GEO 642: Advanced Remote Sensing and GIS

Week 8: Applications of Remote Sensing – Wildfire Mapping

Slides are available at:
https://tinyurl.com/GEEDemoSlides

Week 8 - Learning Objectives:

We will discuss how to apply the theory from previous lectures to mapping the impacts of wildfire.

Outline:

- We will review key topics:
  - Thermal Imaging (Week 6)
  - Atmospheric Remote Sensing (Week 7)
  - Spectral Signatures and False Colour Composites (Week 3)
- We will apply the theoretical concepts using satellite datasets:
  - Thermal: MODIS
  - Atmospheric: TROPOMI (Sentinel 5P)
  - Optical: Landsat
- And a cloud computing platform:
  - Google Earth Engine
  - Register for an account: code.earthengine.google.com

Week 8 - Learning Objectives:

By the end of today, you will be able to perform the following tasks in Google Earth Engine:

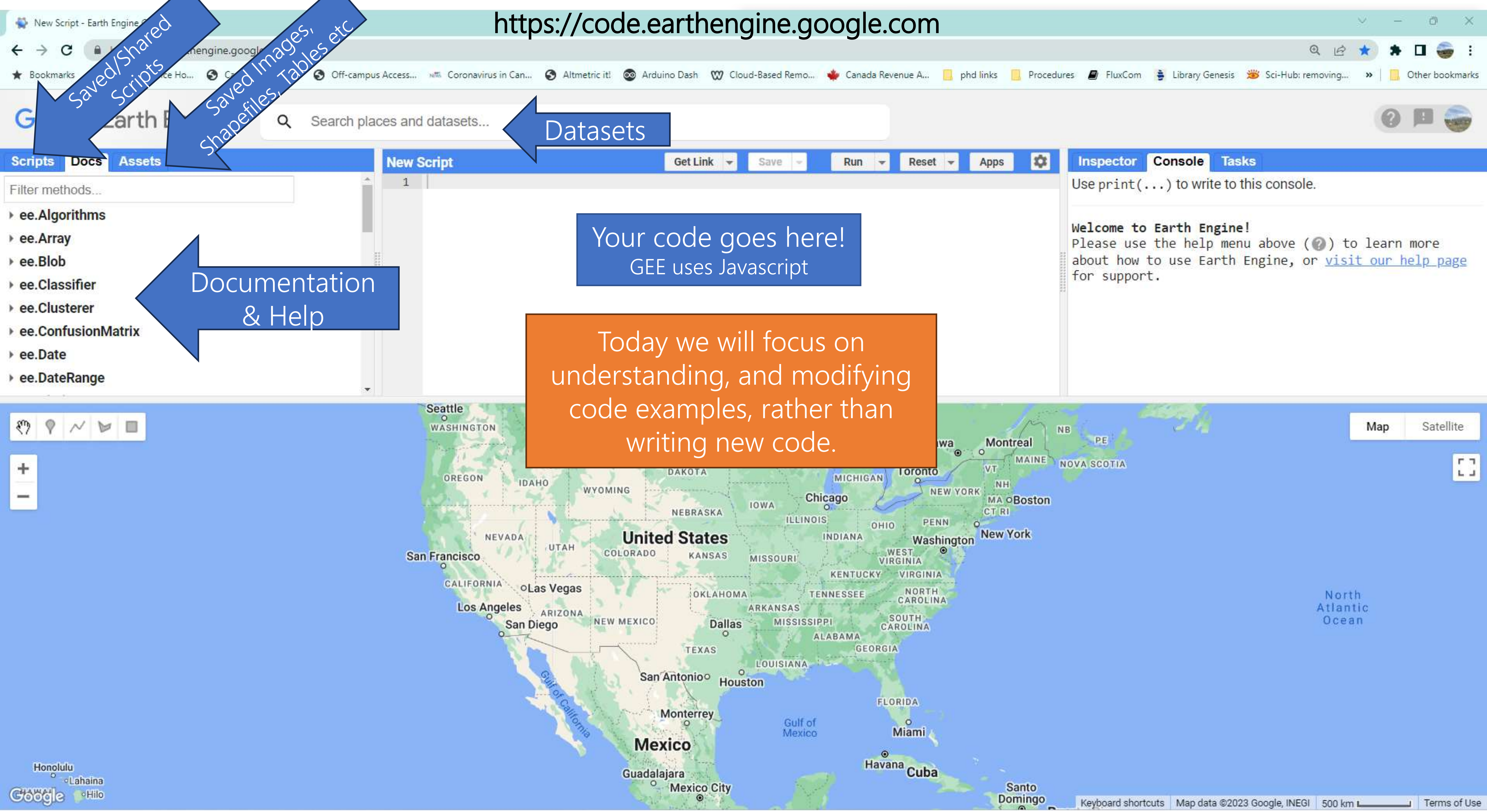
- Import data for analysis
- Filter image collections (raster data) and feature collections (vector data) by criteria such as:
  - Properties
  - Date
  - Image bands
- Mask cloudy and snowy pixels from analysis of Landsat data
- Calculate Vegetation Indices
- Customize image visualizations
- Export images to Google Drive for offline analysis (e.g. in ArcGIS Pro)

Why learn Google Earth Engine?

- Cloud-based geospatial analysis platform
- Petabyte-scale data catalog with >900 datasets
- Preprocessing and analysis at global scale performed on Google’s servers/computing resources

Allows for global scale analysis with no need for large downloads, or local computing resources!

code.earthengine.google.com


https://code.earthengine.google.com
Saved/Shared Scripts
Saved Images, Shapefiles, Tables etc
Datasets
Documentation & Help
Your code goes here!
GEE uses Javascript
Today we will focus on understanding, and modifying code examples, rather than writing new code.
Search places and datasets...
Scripts
Docs
Assets
Filter methods...
ee.Algorithms
ee.Array
ee.Blob
ee.Classifier
ee.Clusterer
ee.ConfusionMatrix
ee.Date
ee.DateRange
New Script
Get Link
Save
Run
Reset
Apps
Inspector
Console
Tasks
Use print(...) to write to this console.
Welcome to Earth Engine!
Please use the help menu above (?) to learn more about how to use Earth Engine, or visit our help page for support.
Map
Satellite
United States
Mexico
North Atlantic Ocean
Gulf of Mexico
Keyboard shortcuts
Map data ©2023 Google, INEGI
500 km
Terms of Use

What happens <u>on the ground</u> during wildfire?

USDA Forest Service (Public Domain)

- Heat
- Smoke
- The Vegetation Burns
  - Initial vegetation removal
  - Post-fire recovery

What can we see from Space?

Mike Lewelling National Park Service (CC-BY-2.0)

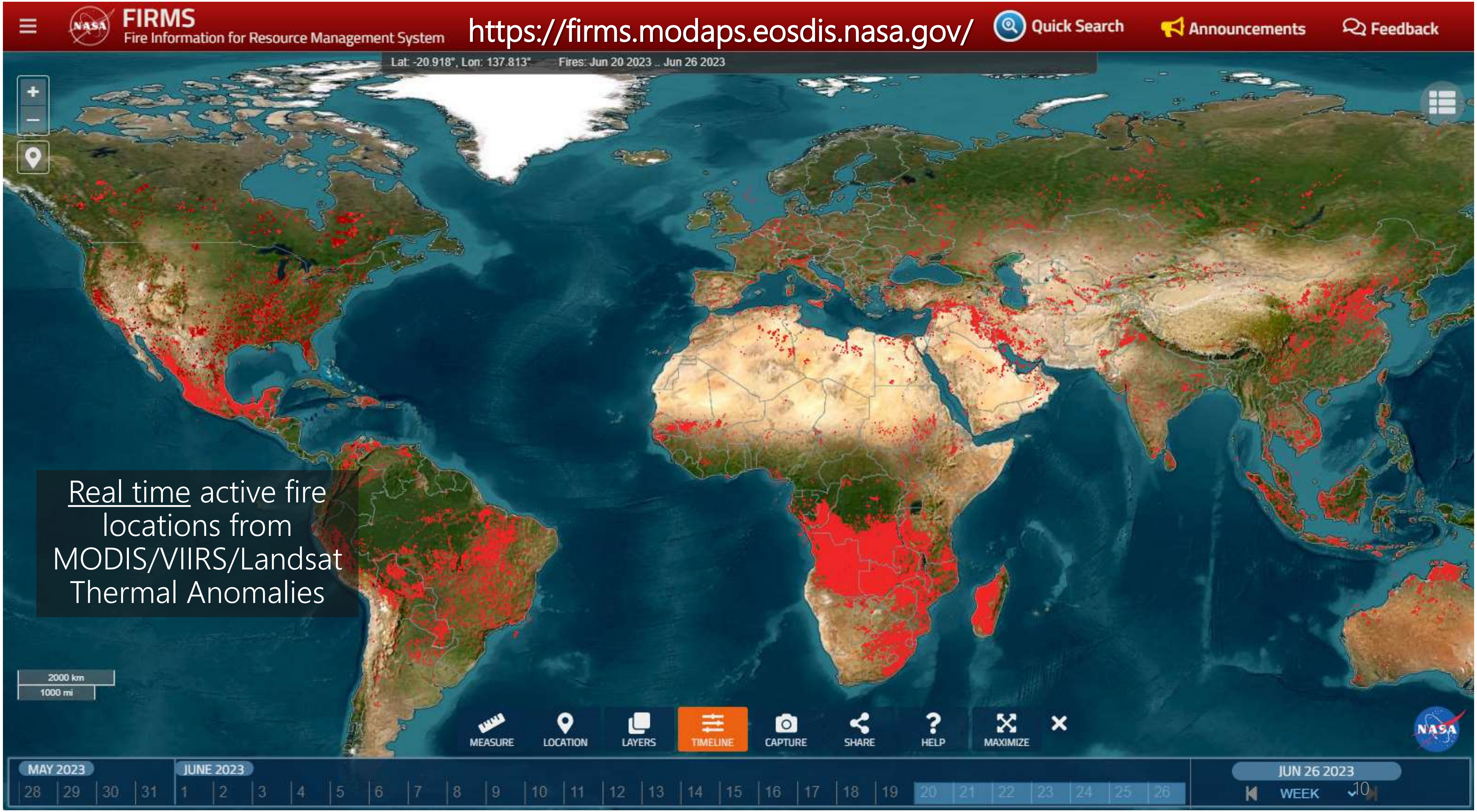
Jeffrey Beall (CC-BY-4.0)

What can we see from Space?

