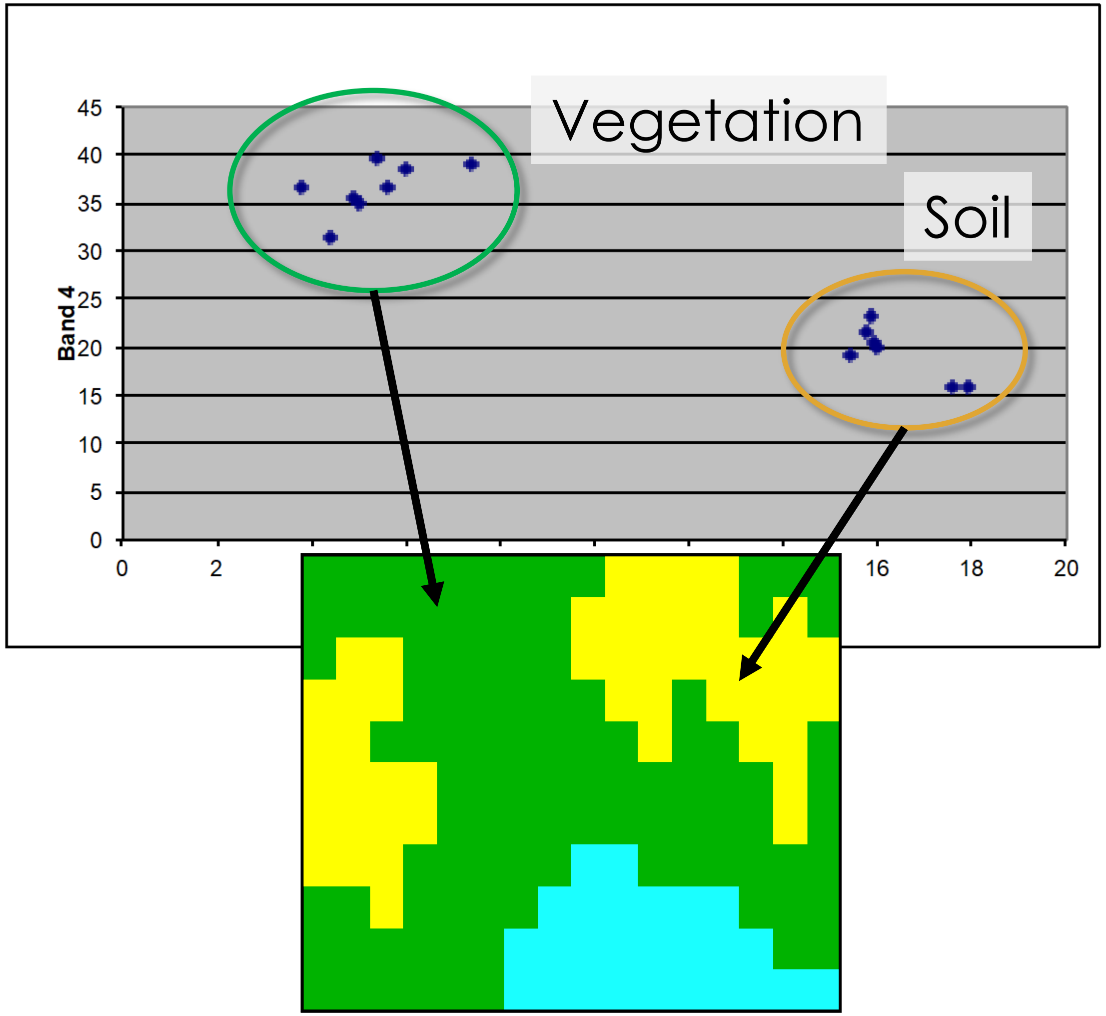
Satellite image of Panama



Land cover map of Panama

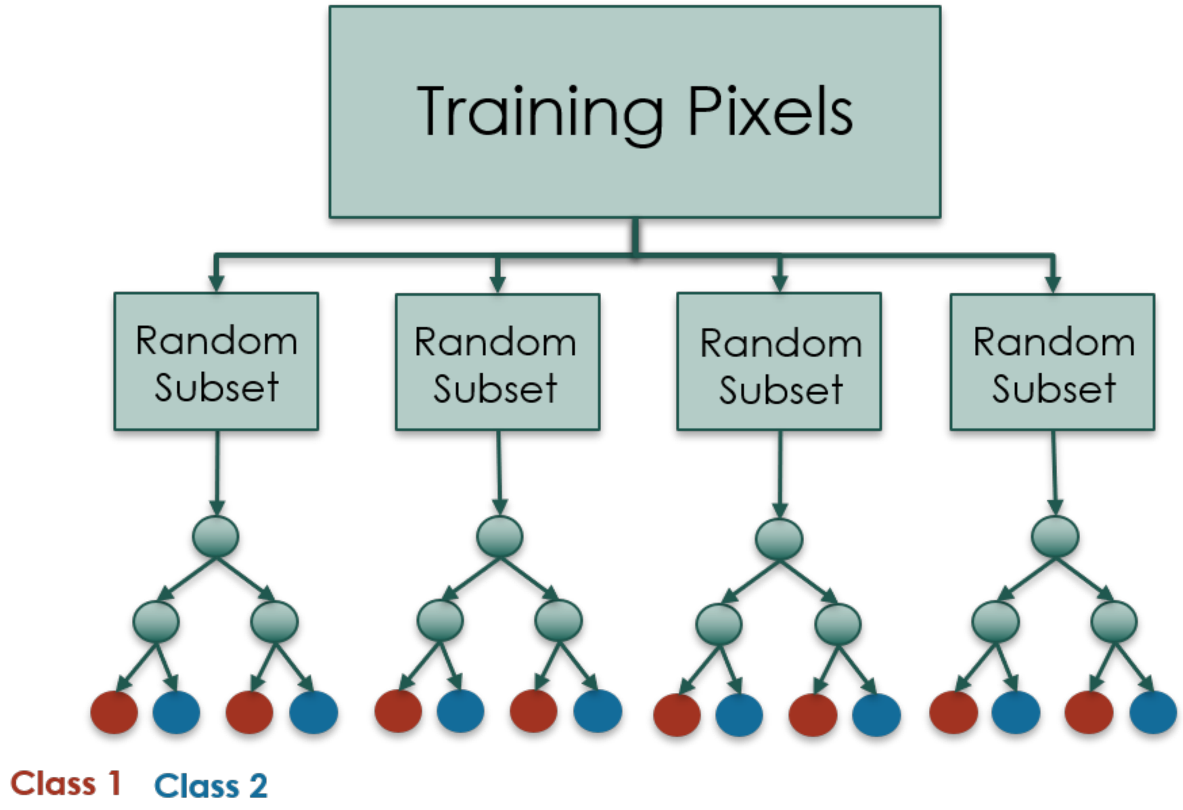
Optical Image Classification

- Requires delineating boundaries of classes in n-dimensional space using class statistics
- Each group of pixels is characterized by:
  - min.
  - max.
  - mean
  - standard deviation
- All the pixels in the image that fall within those statistics are given those labels
- Supervised or Unsupervised



Random Forest Classification Algorithm

- Example of an ensemble model (combines the results from multiple models; logic → result from a combination will be better than from a single model)
- Supervised learning
- Random Forest Algorithm takes a random set of training sites (~2/3) and builds multiple decision (classification) trees; remaining ~1/3 training sites used to estimate error and importance of each predictor variable