What can we see from Space?
1) Heat

Review from Week 6:

- Thermal Infrared
  - Is emitted from the earth's surface (and NOT reflected sunlight)
  - Wavelength range: 3 μm – 1 mm
  - A hotter object will emit MORE radiation
  - The emission spectrum (peak wavelength) depends on the object temperature
    - The sun's emission spectrum is in the visible part of the spectrum because the sun is HOT
    - Hotter objects emit radiation at shorter wavelengths



FIRMS
Fire Information for Resource Management System
https://firms.modaps.eosdis.nasa.gov/
Quick Search
Announcements
Feedback
Lat: -20.918°, Lon: 137.813°
Fires: Jun 20 2023 .. Jun 26 2023
Real time active fire locations from MODIS/VIIRS/Landsat Thermal Anomalies
2000 km
1000 mi
MEASURE
LOCATION
LAYERS
TIMELINE
CAPTURE
SHARE
HELP
MAXIMIZE
MAY 2023
JUNE 2023
28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
JUN 26 2023
WEEK

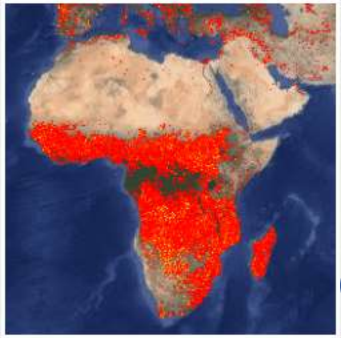
https://developers.google.com/earth-engine/datasets/catalog/FIRMS

Earth Engine Data Catalog

Search

Home | View all datasets | Browse by tags | Landsat | MODIS | Sentinel | API Docs

FIRMS: Fire Information for Resource Management System



Dataset Availability

2000-11-01T00:00:00Z–2023-06-24T00:00:00

Dataset Provider

NASA / LANCE / EOSDIS

Earth Engine Snippet

`ee.ImageCollection("FIRMS")`

How we access the data in Earth Engine

Tags

fire | geophysical | hotspot | modis | nasa | thermal | eosdis | firms | lance

Description | Bands | Terms of Use | Citations

The Earth Engine version of the Fire Information for Resource Management System (FIRMS) dataset contains the LANCE fire detection product in rasterized form. The near real-time (NRT) active fire locations are processed by LANCE using the standard MODIS MOD14/MYD14 Fire and Thermal Anomalies product. Each active fire location represents the centroid of a 1km pixel that is flagged by the algorithm as containing one or more fires within the pixel. The data are rasterized as follows: for each FIRMS active fire point, a 1km bounding box (BB) is defined; pixels in the MODIS sinusoidal projection that intersect the FIRMS BB are identified; if multiple FIRMS BBs intersect the same pixel, the one with higher confidence is retained; in case of a tie, the brighter one is retained.

The data in the near-real-time dataset are not considered to be of science quality.

Additional information can be found here.

Explore in Earth Engine

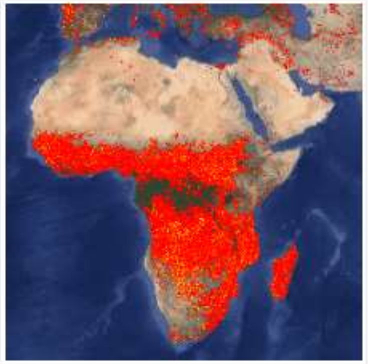
Example Code:

Review the Exam

FIRMS *

```
var dataset = ee.ImageCollection('
.filter(
    ee.Filter.date('2018-08-01', '
    );
var fires = dataset.select('T21');
var firesVis = { //sets up visuali
  min: 325.0,
  max: 400.0,
  palette: ['red', 'orange', 'yell
};
Map.setCenter(-119.086, 47.295, 6)
Map.addLayer(fires, firesVis, 'Fir

```

FIRMS: Fire Information for Resource Management System



Dataset Availability

2000-11-01T00:00:00Z–2023-06-24T00:00:00

Dataset Provider

NASA / LANCE / EOSDIS

Earth Engine Snippet

`ee.ImageCollection("FIRMS")`

Tags

fire geophysical hotspot modis nasa thermal eosdis firms lance

Description | Bands | Terms of Use | Citations

Resolution

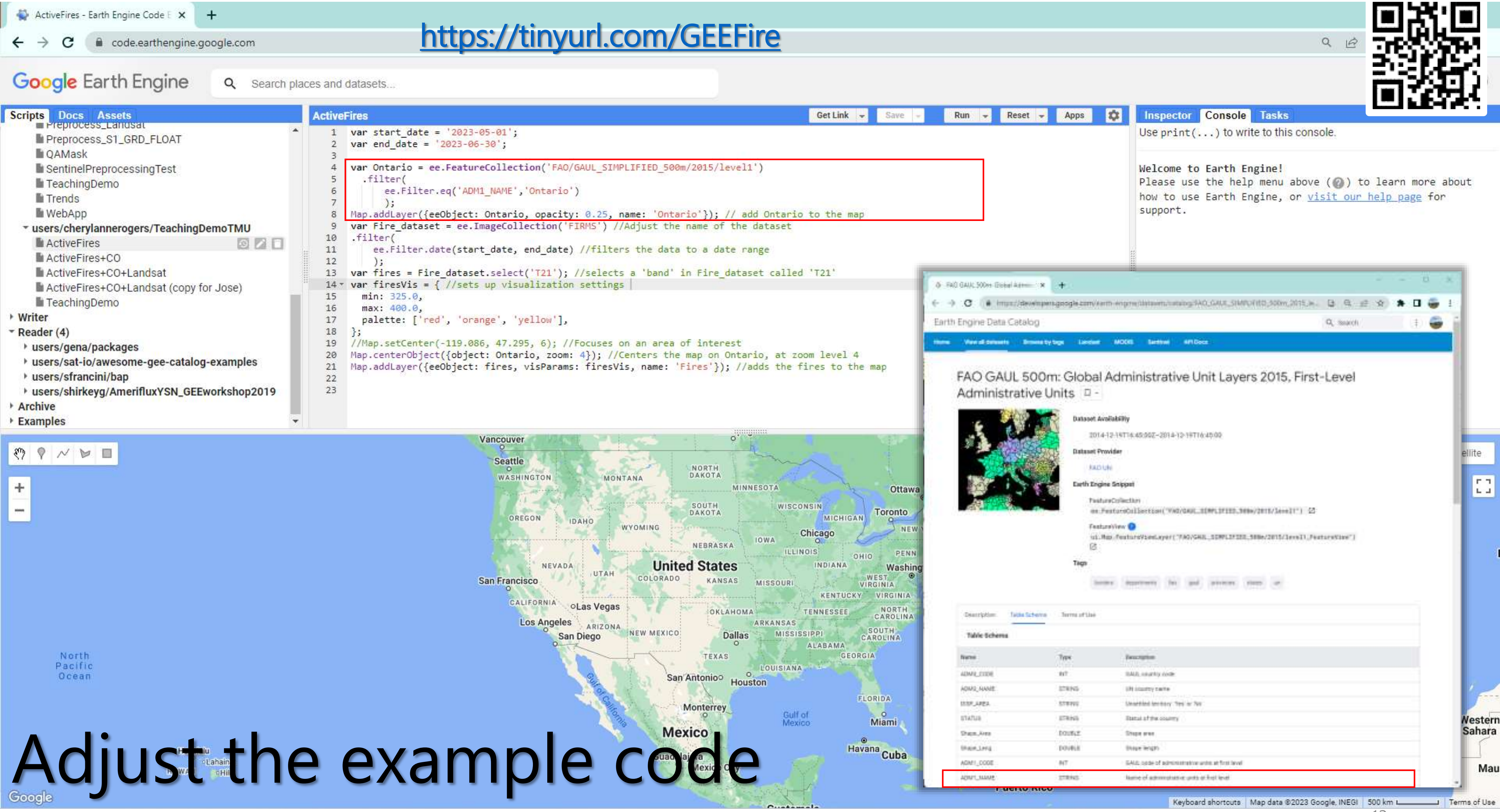
1000 meters

Bands

Name	Units	Min	Max	Description
T21	K	300*	509.29*	The brightness temperature of a fire pixel using MODIS channels 21/22.
confidence	%	0	100	A detection confidence intended to help users gauge the quality of individual active fire pixels. The confidence estimate ranges between 0% and 100% for all fire pixels within the fire mask. The confidence field should be used with caution; it is likely that it will vary in meaning in different parts of the world.
line_ number		1*	35302*	Line number in the FIRMS CSV file that the pixel came from.

* estimated min or max value

https://tinyurl.com/GEEFire



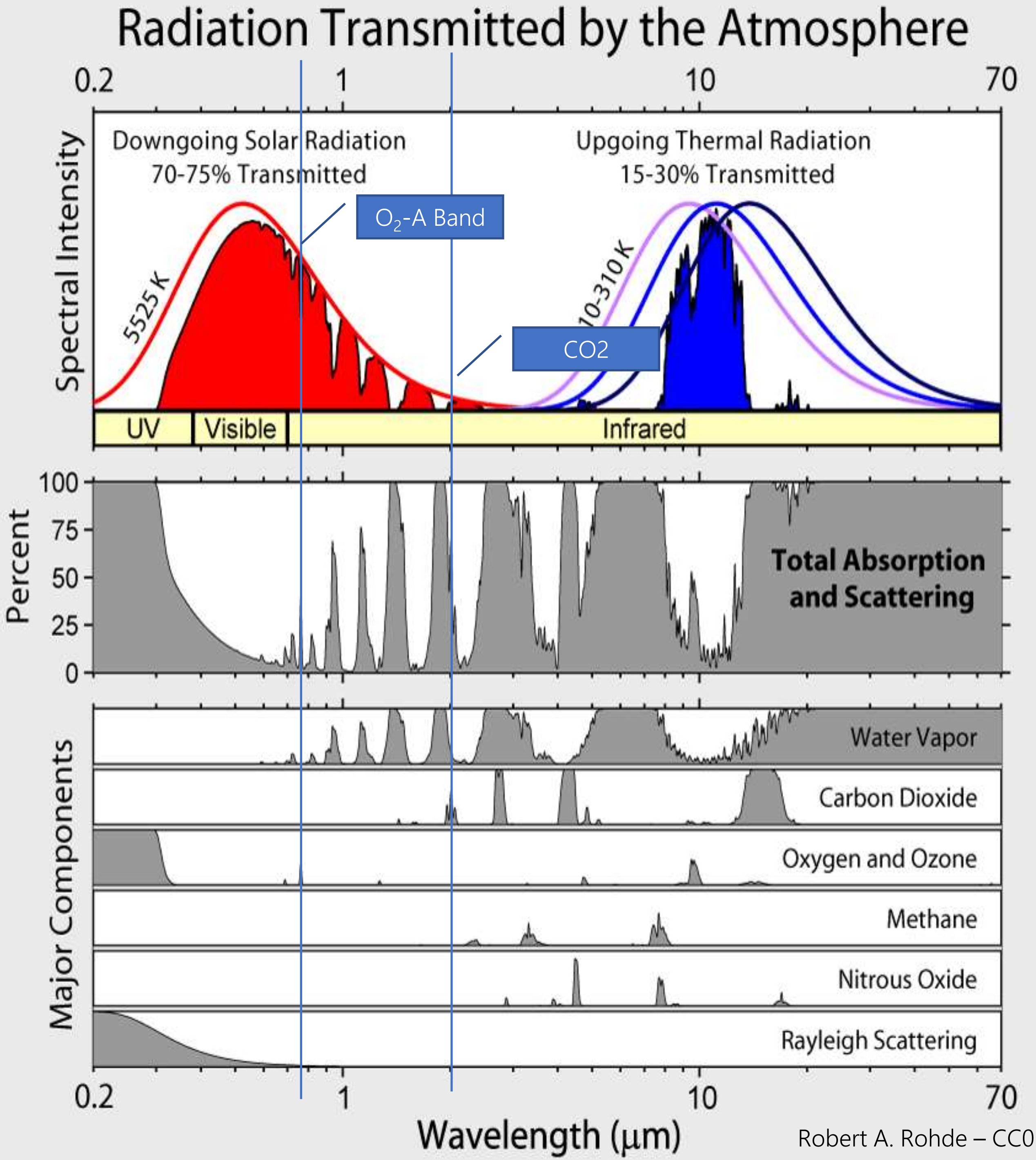
Adjust the example code

What can we see from Space?
2) Smoke

- Review from Week 7:
- Atmospheric Remote Sensing of Trace Gases
  - Trace gases absorb light at specific wavelengths
  - The amount of absorption that occurs as light passes through the atmosphere relates to the amount of that trace gas in the atmosphere

What trace gas datasets are available on Google Earth Engine?

Which are related to wildfire?



Robert A. Rohde – CC0

“TROPOMI” onboard Sentinel-5P

https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S5P_NRTI_L3_CO

How should we adjust the example code?
What does each line do?

COPERNICUS_S5P_NRTI_L3_CO * | Get Link | Save | Run | Reset | Apps

```
1  var collection = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_CO')
2    .select('CO_column_number_density')
3    .filterDate(start_date, end_date); //filters the collection of images by date
4
5  var band_viz = { //sets up visualization settings (this is used in line 11)
6    min: 0,
7    max: 0.05,
8    palette: ['black', 'blue', 'purple', 'cyan', 'green', 'yellow', 'red']
9  };
10
11 Map.addLayer(collection.mean(), band_viz, 'S5P CO'); //.mean() computes the mean value for each pixel
12 // .mean() calculates an image from a collection of images and is called a 'reducer' in GEE
13 Map.setCenter(-25.01, -4.28, 4); //centers the map on -25.01N -4.28E at zoom level 4
14
```

COPERNICUS_S5P_NRTI_L3_CO *

Get Link | Save | Run | Reset | Apps

```
1  var CO_collection = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_CO')
2    .select('CO_column_number_density')
3    .filterDate(start_date, end_date); //filters the collection of images by date
4
5  var band_viz = { //sets up visualization settings (this is used in line 11)
6    min: 0,
7    max: 0.05,
8    palette: ['black', 'blue', 'purple', 'cyan', 'green', 'yellow', 'red']
9  };
10
11 Map.addLayer(CO_collection.mean(), band_viz, 'S5P CO'); //.mean() computes the mean value for each pixel
12 // .mean() calculates an image from a collection of images and is called a 'reducer' in GEE
13
14
```

ActiveFires

Get Link | Save | Run | Reset | Apps

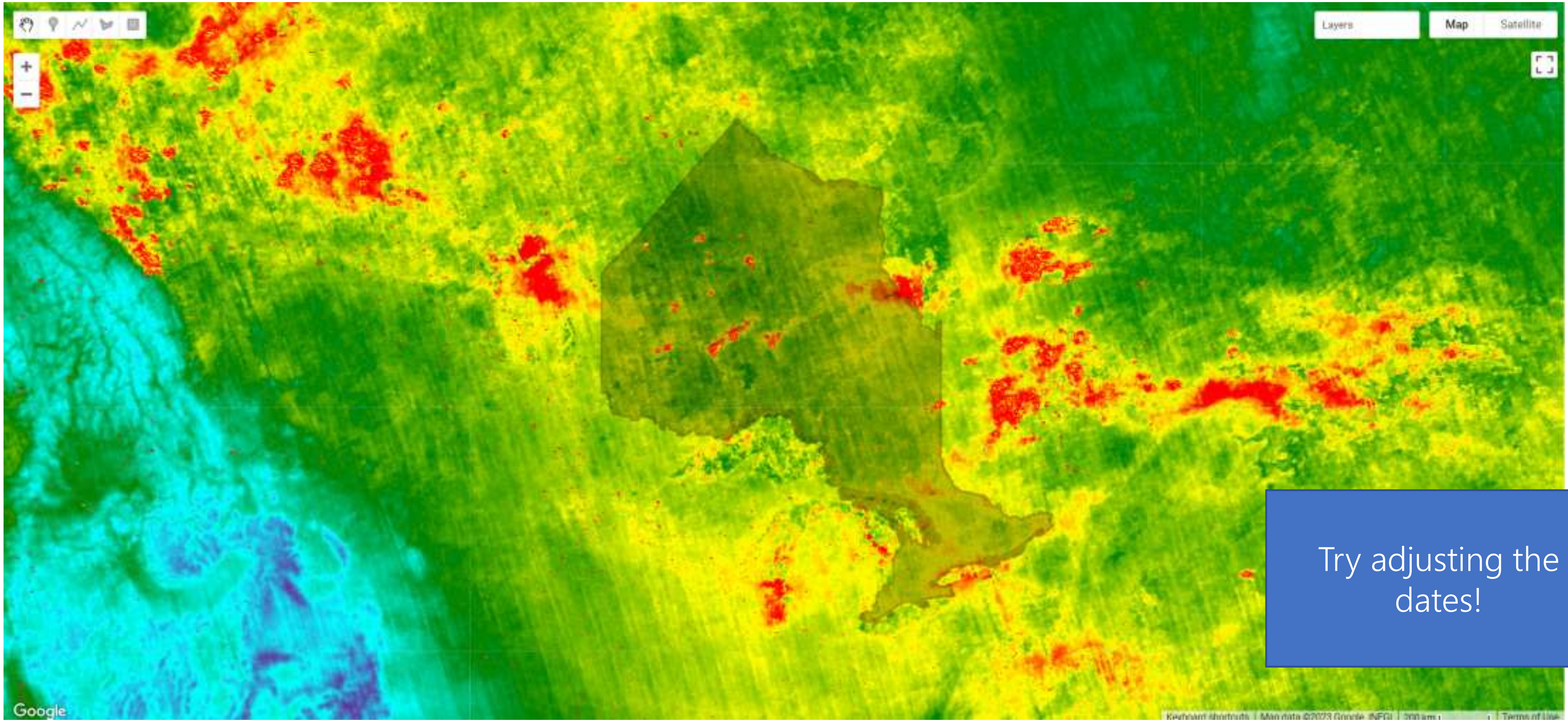
```
1  var start_date = '2023-05-01';
2  var end_date = '2023-06-30';
3
4  var Ontario = ee.FeatureCollection('FAO/GAUL_SIMPLIFIED_500m/2015/level1')
5    .filter(
6        ee.Filter.eq('ADM1_NAME','Ontario')
7        );
8  Map.addLayer({eeObject: Ontario, opacity: 0.25, name: 'Ontario'}); // add Ontario to the map
9  var Fire_dataset = ee.ImageCollection('FIRMS') //Adjust the name of the dataset
10 .filter(
11     ee.Filter.date(start_date, end_date) //filters the data to a date range
12     );
13 var fires = Fire_dataset.select('T21'); //selects a 'band' in Fire_dataset called 'T21'
14 var firesVis = { //sets up visualization settings (this is used later in line 12)
15   min: 325.0,
     max: 400.0,
     palette: ['red', 'orange', 'yellow'],
   };
   //Map.setCenter(-119.086, 47.295, 6); //Focuses on an area of interest
   Map.centerObject({object: Ontario, zoom: 4}); //Centers the map on Ontario, at zoom level 4
   Map.addLayer({eeObject: fires, visParams: firesVis, name: 'Fires'}); //adds the fires to the map
```

Let's add the new dataset to our script. Reorder the Map.addLayer() lines in the order we want things drawn https://tinyurl.com/GEEFireCO

ActiveFires+CO * | Get Link | Save | Run | Reset | Apps

```
//General Settings:
var start_date = '2023-05-01';
var end_date = '2023-06-30';

//Datasets
//Provincial Boundary
var Ontario = ee.FeatureCollection('FAO/GAUL_SIMPLIFIED_500m/2015/level1')
  .filter(
      ee.Filter.eq('ADM1_NAME','Ontario')
      );

//FIRMS Active Fires
var Fire_dataset = ee.ImageCollection('FIRMS') //Adjust the name of the dataset
.filter(
    ee.Filter.date(start_date, end_date) //filters the data to a date range
    );
var fires = Fire_dataset.select('T21'); //selects a 'band' in Fire_dataset called 'T21'

//Tropomi/Sentinel-5P Carbon Monoxide
var CO_collection = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_CO')
  .select('CO_column_number_density')
  .filterDate(start_date, end_date); //filters the collection of images by date

// Visualization settings
var firesVis = {
  min: 325.0,
  max: 400.0,
  palette: ['red', 'orange', 'yellow'],
};
var COviz = {
  min: 0,
  max: 0.05,
  palette: ['black', 'blue', 'purple', 'cyan', 'green', 'yellow', 'red']
};


//Add layers to the map:
Map.centerObject({object: Ontario, zoom: 4}); //Centers the map on Ontario, at zoom level 4

Map.addLayer(CO_collection.mean(), COviz, 'S5P CO'); //adds mean CO column density to the map
Map.addLayer({eeObject: Ontario, opacity: 0.5, name: 'Ontario'}); // add Ontario to the map
Map.addLayer({eeObject: fires, visParams: firesVis, name: 'Fires'}); //adds the fires to the map
```

- Run what we have so far:
- https://tinyurl.com/GEEFireCO

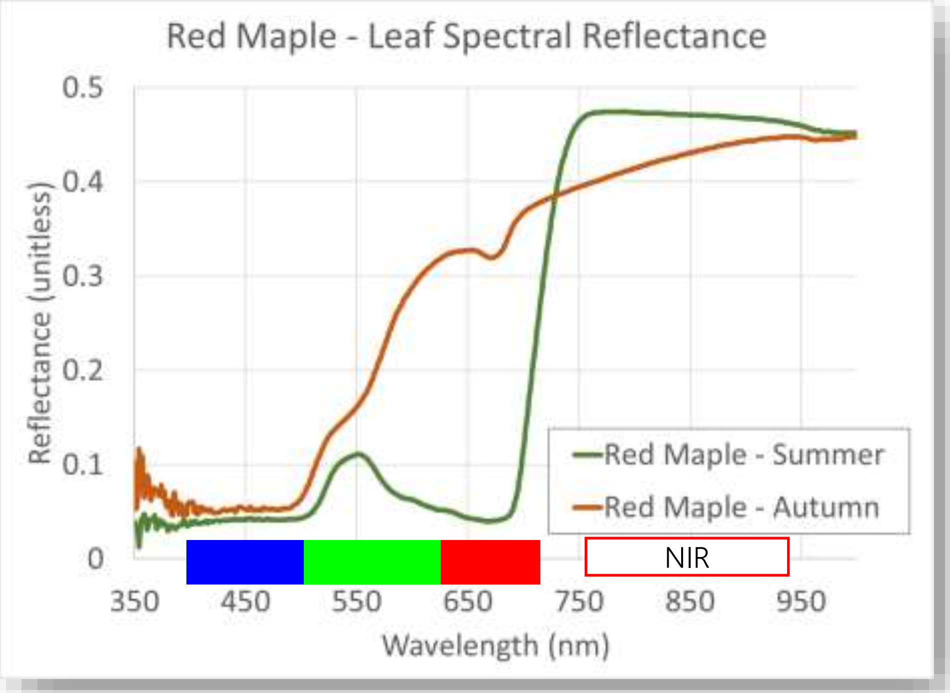


Try adjusting the dates!