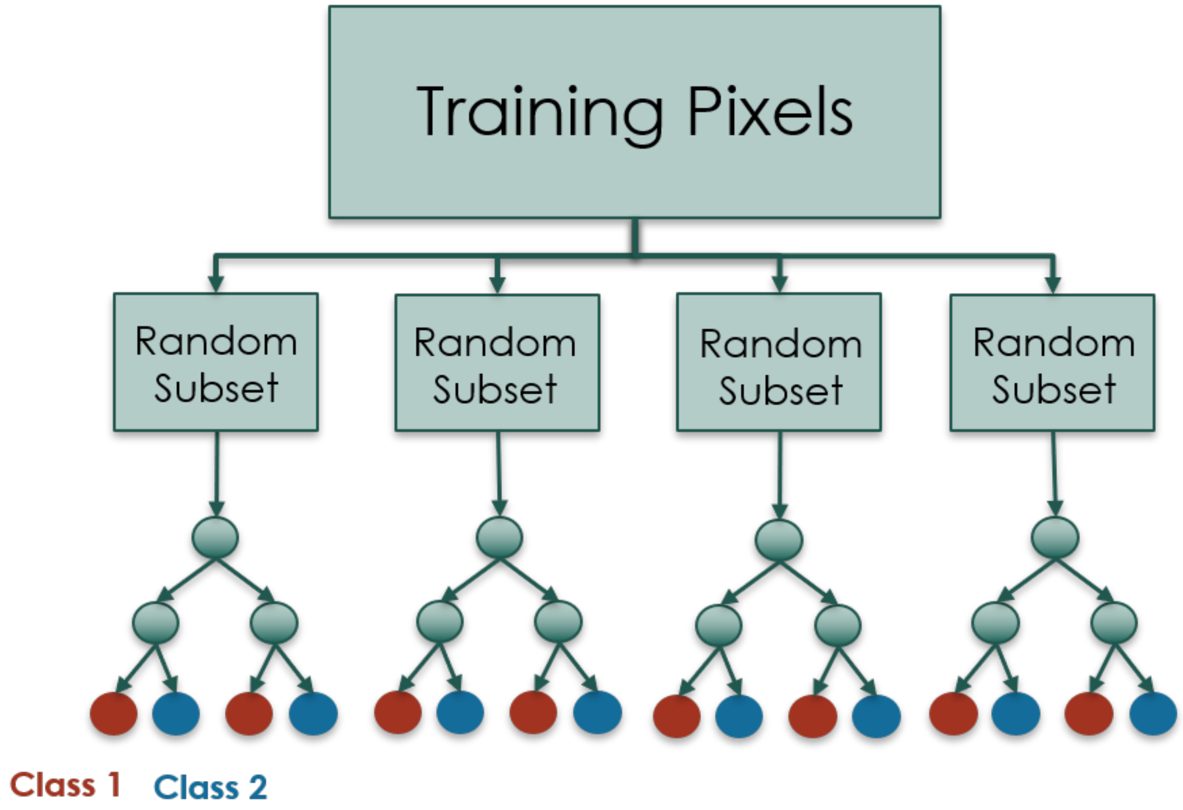
Random Forest Algorithm

Advantages

- No need for pruning
- Overfitting is not a problem
- Not sensitive to outliers in training data
- Easy to parameterize

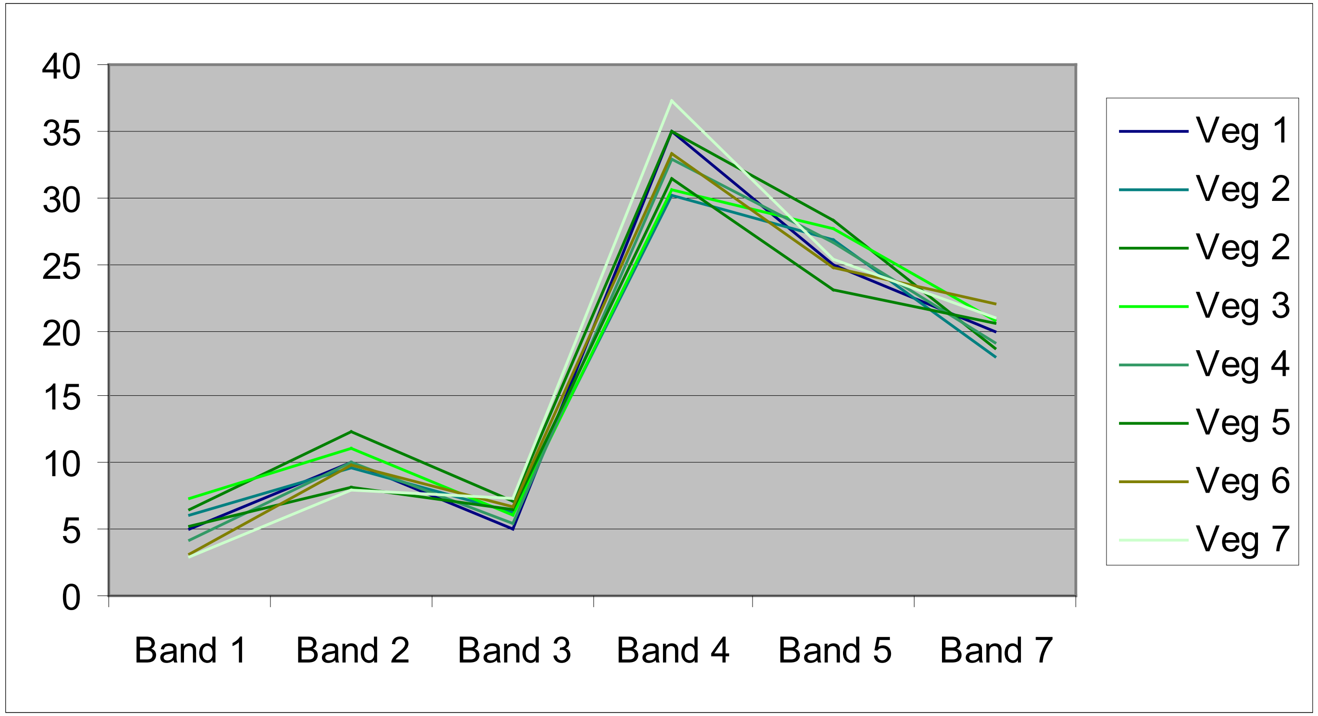
Limitations

- Algorithm cannot predict spectral range beyond training data
- Training data much capture entire spectral range

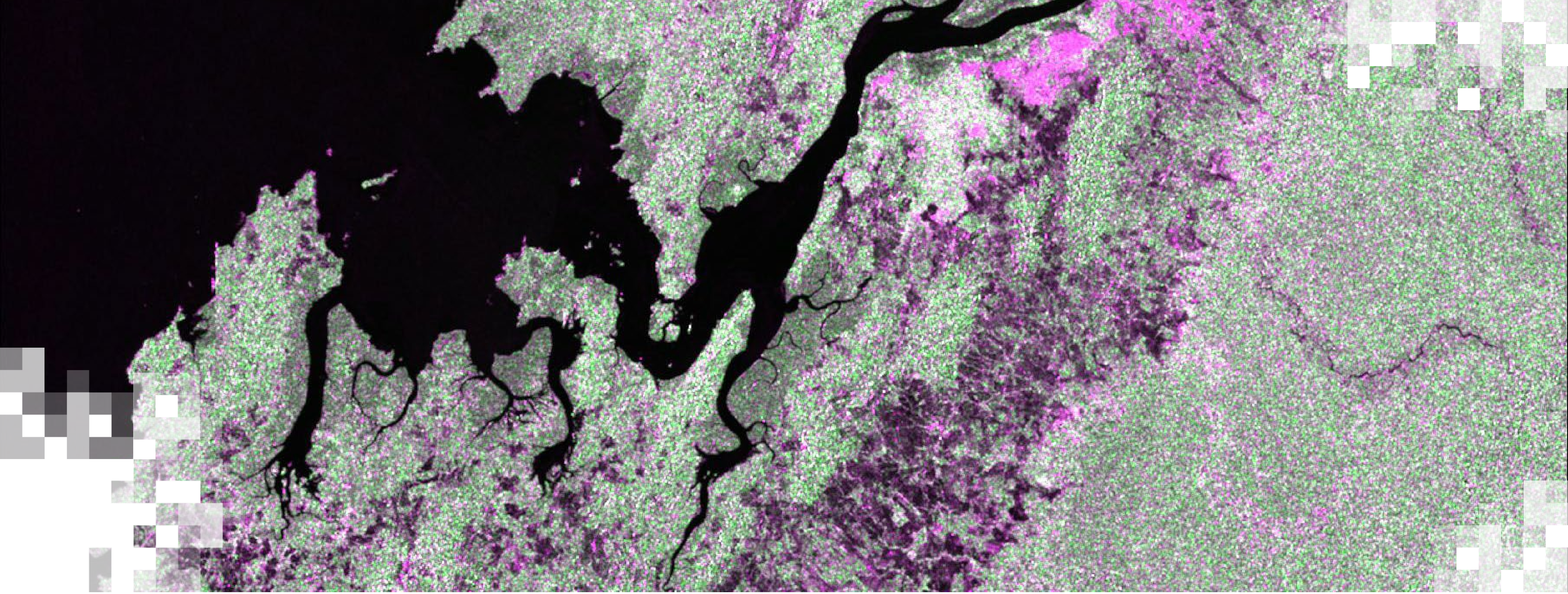


Limitations of Optical Data

- Spatial resolution is often too coarse (for NASA data) to provide high level of detail on the ground
- Spectral resolution is often too coarse to distinguish between different vegetation types
- Does not penetrate clouds and smoke
- Cannot penetrate forest canopy