What can we see from Space?
3) Burned Landscape

We can map **false colour composites** to highlight spectral differences

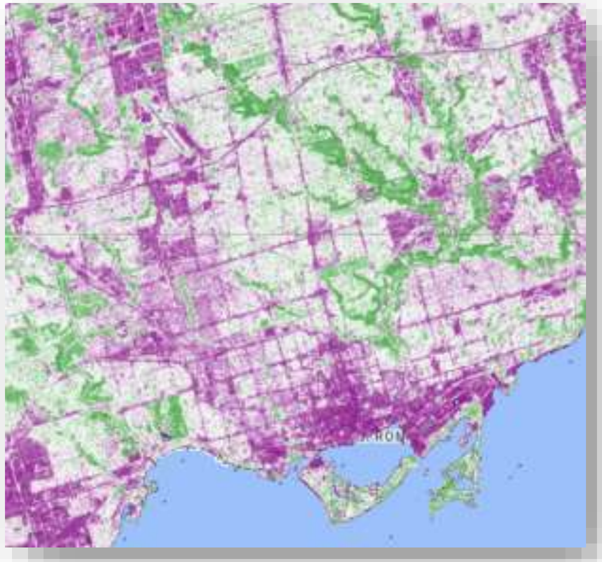
Review from week 3:



A pixel's surface reflectance is affected by the spectral signatures of objects within the image pixel

True Colour Landsat 9 Image
R= Red; G=Green; B=Blue

False Colour Landsat 9 Image
R = NIR; G = Red; B = Green

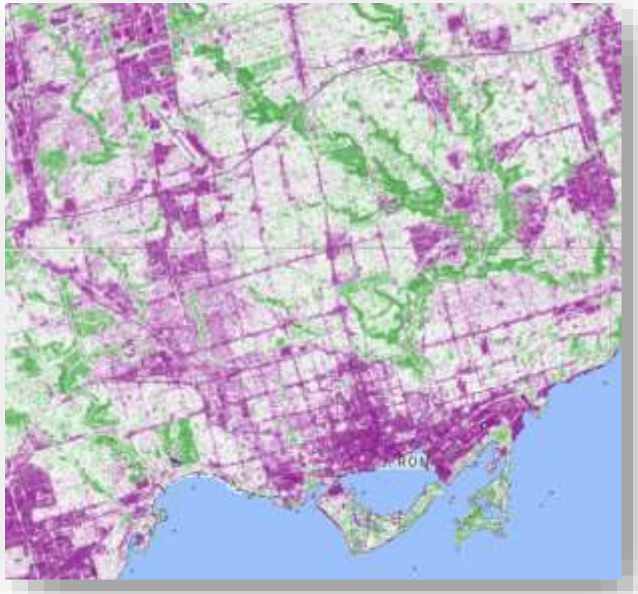


We can calculate indices to highlight spectral information:

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

What can we see from Space?
3) Burned Landscape

Purple: Low NDVI
Green: High NDVI



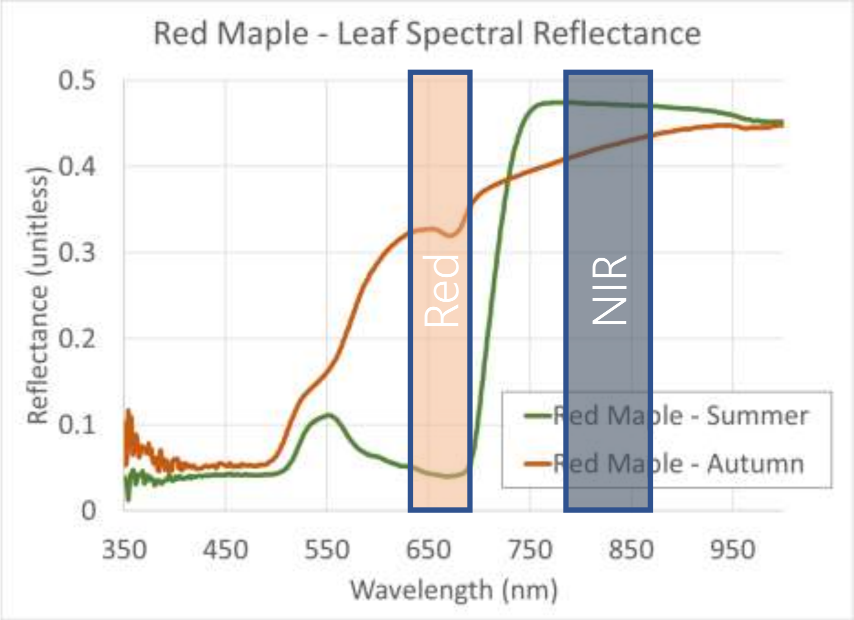
Review from week 3:

- ***Spectral Indices:*** combine pixel values from 2 bands (or more) to highlight spectral differences.
- ***NDVI:*** Normalized Difference Vegetation Index

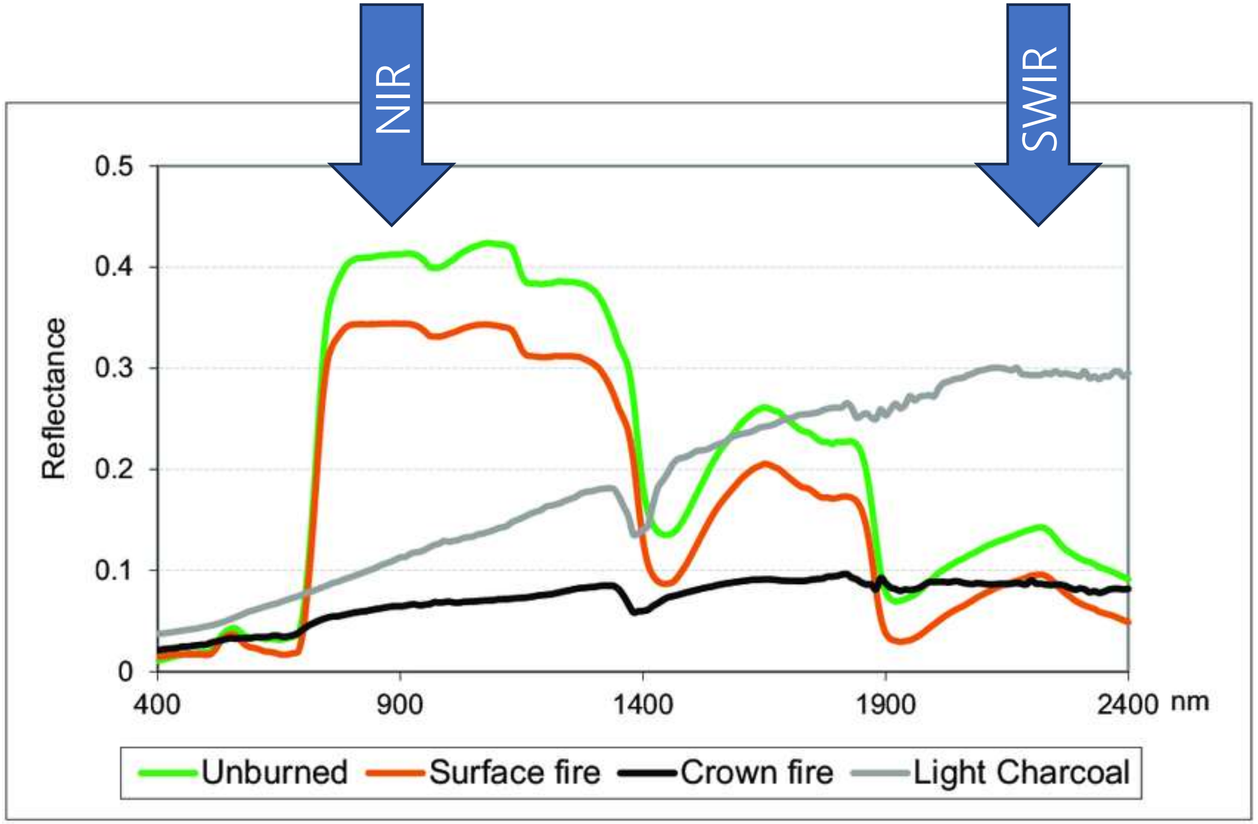
$$NDVI = \frac{NIR - Red}{NIR + Red}$$

$$Summer\ Leaf\ NDVI = \frac{0.48 - 0.05}{0.48 + 0.05} = \frac{0.43}{0.53} = 0.81$$

$$Autumn\ Leaf\ NDVI = \frac{0.43 - 0.33}{0.43 + 0.33} = \frac{0.1}{0.76} = 0.13$$



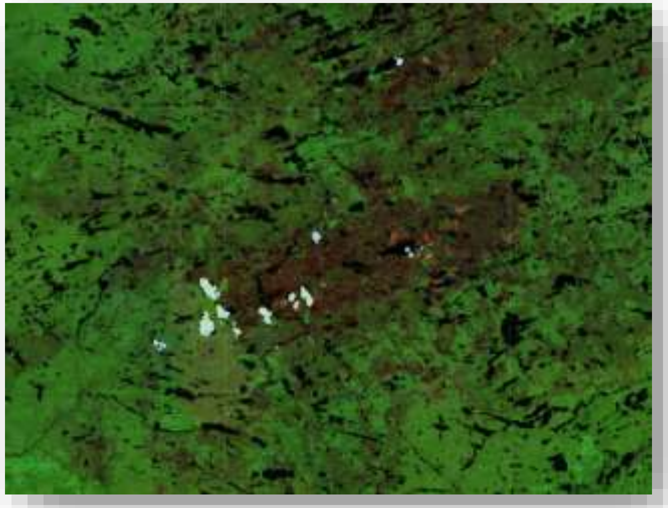
What can we see from Space?
3) Burned Landscape