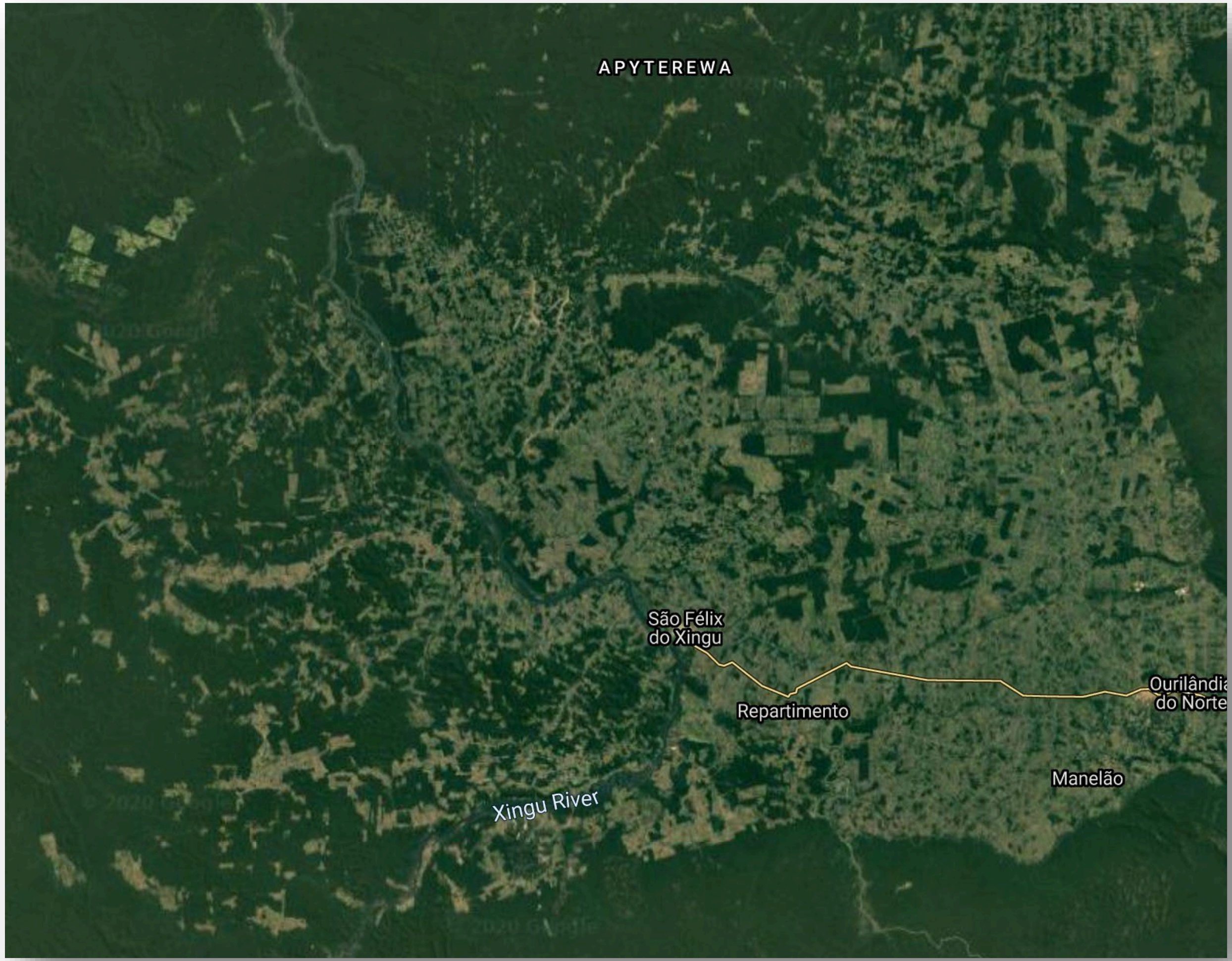


Example of the spectral similarities among different vegetation types

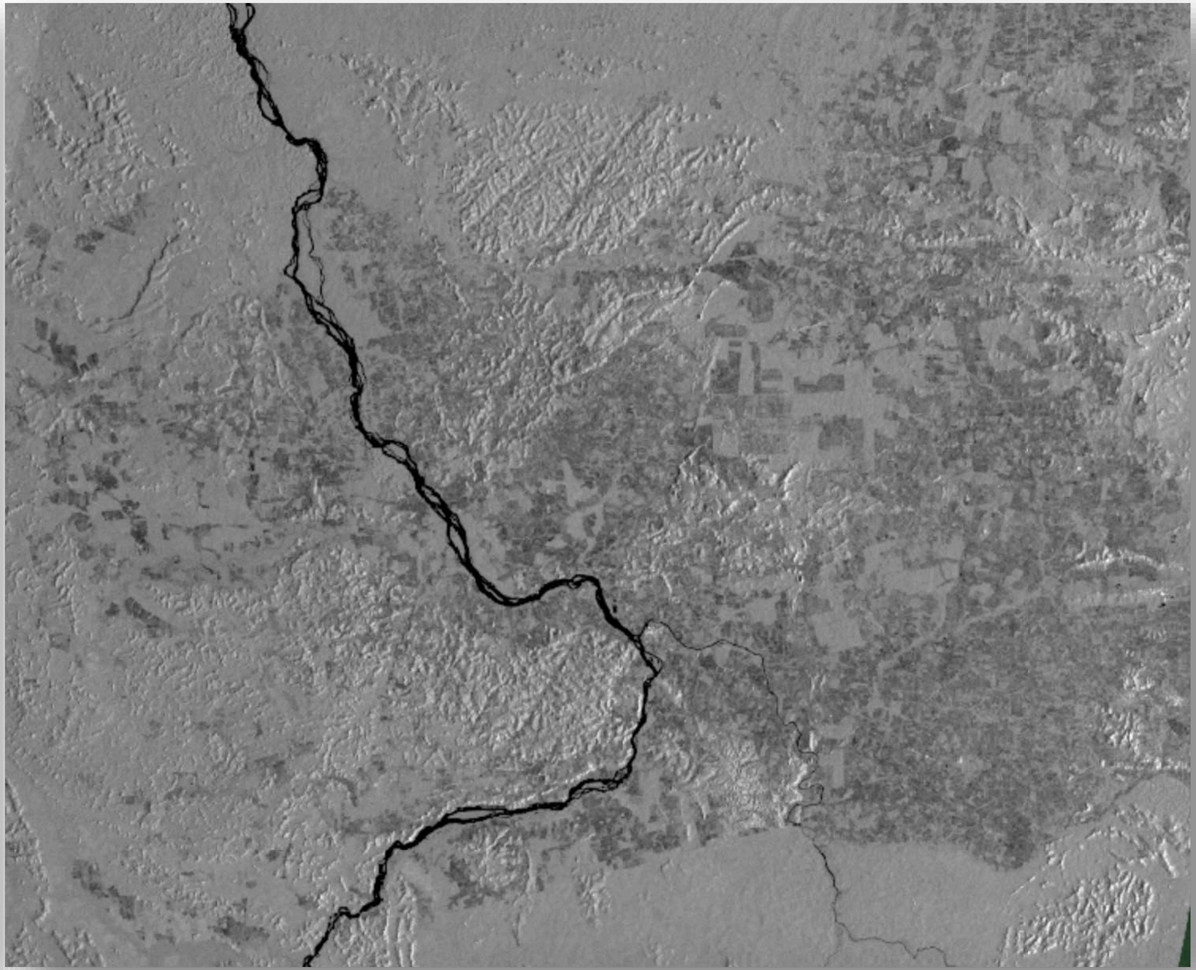


Optical vs. Radar Data Overview

Forest Monitoring with Optical and Radar Data



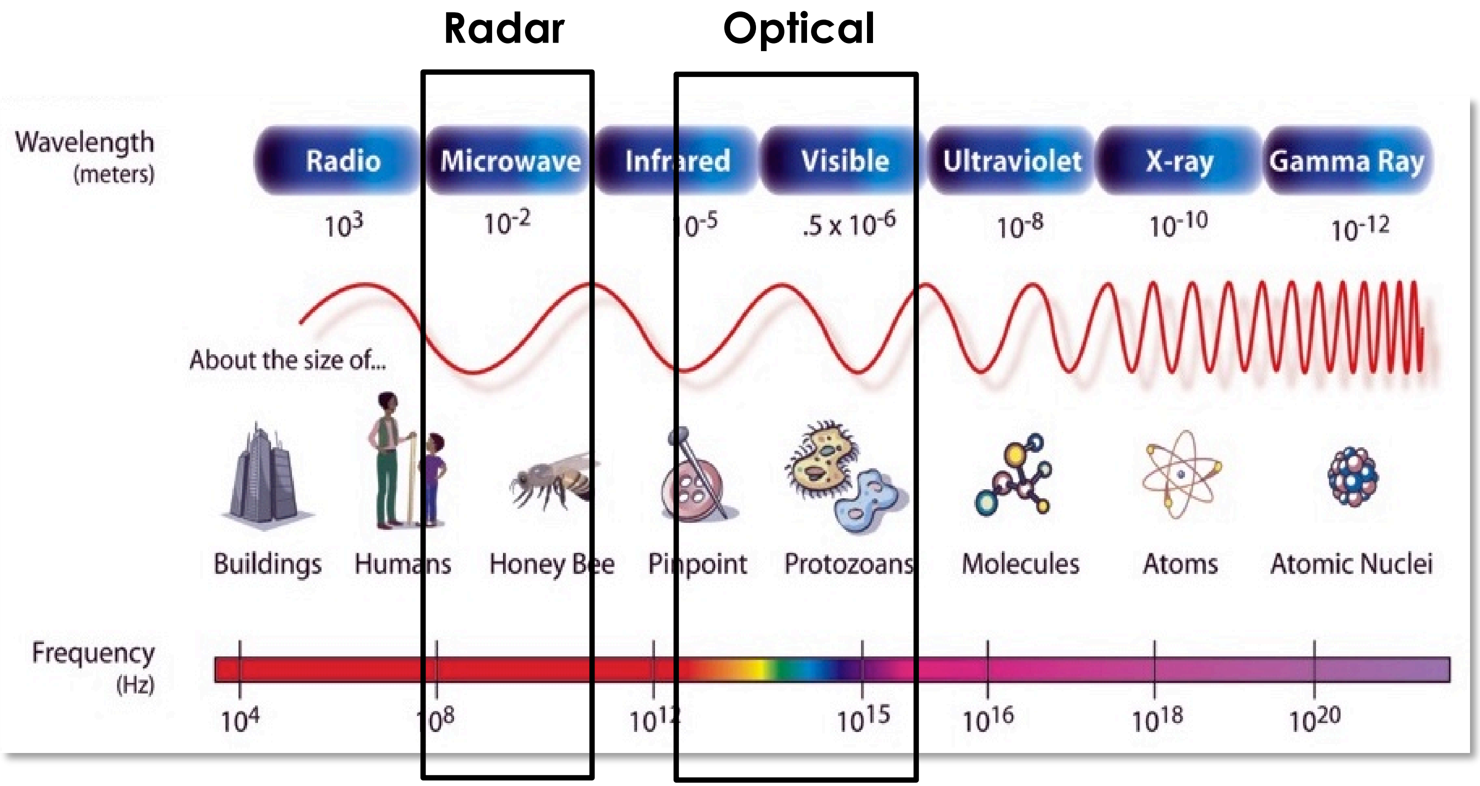
Landsat (Optical)



Sentinel-1 (SAR)

The Electromagnetic Spectrum

- Optical sensors measure reflected solar light and only function in the daytime.
- The surface of the Earth cannot be imaged with visible or infrared sensors when there are clouds.
- Microwaves can penetrate through clouds and vegetation and can operate in day or night conditions.



Land Cover Mapping: Optical vs Radar

Optical

- Energy reflected by vegetation is dependent on leaf structure, pigmentation, and moisture.
- Products are available from visible to infrared wavelengths consisting of several bands of data.
- Optical sensors only see surface tops, because the canopy blocks the understory.

Radar

- Microwave energy scattered by vegetation depends on the structure and moisture/water content of the target.
- Radar data usually consists of 1-2 bands of data.
- The signal can penetrate through the canopy, providing information on soil conditions or inundation state.

Joshi et al., Remote Sens. 2016, 8(1), 70; https://doi.org/10.3390/rs8010070

Advantages and Disadvantages of Radar Over Optical Remote Sensing

Advantages

- Nearly all-weather capability
- Day or night capability
- Penetration through the vegetation canopy
- Penetration through the soil
- Minimal atmospheric effects
- Sensitivity to moisture/water content of the land surface
- Sensitivity to structure

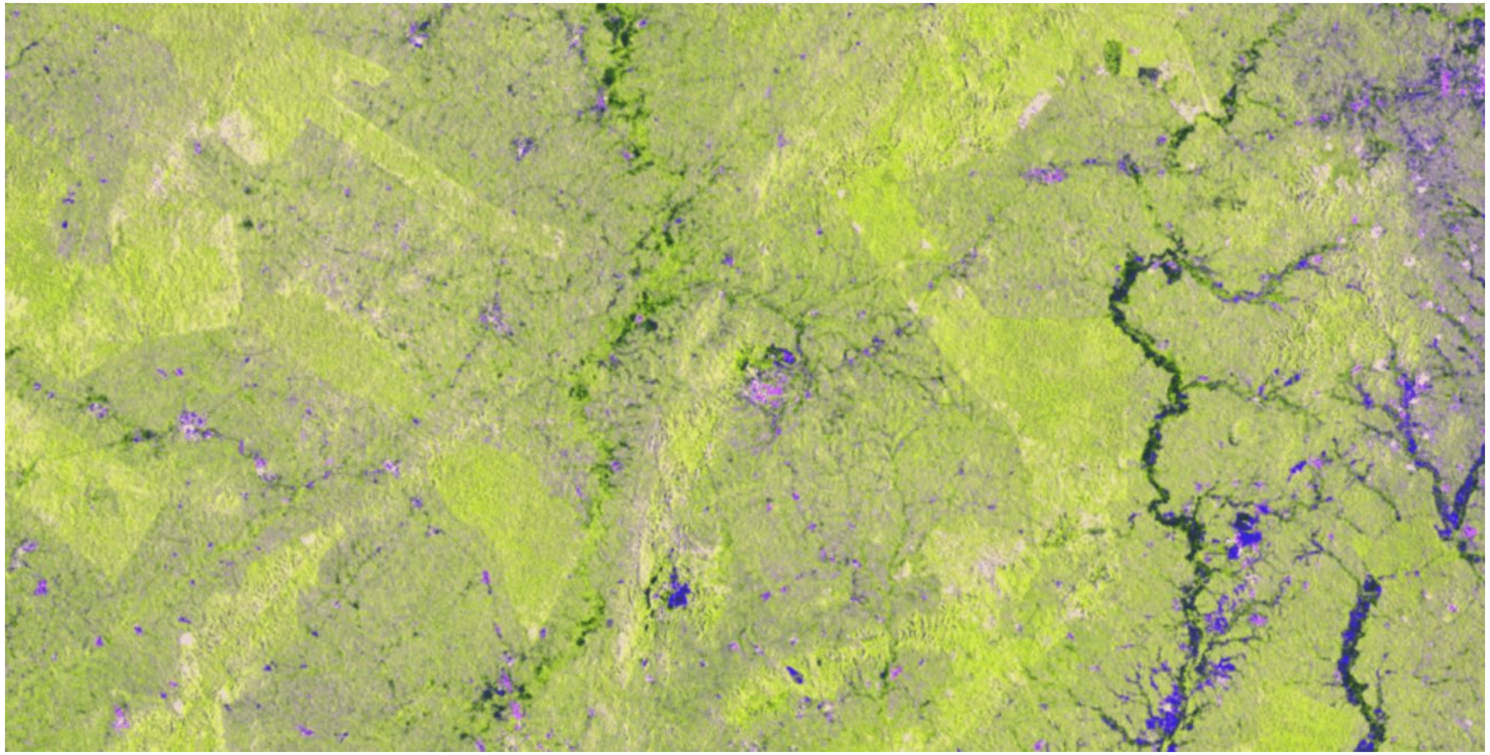
Disadvantages

- Information content is different than optical and sometimes difficult to interpret
- Speckle effects (graininess in the image)
- Topographic differences introduce distortions in the data