Chuveico et al. 2019; DOI:10.1016/j.rse.2019.02.013

True Colour
R= Red; G=Green; B=Blue

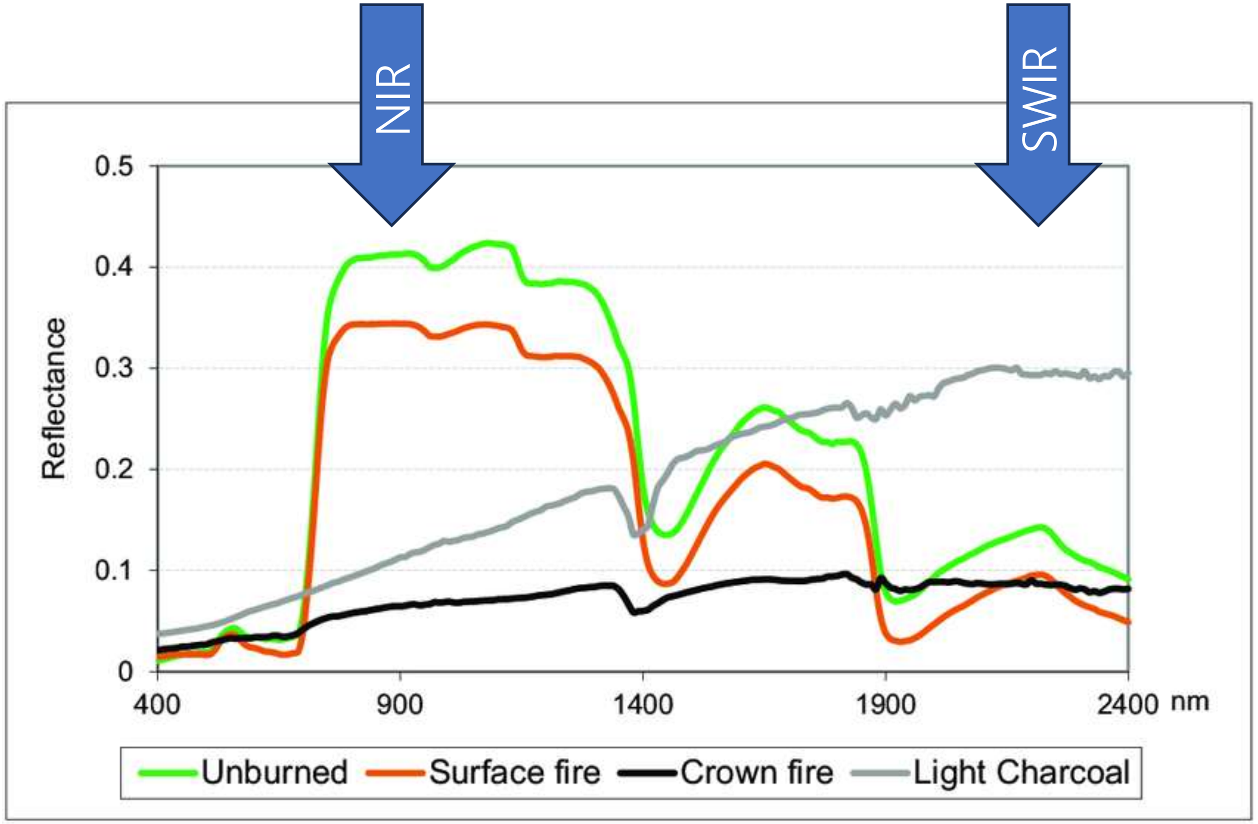
False Colour
R= SWIR; G=NIR; B=Red



Fire tends to reduce reflectance in the Near Infrared (NIR), and increase reflectance in the Shortwave Infrared (SWIR)

What can we see from Space?

3) Burned Landscape



Chuveico et al. 2019; DOI:10.1016/j.rse.2019.02.013

Normalized Burn Ratio:

$$NBR = \frac{NIR - SWIR}{NIR + SWIR}$$

Burn Category:	NIR	SWIR	NBR
Unburned:	0.41	0.14	0.49
Surface Fire:	0.34	0.08	0.62
Crown Fire:	0.06	0.08	-0.14
Charcoal:	0.10	0.30	-0.50

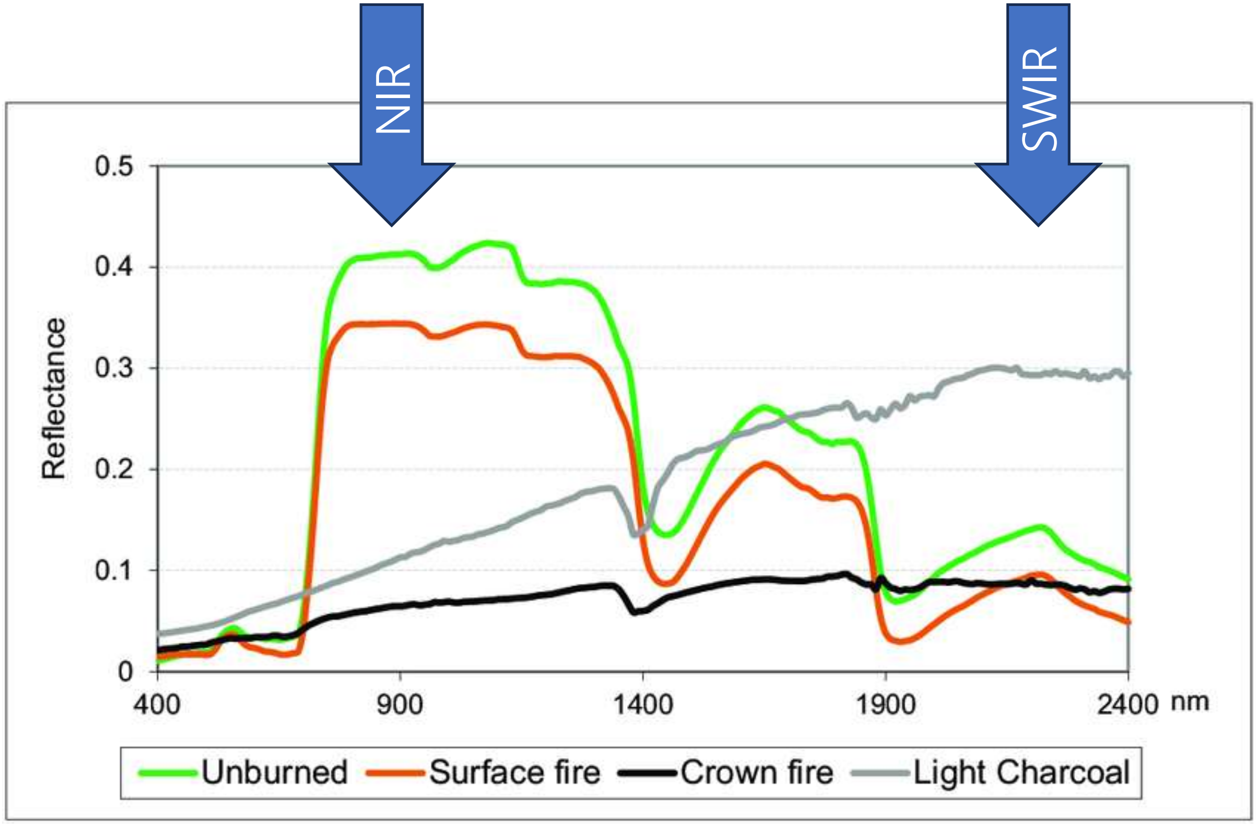
*NBR First proposed by López García and Caselles (1991) DOI:10.1080/10106049109354290

Fire tends to reduce reflectance in the Near Infrared (NIR), and increase reflectance in the Shortwave Infrared (SWIR)

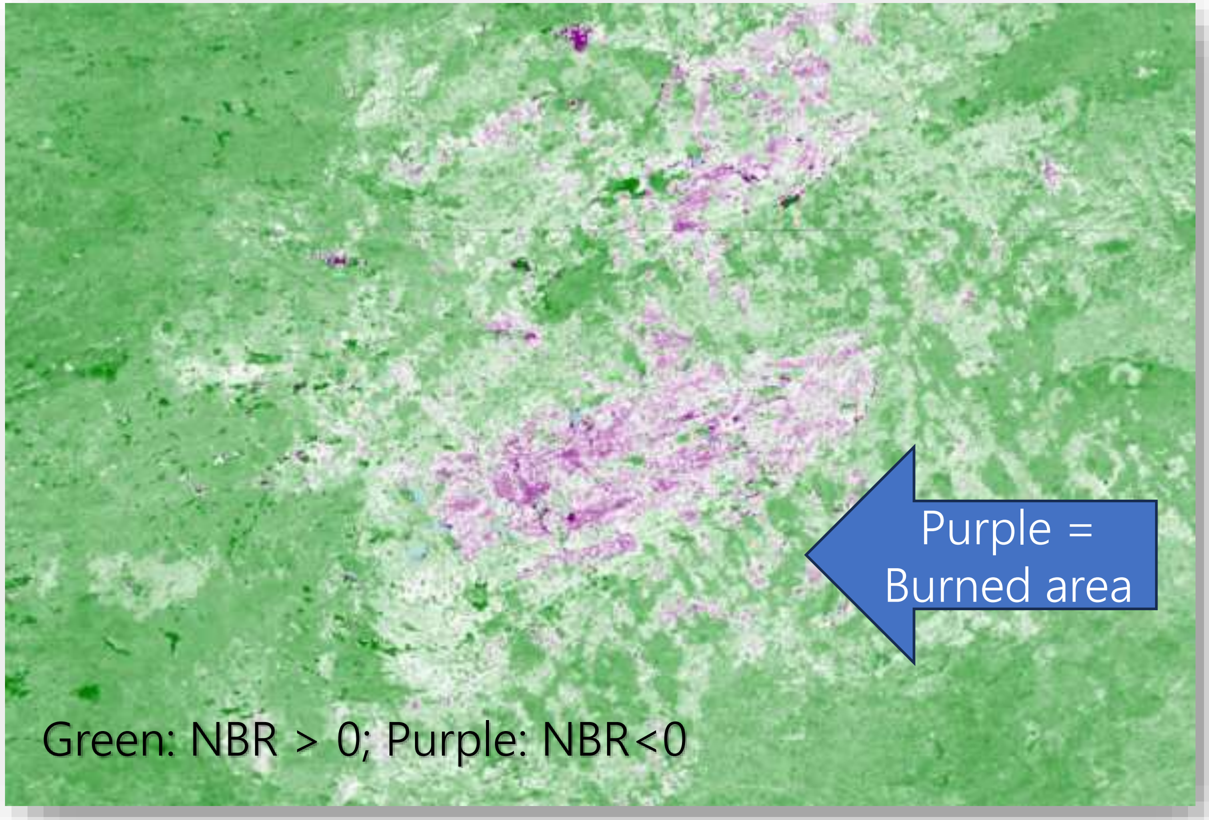
What can we see from Space?
3) Burned Landscape

Normalized Burn Ratio:

$$NBR = \frac{NIR - SWIR}{NIR + SWIR}$$



Chuveico et al. 2019; DOI:10.1016/j.rse.2019.02.013



Fire tends to reduce reflectance in the Near Infrared (NIR), and increase reflectance in the Shortwave Infrared (SWIR)

Mapping burned areas in GEE using Landsat 9 surface reflectance:

- Data Catalog description: https://tinyurl.com/Landsat9Description
- Start from the Example Code
  - what should we change or keep?

```
var dataset = ee.ImageCollection('LANDSAT/LC09/C02/T1_L2')
    .filterDate('2022-01-01', '2022-02-01');

// Applies scaling factors.
function applyScaleFactors(image) {
  var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);
  var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
  return image.addBands(opticalBands, null, true)
              .addBands(thermalBands, null, true);
}

dataset = dataset.map(applyScaleFactors);

var visualization = {
  bands: ['SR_B4', 'SR_B3', 'SR_B2'],
  min: 0.0,
  max: 0.3,
};

Map.setCenter(-114.2579, 38.9275, 8);

Map.addLayer(dataset, visualization, 'True Color (432)');
```