Applications of Radar for Land Cover

Mapping and Monitoring:

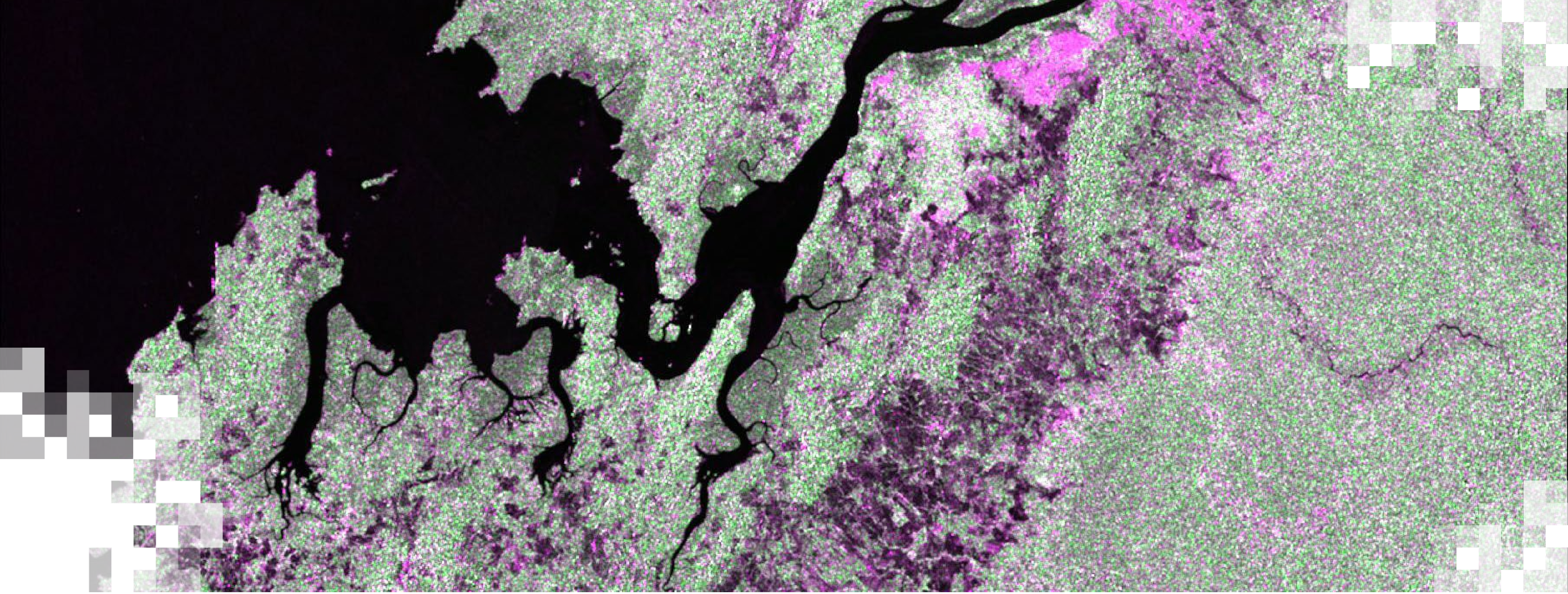
- Forests
- Wetlands
- Biomass
- Disturbances
  - Wildfire
  - Selective Logging
  - Deforestation
  - Reforestation



Identification of vegetation change using Sentinel-1 radar imagery in Ghana.
Image Credit: Satelligence

Benefits to Using both Radar and Optical Data

- Improved land cover classification
- Ability to provide more detailed characterization of land changes
  - Broad classes of land cover and change (optical)
  - Land surface roughness and soil moisture (radar)
- Ability to more accurately monitor vegetation health for agricultural purposes, forest disturbances, and land degradation
  - NDVI and/or EVI (optical)
  - Plant structure and volume (radar)



Hands-on Exercise: Land Cover Classification

Exercise Overview

- Explore the characteristics of Sentinel-1 and Landsat-8 Data
- Select an area of interest to run the analysis
- Load Sentinel-1 and Landsat-8 data
- Apply a speckle filter to the Sentinel-1 images
- Select training classes
- Train and run a Random Forest classifier on the Sentinel-1 image
- Train and run a Random Forest classifier on the Landsat-8 image
- Train and run a Random Forest classifier on the Sentinel-1 and Landsat-8 images
- Generate a confusion matrix and accuracy result for each of the classifications and compare the results

Focus Area

Rondonia, Brazil

Google Earth Engine Optical/Radar Classification Demo Code

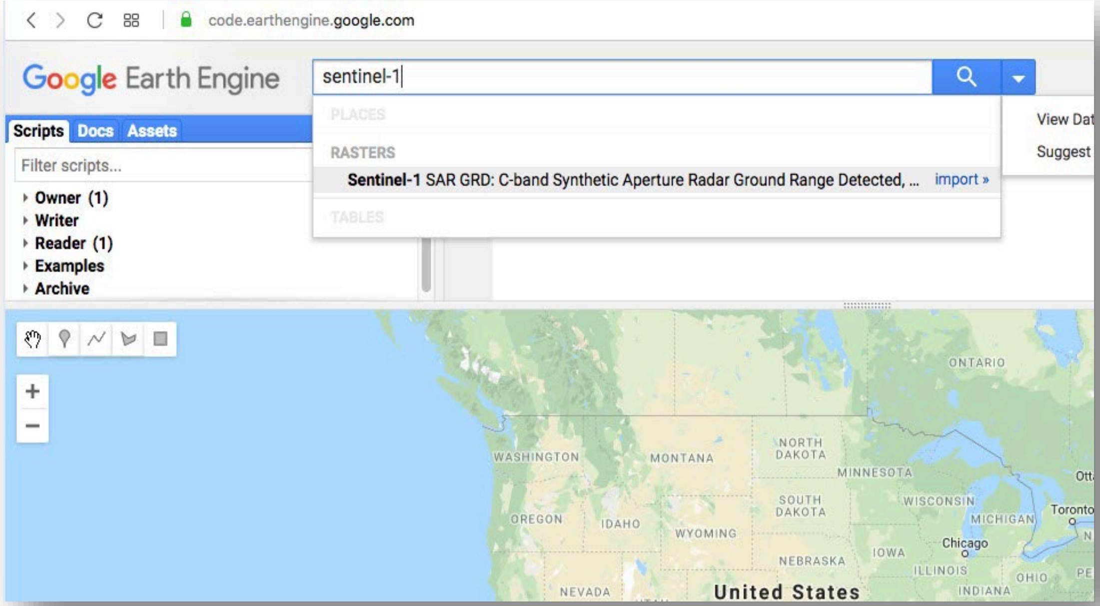
https://code.earthengine.google.com/283acb2ceedee98e77b20ef315b2fab7?accept_repo=users%2Fwolterpt%2FSAR_TimeSeries_PTW

Visualize Sentinel-1 Data

1. Start by opening Google Earth Engine: https://code.earthengine.google.com

2. Search for **Sentinel-1** data
A window with a description of the data will open showing:

- the steps taken to process the data (thermal noise removal, radiometric calibration, terrain correction)
- bands and resolution
- metadata (important parameters are mode and orbit properties - descending or ascending)



3. Repeat a similar search for Landsat 8 and select Landsat 8 Surface Reflectance (SR) Tier 1. Review the description, bands and image properties.

4. Click on the map portion of the page to exit out of the Landsat 8 information tab.

Select Area of Interest

Define your area of interest

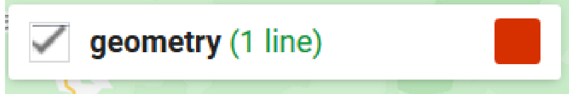
5. Zoom into **Rondonia, Brazil**

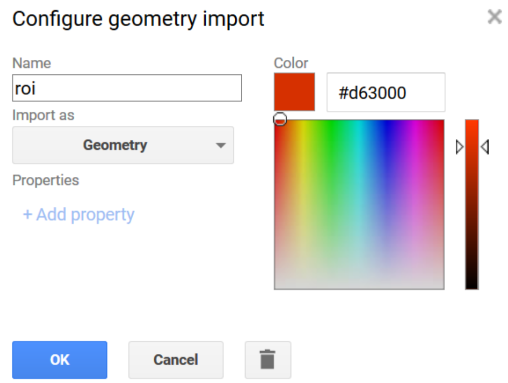
- You can do this by typing Rondonia, Brazil into the search bar along the top, and then zooming into the region until you see the town of Porto Velho

6 Select the **draw a line** icon

7. Draw a rectangle like the one here over our area of interest

8. Hover over **geometry** and then click on the wheel icon to change the name to **roi** (region of interest), then click **OK.**





Filter the Sentinel-1 Data

9. Load the Sentinel-1 database and filter for images that are in Interferometric Wide Swath Mode (IW), Descending Pass, 10-meter resolution, and VV polarization.
In the script editor, add the following code:

```
// Load Sentinel-1 C-band SAR Ground Range collection (log scale, VV,
descending)
var collectionVV = ee.ImageCollection('COPERNICUS/S1_GRD')
.filter(ee.Filter.eq('instrumentMode', 'IW'))
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'))
.filterMetadata('resolution_meters', 'equals' , 10)
.filterBounds(roi)
.select('VV');
print(collectionVV, 'Collection VV');
```

** Note that all Sentinel-1 images in Google Earth Engine are in dB*

Filter the Sentinel-1 Data

10. Repeat step 9 but this time filter the data for VH polarization.
Click enter or return and add the following code:

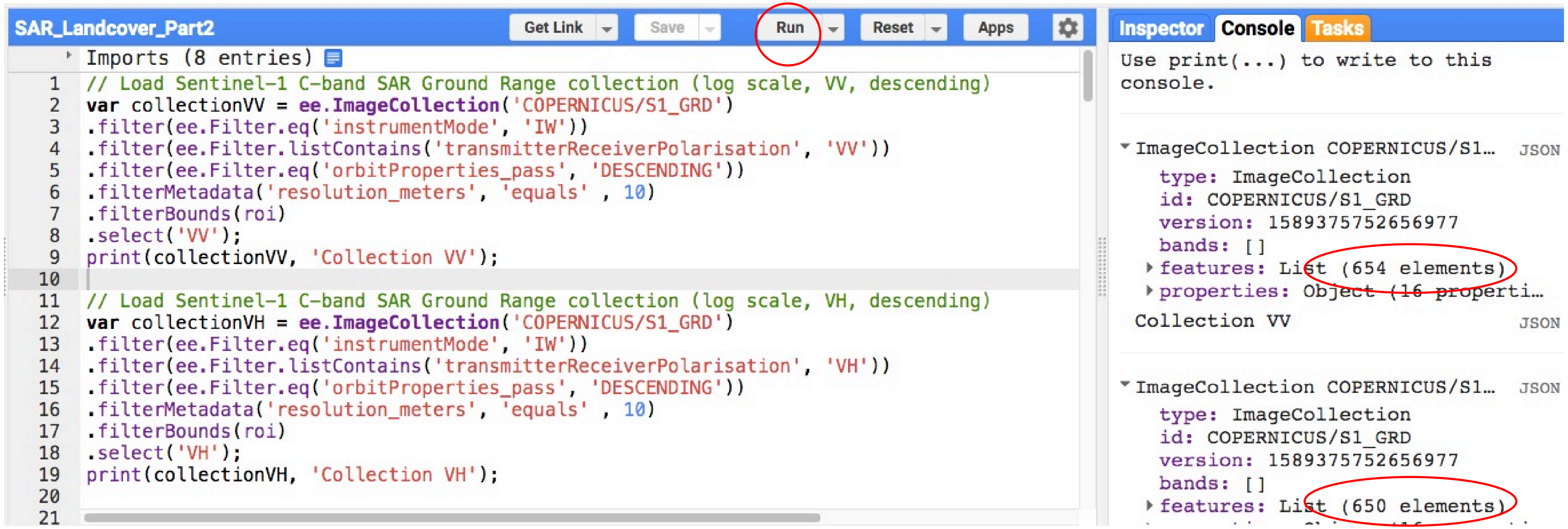
```
// Load Sentinel-1 C-band SAR Ground Range collection (log
scale, VH, descending)
var collectionVH = ee.ImageCollection('COPERNICUS/S1_GRD')
.filter(ee.Filter.eq('instrumentMode', 'IW'))
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'))
.filterMetadata('resolution_meters', 'equals' , 10)
.filterBounds(roi)
.select('VH');
print(collectionVH, 'Collection VH');
```

** Note that all Sentinel-1 images in Google Earth Engine are in dB*

Filter the Sentinel-1 Data

11. Ensure that your script is the same as the below image
- Note, you may need to enter each piece of the code to be on a separate line so that they are indicated as actions.

12. Click on **Run** in the top menu
- The **Console** window on the right shows the results for VV (654 images) and VH (650 images)



Filter the Sentinel-1 Data

Filter the Sentinel-1 by date

13. Filter by date range.
Click enter or return and add the code below:

```
//Filter by date
var SARVV = collectionVV.filterDate('2019-08-01', '2019-08-10').mosaic();
var SARVH = collectionVH.filterDate('2019-08-01', '2019-08-10').mosaic();
```

14. Click on **Run** in the top menu.

Add the Images to Layers

Add the images to “Layers” in order to display them

15. Add the VV and VH images that were identified in the previous step onto the ”layers” bar in order to then visualize the images.
Click enter or return and add the code below:

```
// Add the SAR images to "layers" in order to display them
Map.centerObject(roi, 7);
Map.addLayer(SARVV, {min:-15,max:0}, 'SAR VV', 0);
Map.addLayer(SARVH, {min:-25,max:0}, 'SAR VH', 0);
```

16. Click on **Run** in the top menu.

Create a Function That Masks Cloud Shadows and Clouds

17. This function uses the Quality Assessment (QA) band of Landsat 8 SR:

```
// Function to cloud mask from the pixel QA band of Landsat 8 SR data.
function maskL8sr(image) {
// Bits 3 and 5 are cloud shadows and clouds, respectively.
  var cloudShadowBitMask = 1 << 3;
  var cloudsBitMask = 1 << 5;
// Get the pixel QA band.
  var qa = image.select('pixel_qa');
// Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
      .and(qa.bitwiseAnd(cloudsBitMask).eq(0));
// Return the masked image, scaled to reflectance, without the QA bands.
  return image.updateMask(mask).divide(10000)
      .select("B[0-9]*")
      .copyProperties(image, ["system:time_start"]);
}
```

Extract Images from the Landsat 8 Collection and Calculate NDVI

18. Extract the images from the Landsat 8 Surface Reflectance (SR) Tier 1 Collection:

```
// Extract the images from the Landsat8 collection
var collectionl8 = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')
.filterDate('2019-08-01', '2019-08-10')
.filterBounds(roi)
.map(maskL8sr);
```

19. Calculate NDVI and add it as an extra band to the Landsat image selected:

```
//Calculate NDVI and create an image that contains all Landsat 8 bands and NDVI
var comp = collectionl8.mean();
var ndvi = comp.normalizedDifference(['B5', 'B4']).rename('NDVI');
var composite = ee.Image.cat(comp,ndvi);
```

Add the Images to Layers

<u>Add the images to "Layers" in order to display them</u>

20. Add the Landsat image to the "layers" bar in order to then visualize the images. Click enter or return and add the below code:

```
// Add images to layers in order to display them
Map.centerObject(roi, 7);
Map.addLayer(composite, {bands: ['B4', 'B3', 'B2'], min: 0, max: 0.2}, 'Optical');
```

Apply a Speckle Filter to the SAR Data and Display the Images

21. Apply a speckle filter.

Click enter or returnand add the below code:

```
//Apply filter to reduce speckle
var SMOOTHING_RADIUS = 50;
var SARVV_filtered = SARVV.focal_mean(SMOOTHING_RADIUS, 'circle', 'meters');
var SARVH_filtered = SARVH.focal_mean(SMOOTHING_RADIUS, 'circle', 'meters');
```

22. Add the speckle filtered images to the “layers” bar and display them.

```
//Display the SAR filtered images
Map.addLayer(SARVV_filtered, {min:-15,max:0}, 'SAR VV Filtered',0);
Map.addLayer(SARVH_filtered, {min:-25,max:0}, 'SAR VH Filtered',0);
```

Select Training Data

23. The first step in running a supervised classification is to collect training data to "train" the classifier.

- This involves collecting representative samples of backscatter for each landcover class of interest.
- Display the after VH image and go to the **Geometry Imports** box next to the geometry drawing tools and click **+ new layer**.
- Next to it select the draw a **polygon** icon.
- Each new layer represents one class within the training data, for example **open_water**.