Description | Bands | Image Properties | Terms of Use

Resolution
30 meters

Bands

Name	Units	Min	Max	Scale	Offset	Wavelength	Description
SR_B1		1	65455	2.75e-05	-0.2	0.435-0.451 µm	Band 1 (ultra blue, coastal aerosol) surface reflectance
SR_B2		1	65455	2.75e-05	-0.2	0.452-0.512 µm	Band 2 (blue) surface reflectance
SR_B3		1	65455	2.75e-05	-0.2	0.533-0.590 µm	Band 3 (green) surface reflectance
SR_B4		1	65455	2.75e-05	-0.2	0.636-0.673 µm	Band 4 (red) surface reflectance
SR_B5		1	65455	2.75e-05	-0.2	0.851-0.879 µm	Band 5 (near infrared) surface reflectance
SR_B6		1	65455	2.75e-05	-0.2	1.566-1.651 µm	Band 6 (shortwave infrared 1) surface reflectance
SR_B7		1	65455	2.75e-05	-0.2	2.107-2.294 µm	Band 7 (shortwave infrared 2) surface reflectance

Landsat Quality Flags:

Bit #:	7 Water	6 Clear	5 Snow	4 Cloud Shadow	3 Cloud	2 Cirrus	1 Dilated Cloud	0 Unused
True	1	1	1	1	1	1	1	1
False	0	0	0	0	0	0	0	0

QA_PIXEL

Bitmask for QA_PIXEL

- Bit 0: Fill
- Bit 1: Dilated Cloud
- Bit 2: Cirrus (high confidence)
- Bit 3: Cloud
- Bit 4: Cloud Shadow
- Bit 5: Snow
- Bit 6: Clear
  - 0: Cloud or Dilated Cloud bits are set
  - 1: Cloud and Dilated Cloud bits are not set
- Bit 7: Water

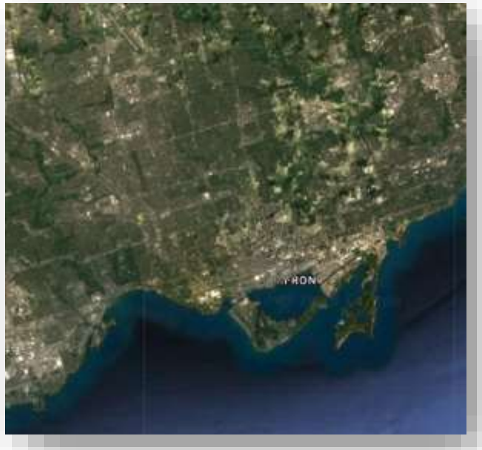
```
function QALandsat(image) {
    var qa = image.select('QA_PIXEL');
    // QA flags are stored as a bitmask. We need to decode the information in the 16 bit binary number.
    // the leftShift function shifts a binary number by the specified number of positions e.g. 0000001 to 0000010.
    // This allows us to extract the bit we want to check without converting to/from binary numbers.
    // First we identify the bit we're interested in for each quality flag:
    // Note: the first bit position is bit 0
    var DilatedCloud = ee.Number(1).leftShift(1);  // 000000010 i.e. from shifting 0000001 left by 1 position
    var Cirrus =       ee.Number(1).leftShift(2);  // 000000100 i.e. from shifting 0000001 left by 2 positions
    var Cloud =        ee.Number(1).leftShift(3);  // 000001000
    var CloudShadow =  ee.Number(1).leftShift(4);  // 000010000
    var Snow =         ee.Number(1).leftShift(5);  // 000100000
    var Clear =        ee.Number(1).leftShift(6);  // 001000000
    var Water =        ee.Number(1).leftShift(7);  // 010000000

    // Next we test if the bit value in the QA pixel
    // and our value defined above are set to 0 in the bit position we're interested in
    // i.e. bitwiseAnd of 110111000 and
    //                    001000000 is 0000000000, or equal to 0
    // whereas bitwiseAnd of  001001000 and
    //                        001000000 is 001000000, or not equal to 0
    // we test each of the bits we're interested in separately:

    //Two options:
    var clearPixels1 = qa.bitwiseAnd(Clear).neq(0);

    var clearPixels2 =
           qa.bitwiseAnd(DilatedCloud).eq(0)
      .and(qa.bitwiseAnd(Cirrus).eq(0))
      .and(qa.bitwiseAnd(Cloud).eq(0))
      .and(qa.bitwiseAnd(CloudShadow).eq(0))
      .and(qa.bitwiseAnd(Snow).eq(0))
      //.and(qa.bitwiseAnd(Water).eq(0)); // we can comment/uncomment each line to select what we wish to mask

    return image.updateMask(clearPixels2); //applies the mask to each pixel in the image
}
```

Load the Landsat data and apply the scale factors and QA mask

```
var L9_BANDS = ['SR_B2', 'SR_B3', 'SR_B4', 'SR_B5', 'SR_B6', 'SR_B7', 'ST_B10'];
var STD_NAMES = ['blue', 'green', 'red', 'nir', 'swir1', 'swir2', 'temp'];

var Landsat9 = ee.ImageCollection('LANDSAT/LC09/C02/T1_L2') //Loads the Landsat 9 image collection
    .filterDate(start_date, end_date) //filters the image collection by date
    .map(applyScaleFactors) //applies the applyScaleFactors function to each image in the collection
    .map(QALandsat) //applies the QALandsat function to each image in the collection
    .select(L9_BANDS, STD_NAMES) //selects only the bands we want, and renames them
    .median() ; //calculates the median pixel value across the filtered, masked, collection.
    // The 'median()' is a function called a 'reducer' in earth engine that converts an image collection
    // to a single image.
```

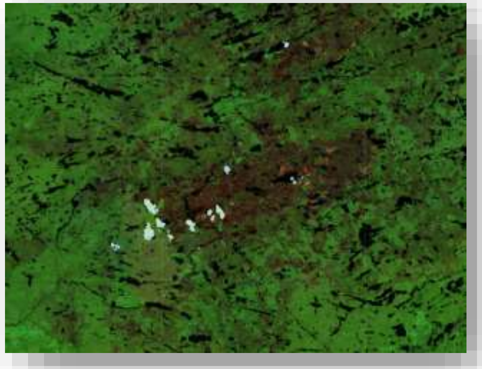
Set Visualization Settings

```
//Landsat True Colour
var visualizationTrueColor = {
  bands: ['red', 'green', 'blue'],
  min: 0.0,
  max: 0.3
};

//Landsat False Colour - Vegetation
var visualizationFC = {
  bands: ['nir', 'red', 'green'],
  min: 0.0,
  max: 0.5
};

//Landsat False Colour - Fire
var visualizationFire = {
  bands: ['swir2', 'nir', 'red'],
  min: 0.0,
  max: 0.5
};
```



True Colour Landsat 9 Image
R= Red; G=Green; B=Blue

False Colour Landsat 9 Image
R = NIR; G = Red; B = Green



False Colour
R= SWIR; G=NIR; B=Red

Calculate Indices:

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

$$NBR = \frac{NIR - SWIR}{NIR + SWIR}$$

```
// NDVI = (NIR-RED)/(NIR+RED)
var NDVI = Landsat9.select('nir').subtract(Landsat9.select('red'))
  .divide(Landsat9.select('nir').add(Landsat9.select('red')));

// NBR = (NIR-SWIR)/(NIR+SWIR)
var NBR = Landsat9.select('nir').subtract(Landsat9.select('swir2'))
  .divide(Landsat9.select('nir').add(Landsat9.select('swir2')));
```

Add Everything to the Map

eeObject

visParams
(visualization settings)

Name
(for the layer selector)

```
Map.addLayer(Landsat9, visualizationTrueColor, 'True Colour (432)');
Map.addLayer(Landsat9, visualizationFire, 'False Colour (754)');
Map.addLayer(Landsat9, visualizationFC, 'False Colour (543)');
Map.addLayer(
  {eeObject: NDVI,
   visParams: {palette : ['purple','white','green'],
   min: 0, max: 0.7},
   name: 'NDVI'
  });
Map.addLayer(
  {eeObject: NBR,
    visParams: {palette : ['purple','white','green'],
    min: -0.7, max: 0.7},
    name: 'NBR'
  });
```

Since NDVI and NBR are single band images, we use a “palette”, rather than assigning bands to Red, Green, and Blue

Adjust the dates:

- We will want to make sure the fire occurred before the Landsat imagery was acquired:

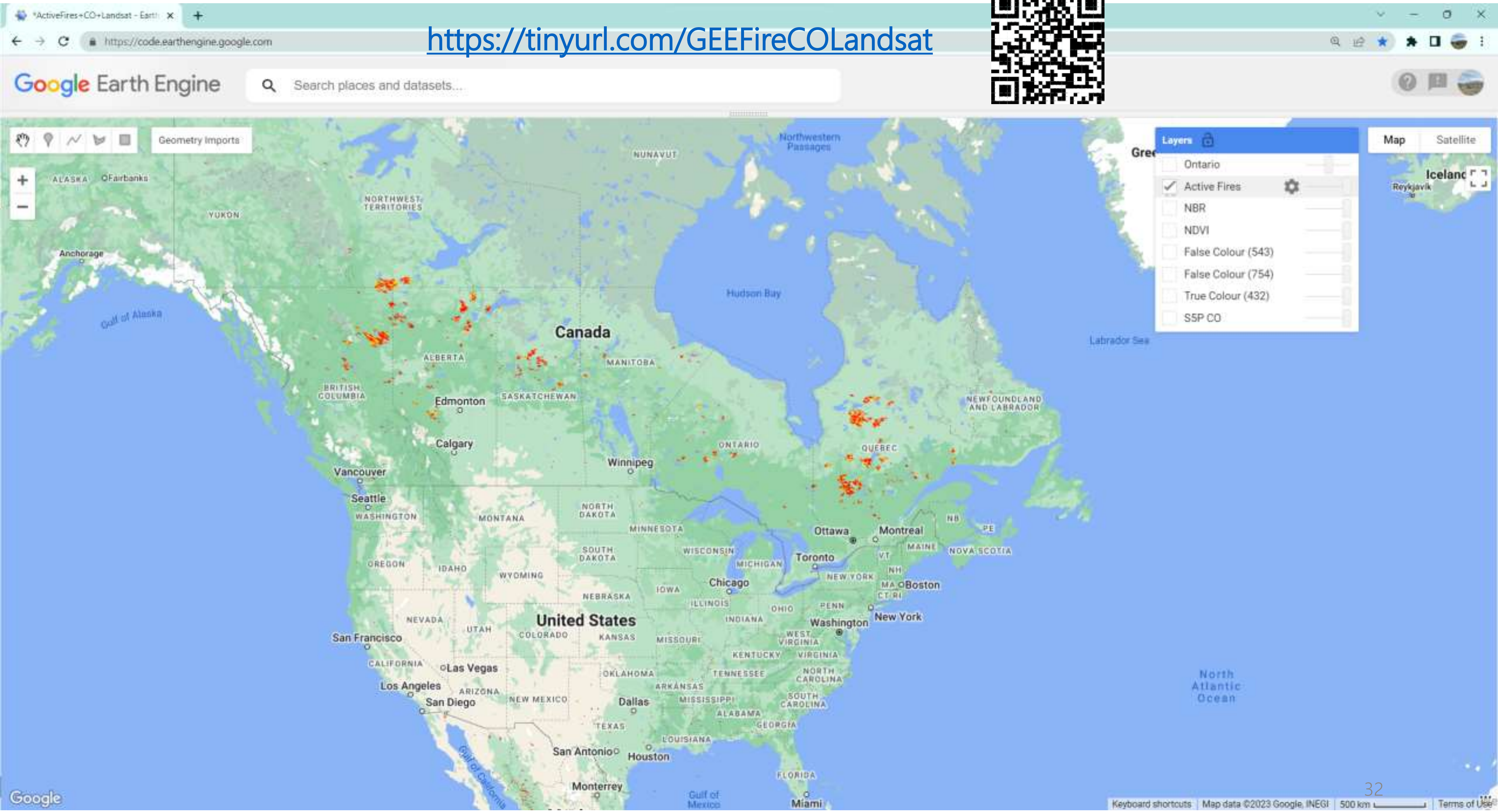
```
//Date range for active fire and carbon monoxide data:
var start_date_fire_co = '2023-06-01';
var end_date_fire_co = '2023-06-15';

//Date range for Landsat imagery:
var start_date_landsat = '2023-06-15';
var end_date_landsat = '2023-06-30';
```

```
//FIRMS Active Fires
var Fire_dataset = ee.ImageCollection('FIRMS') //Adjust the name of the dataset
.filter(
    ee.Filter.date(start_date_fire_co, end_date_fire_co) //filters the data to a date range
    );
var fires = Fire_dataset.select('T21').mean(); //selects a 'band' in Fire_dataset called 'T21'

//Tropomi/Sentinel-5P Carbon Monoxide
var CO_collection = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_CO')
  .select('CO_column_number_density')
  .filterDate(start_date_fire_co, end_date_fire_co)  //filters the collection of images by date
```