Select Training Data

24. Define the first new layer as **open_water**.

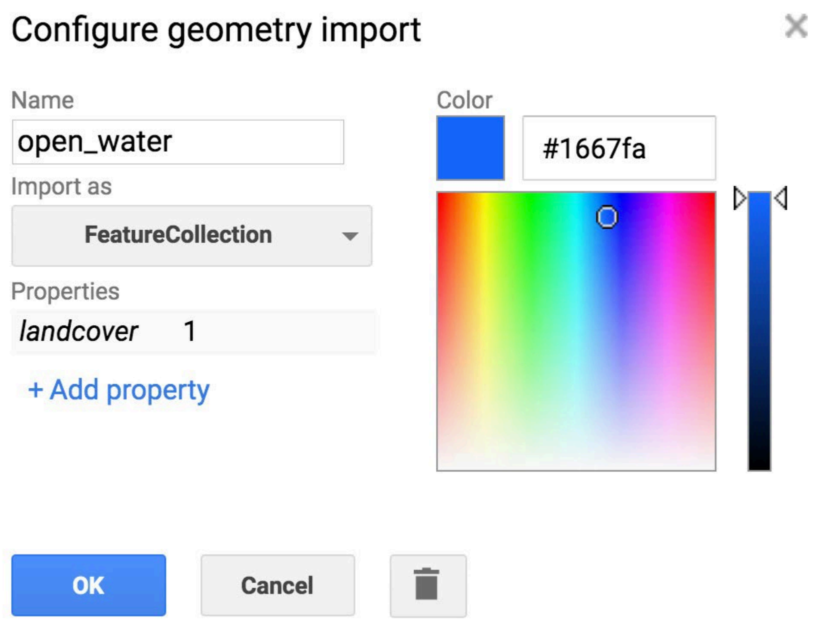
25. Locate open water areas along the river in the new layer and click to collect them.

26. Collect a representative sample of polygons and rename the **geometry** as **open_water**.

27. Configure the open_water geometry import (cog-wheel, top of the script in imports section).

28. Click the cog-wheel icon to configure it, change **Import as** from **Geometry** to **FeatureCollection**.

29. Use **Add property** landcover and set its value to 1. Subsequent classes will be 2, 3, 4 etc.). Change the color to blue, when finished, click **OK**.

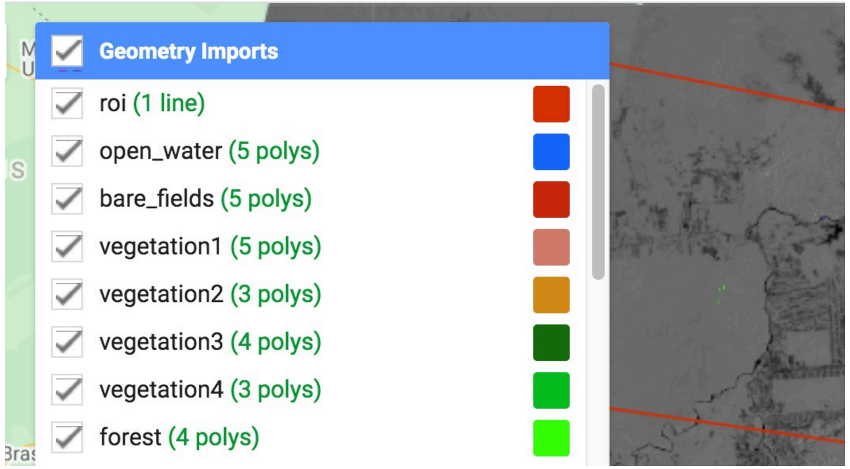


Merge the Defined Classes

Identify seven classes total. The next step is to merge them into a single collection, called a FeatureCollection.

30. Run the following line to merge the geometries into a single FeatureCollection:

```
//Merge Feature Collections
var newfc =
open_water.merge(bare_fields).merge(vegetation1).merge(vegetation2)
.merge(vegetation3).merge(vegetation4).merge(forest);
```



Classify the SAR Image Only

Create Training Data from Sentinel-1

We will use the FeatureCollection created to extract backscatter values for each landcover class identified for the Sentinel-1 image to be used in the classification.

31. The training data is created by overlaying the training points on the image.

```
//Define the SAR bands to train your data
var final = ee.Image.cat(SARVV_filtered,SARVH_filtered);
var bands = ['VH','VV'];
var training = final.select(bands).sampleRegions({
  collection: newfc,
  properties: ['landcover'],
  scale: 30 });
```

Train the Classifier

32. Train the Random Forest classifier.

```
//Train the classifier
var classifier = ee.Classifier.randomForest().train({
  features: training,
  classProperty: 'landcover',
  inputProperties: bands
});
```

Run the Classifier and Display the Results

33. Run the classifier by applying the knowledge from our training areas to the rest of the image:

```
//Run the Classifier
var classified = final.select(bands).classify(classifier);
```

34. Display the results using the code below. The colors may need to be adjusted, however, if colors and numbers have been assigned to the training data, the result will be rendered with those class numbers and colors.

```
//Display the Classification
Map.addLayer(classified,
{min: 1, max: 7, palette: ['1667fa', 'c9270d', 'cf7b68', 'ee9a1c', '146d0e', '04bd23',
'37fe05']},
'SAR Classification');
```

Classification Accuracy

35. Create a confusion matrix and calculate the accuracy of the results.

– Here were are only looking at the training area accuracy, which describes how well the classifier was able to correctly label resubstituted training data.
– For true validation accuracy, we need to use new 'testing' data.

```
RF- SAR error matrix:                    JSC
List (8 elements)                        JSC
 0: [0,0,0,0,0,0,0,0]
 1: [0,2293,1,0,0,0,0,0]
 2: [0,0,1937,32,0,0,0,0]
 3: [0,0,46,2853,188,9,10,1]
 4: [0,0,0,175,2735,251,1,0]
 5: [0,0,0,17,246,2452,132,0]
 6: [0,0,0,4,1,139,2706,273]
 7: [0,0,0,0,0,0,275,4744]

RF- SAR accuracy:                        JSC
0.916314297662748
```

```
// Create a confusion matrix representing resubstitution accuracy.
print('RF- SAR error matrix: ', classifier.confusionMatrix());
print('RF- SAR accuracy: ', classifier.confusionMatrix().accuracy());
```

Classify the Landsat Image Only

Create Training Data from Landsat

We will use the FeatureCollection created to extract reflectance values for each landcover class identified for the Landsat 8 image to be used in the classification.

36. The training data is created by overlaying the training points on the image.

```
//Define the Landsat bands to train your data
var bandsl8 = ['B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B10', 'B11', 'NDVI' ];
//var bandsl8 = ['NDVI' ];
var trainingl8 = composite.select(bandsl8).sampleRegions({
  collection: newfc,
  properties: ['landcover'],
  scale: 30
});
```

Train the Classifier

37. Train the Random Forest classifier.

```
//Train the classifier
var classifierl8 = ee.Classifier.randomForest().train({
  features: trainingl8,
  classProperty: 'landcover',
  inputProperties: bandsl8
});
```

Run the Classifier and Display the Results

38. Run the classifier by applying the knowledge from our training areas to the rest of the image:

```
//Run the Classifier
var classifiedl8 = composite.select(bandsl8).classify(classifierl8);
```

39. Display the results using the mapping function below. The colors may need to be adjusted, however, if colors and numbers have been assigned to the training data, the result will be rendered with those class numbers and colors.

```
//Display the Classification
Map.addLayer(classifiedl8,
{min: 1, max: 7, palette: ['1667fa', 'c9270d', 'cf7b68', 'ee9a1c', '146d0e', '04bd23', '37fe05']},
'Optical Classification');
```

Classification Accuracy

40. Create a confusion matrix and calculate the accuracy of the results.

– Here were are only looking at the training area accuracy, which describes how well the classifier was able to correctly label resubstituted training data.
– For true validation accuracy, we need to use new ‘testing’ data.

```
RF-L8 error matrix:                    JSON
List (8 elements)                      JSON
  0: [0,0,0,0,0,0,0,0]
  1: [0,2294,0,0,0,0,0,0]
  2: [0,0,1921,4,15,15,14,0]
  3: [0,0,9,2951,55,46,42,4]
  4: [0,0,24,72,2929,91,46,0]
  5: [0,0,12,42,66,2676,51,0]
  6: [0,0,14,22,32,63,2992,0]
  7: [0,0,0,3,0,0,1,5015]

RF-L8 accuracy:                        JSON
0.9654755819896845
```

```
// Create a confusion matrix representing resubstitution accuracy.
print('RF-L8 error matrix: ', classifierl8.confusionMatrix());
print('RF-L8 accuracy: ', classifierl8.confusionMatrix().accuracy());
```

Classify Landsat and Sentinel-1

Create Training Data from Landsat and Sentinel-1

We will use the FeatureCollection created to extract reflectance, NDVI, and backscatter values for each landcover class identified for the Landsat 8 and Sentinel-1 images to be used in the classification.