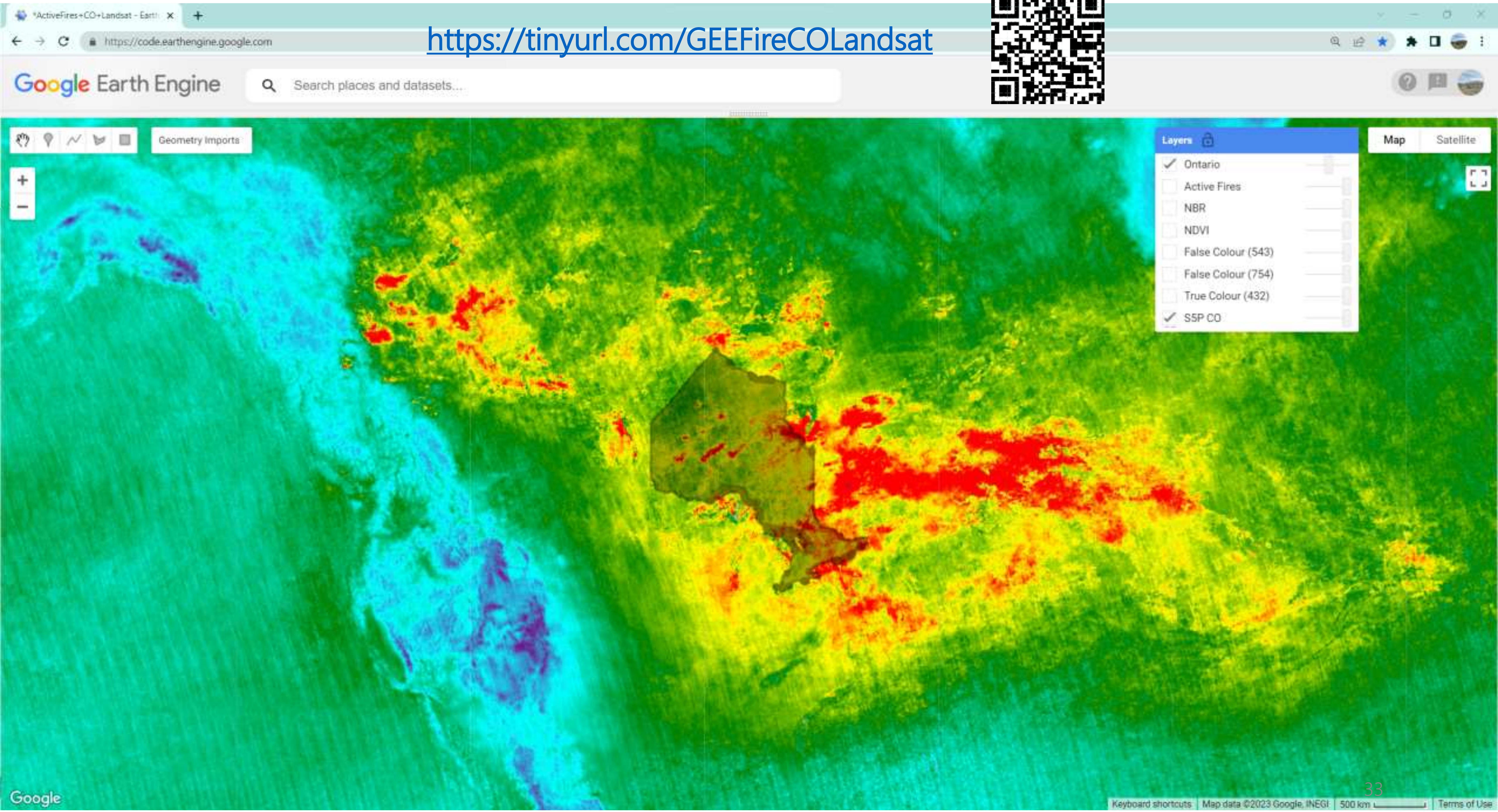
```
var Landsat9 = ee.ImageCollection('LANDSAT/LC09/C02/T1_L2') //Loads the Landsat 9 image collection
    .filterDate(start_date_landsat, end_date_landsat) //filters the image collection by date
    .map(applyScaleFactors) //applies the applyScaleFactors function to each image in the collection
    .map(QALandsat) //applies the QALandsat function to each image in the collection
    .select(L9_BANDS, STD_NAMES) //selects only the bands we want, and renames them
    .median() ; //calculates the median pixel value across the filtered, masked, collection.
    // The 'median()' is a function called a 'reducer' in earth engine that converts an image collection
    // to a single image.
```