41. The training data is created by overlaying the training points on the image.

```
//Define both optical and SAR to train your data
var opt_sar = ee.Image.cat(composite, SARVV_filtered,SARVH_filtered);
var bands_opt_sar = ['VH','VV','B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B10', 'B11', 'NDVI'];
var training_opt_sar = opt_sar.select(bands_opt_sar).sampleRegions({
  collection: newfc,
  properties: ['landcover'],
  scale: 30 });
```

Train the Classifier

42. Train the Random Forest classifier.

```
//Train the classifier
var classifier_opt_sar = ee.Classifier.randomForest().train({
  features: training_opt_sar,
  classProperty: 'landcover',
  inputProperties: bands_opt_sar
});
```

Run the Classifier and Display the Results

43. Run the classifier by applying the knowledge from our training areas to the rest of the image:

```
//Run the classifier
var classifiedboth = opt_sar.select(bands_opt_sar).classify(classifier_opt_sar);
```

44. Display the results using the mapping function below. The colors may need to be adjusted, however, if colors and numbers have been assigned to the training data, the result will be rendered with those class numbers and colors.

```
//Display the Classification
Map.addLayer(classifiedboth,
{min: 1, max: 7, palette: ['1667fa', 'c9270d', 'cf7b68', 'ee9a1c', '146d0e', '04bd23', '37fe05']},
'Optical/SAR Classification');
```

Classification Accuracy

45. Create a confusion matrix and calculate the accuracy of the results.

- Here were are only looking at the training area accuracy, which describes how well the classifier was able to correctly label resubstituted training data.
- For true validation accuracy, we need to use new 'testing' data.

```
RF-Opt/SAR error matrix:                JSON
▾List (8 elements)                      JSON
  ▸0: [0,0,0,0,0,0,0,0]
  ▸1: [0,2294,0,0,0,0,0,0]
  ▸2: [0,0,1959,6,4,0,0,0]
  ▸3: [0,0,4,3028,63,10,1,1]
  ▸4: [0,0,5,52,3024,71,10,0]
  ▸5: [0,0,0,3,97,2705,42,0]
  ▸6: [0,0,0,5,3,49,3066,0]
  ▸7: [0,0,0,2,0,0,0,5017]

RF-Opt/SAR accuracy:                    JSON
0.9801124483063055
```

```
// Create a confusion matrix representing resubstitution accuracy.
print('RF-Opt/SAR error matrix: ', classifier_opt_sar.confusionMatrix());
print('RF-Opt/SAR accuracy: ', classifier_opt_sar.confusionMatrix().accuracy());
```

Export the Result as GeoTIFF

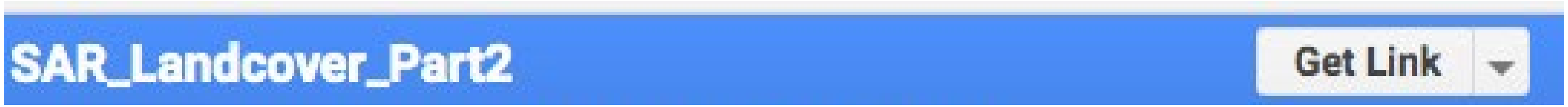
46. Export your classification as a GeoTIFF to your Google Drive
Click enter and add the code below:

```
// Export the image, specifying scale and region.
 Export.image.toDrive({
   image: classifiedboth,
  description: 'Optical_Radar',
   scale: 100,
    fileFormat: 'GeoTIFF',
 });
```

47. Click on **Run** in the top menu.

Save your Forest Change Code

52. Along the top panel, click on Save and save your code as: **SAR_Landcover_Part2** to your directory.



Google Earth Engine Optical/Radar Classification Demo Code

https://code.earthengine.google.com/283acb2ceedee98e77b20ef315b2fab7?accept_repo=users%2Fwolterpt%2FSAR_TimeSeries_PTW

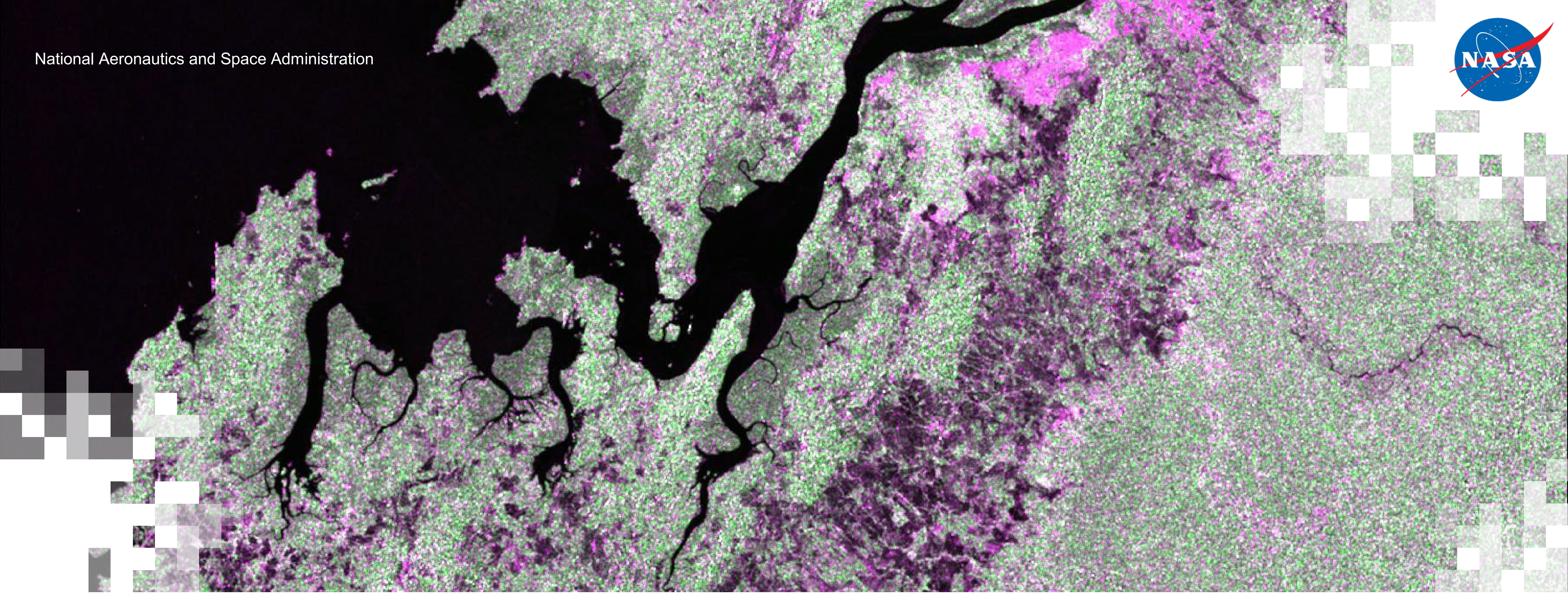
Exercise Summary

- In this demo we identified Sentinel-1 and Landsat-8 images for a given area and applied a supervised classification - Random Forest.
- A classification was generated for the Sentinel-1 image, the Landsat-8 image, and both optical and radar images.
- A confusion matrix and accuracy results were generated for each classification

Contacts

- ARSET Contacts
  - Erika Podest: Erika.Podest@jpl.nasa.gov
  - Amber McCullum: AmberJean.Mccullum@nasa.gov
  - Juan Torres-Perez: juan.l.torresperez@nasa.gov
- General ARSET Inquiries
  - Ana Prados: aprados@umbc.edu
- ARSET Website:
  - http://arset.gsfc.nasa.gov

National Aeronautics and Space Administration



Next Session: Mangrove Mapping

May 19, 2020

Questions

- Please enter your questions into the chat box.
- We will post the questions and answers to the training website following the conclusion of the course.

Thank You!