https://tinyurl.com/GEEFireCOLandsat

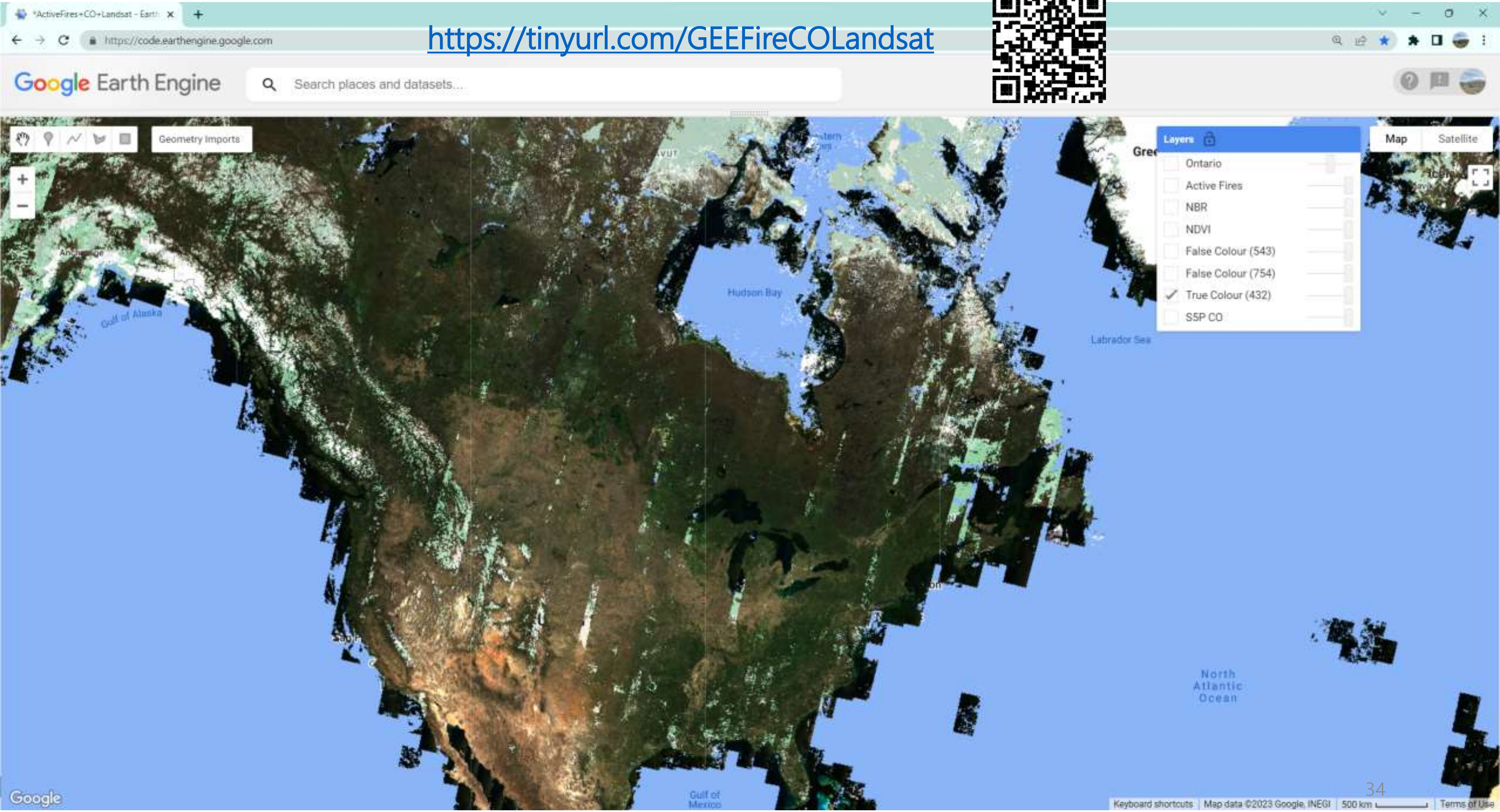
https://tinyurl.com/GEEFireCOLandsat



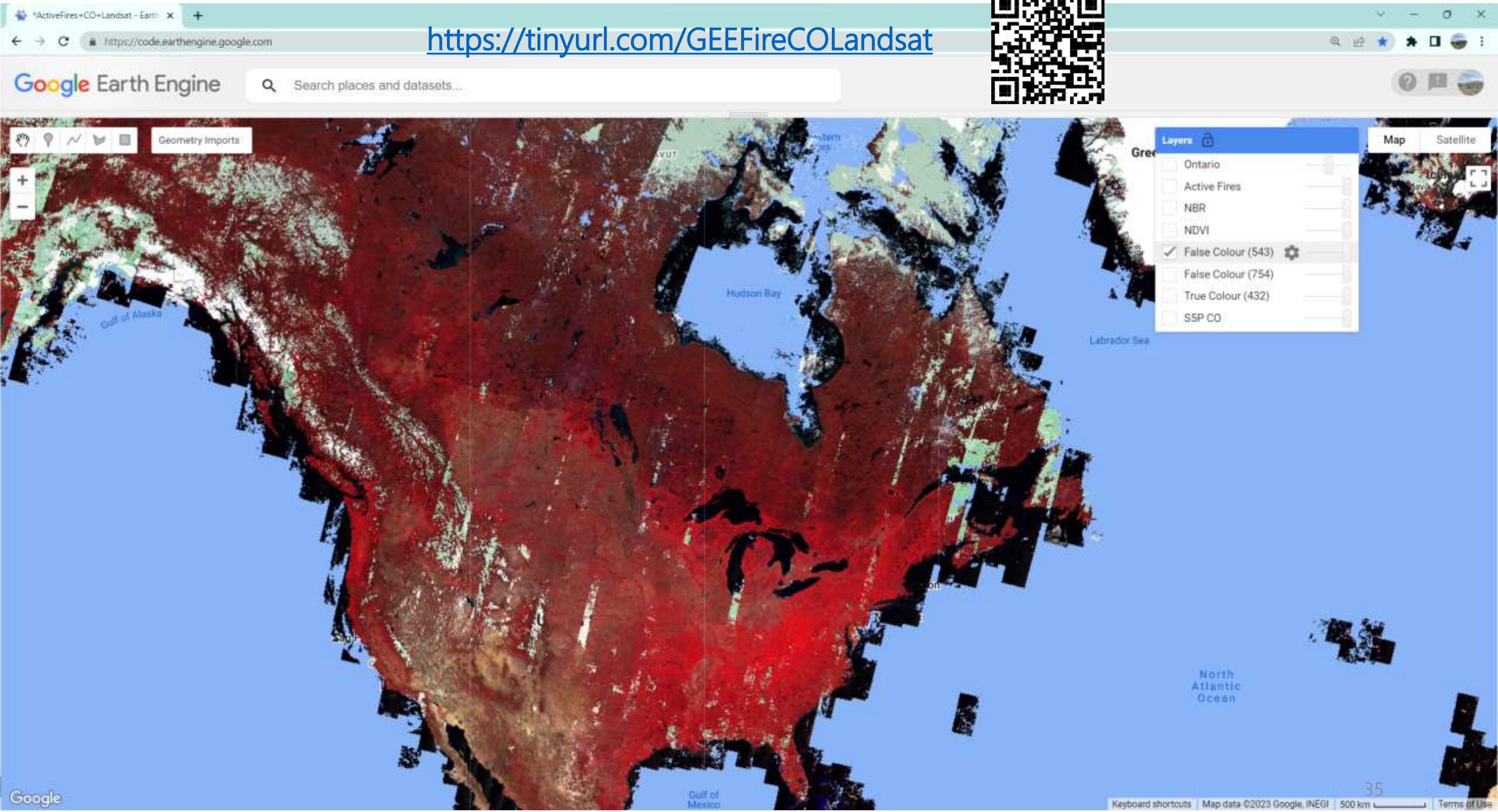
https://tinyurl.com/GEEFireCOLandsat



https://tinyurl.com/GEEFireCOLandsat

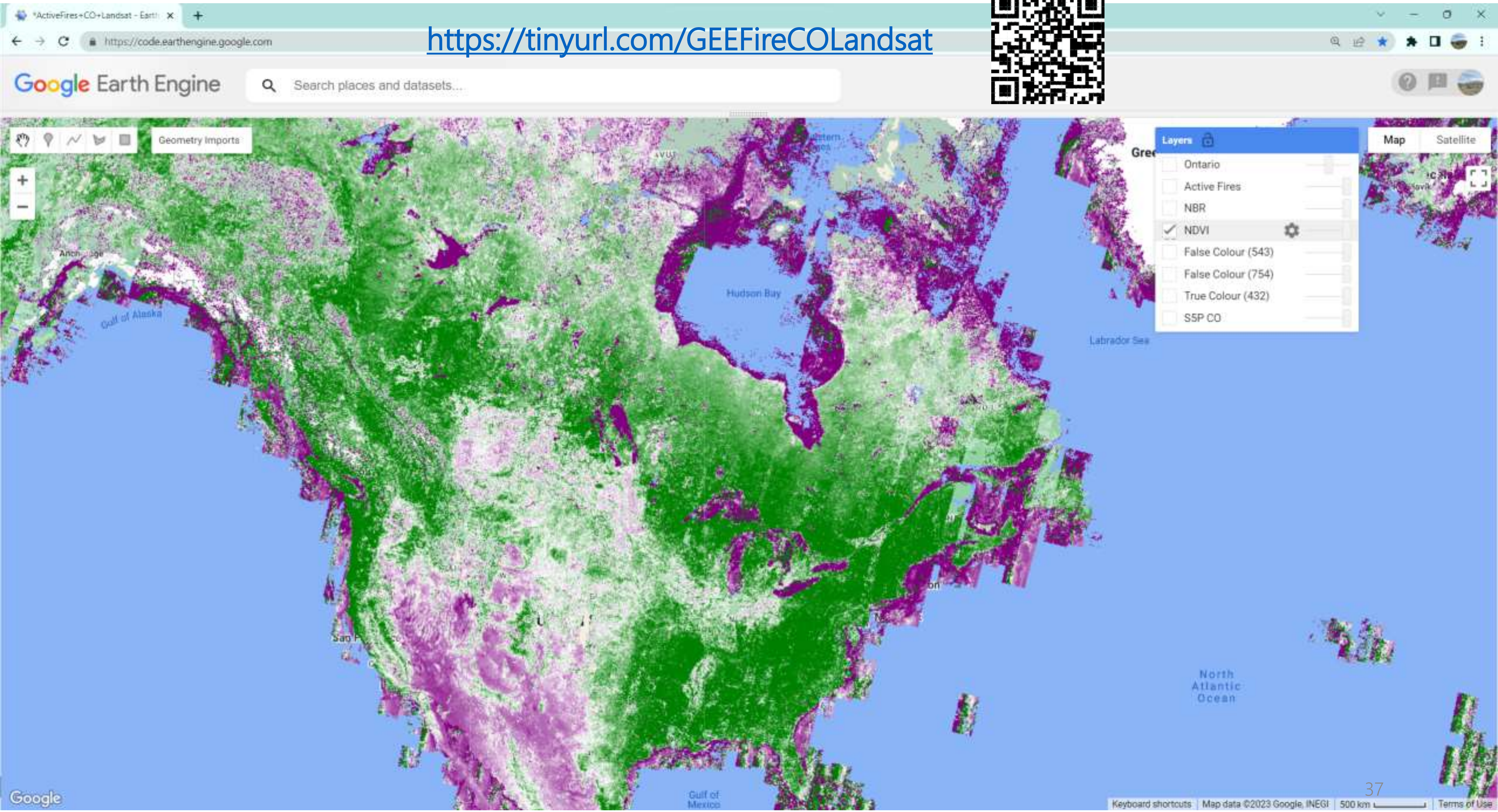


https://tinyurl.com/GEEFireCOLandsat

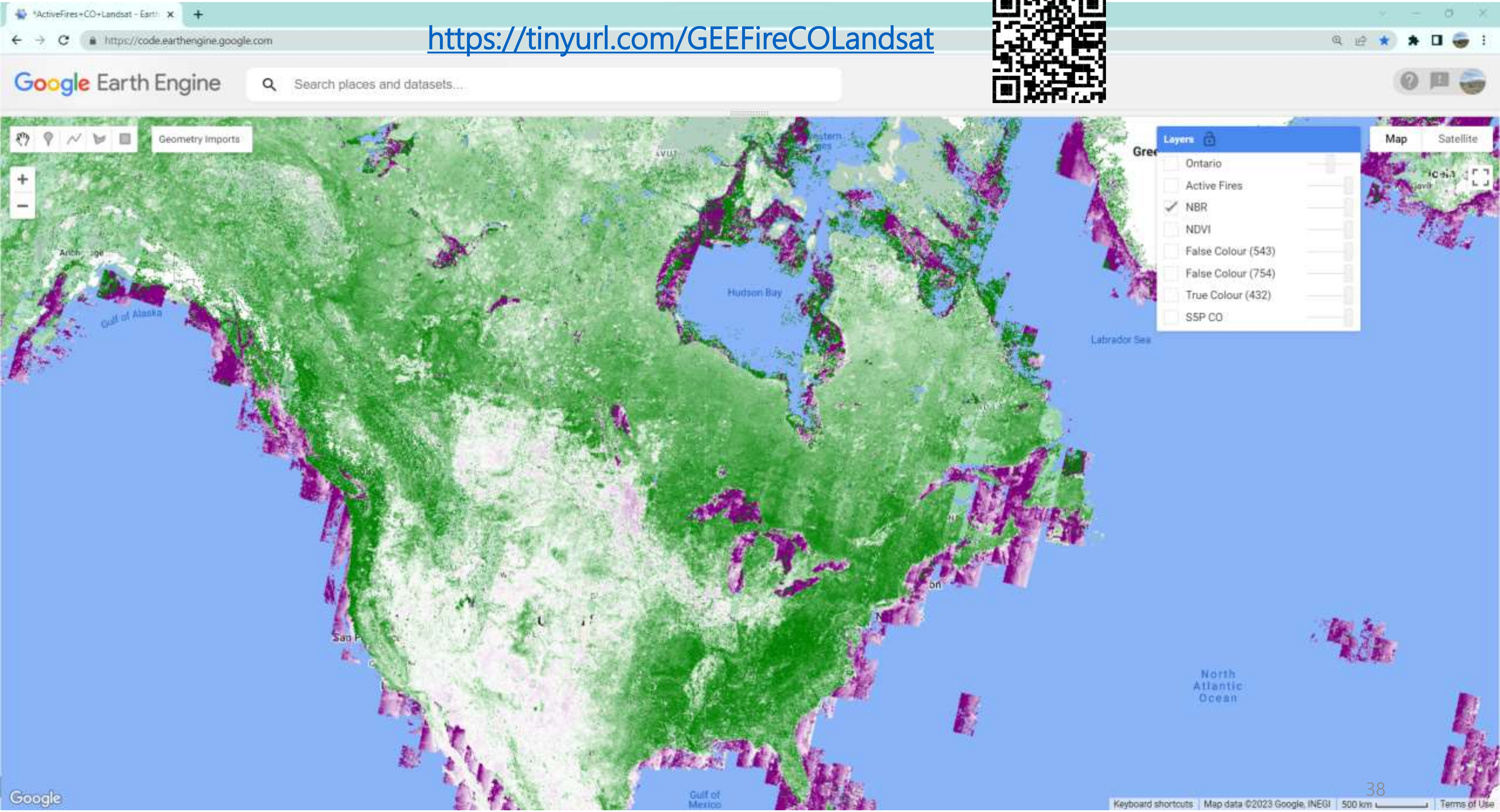


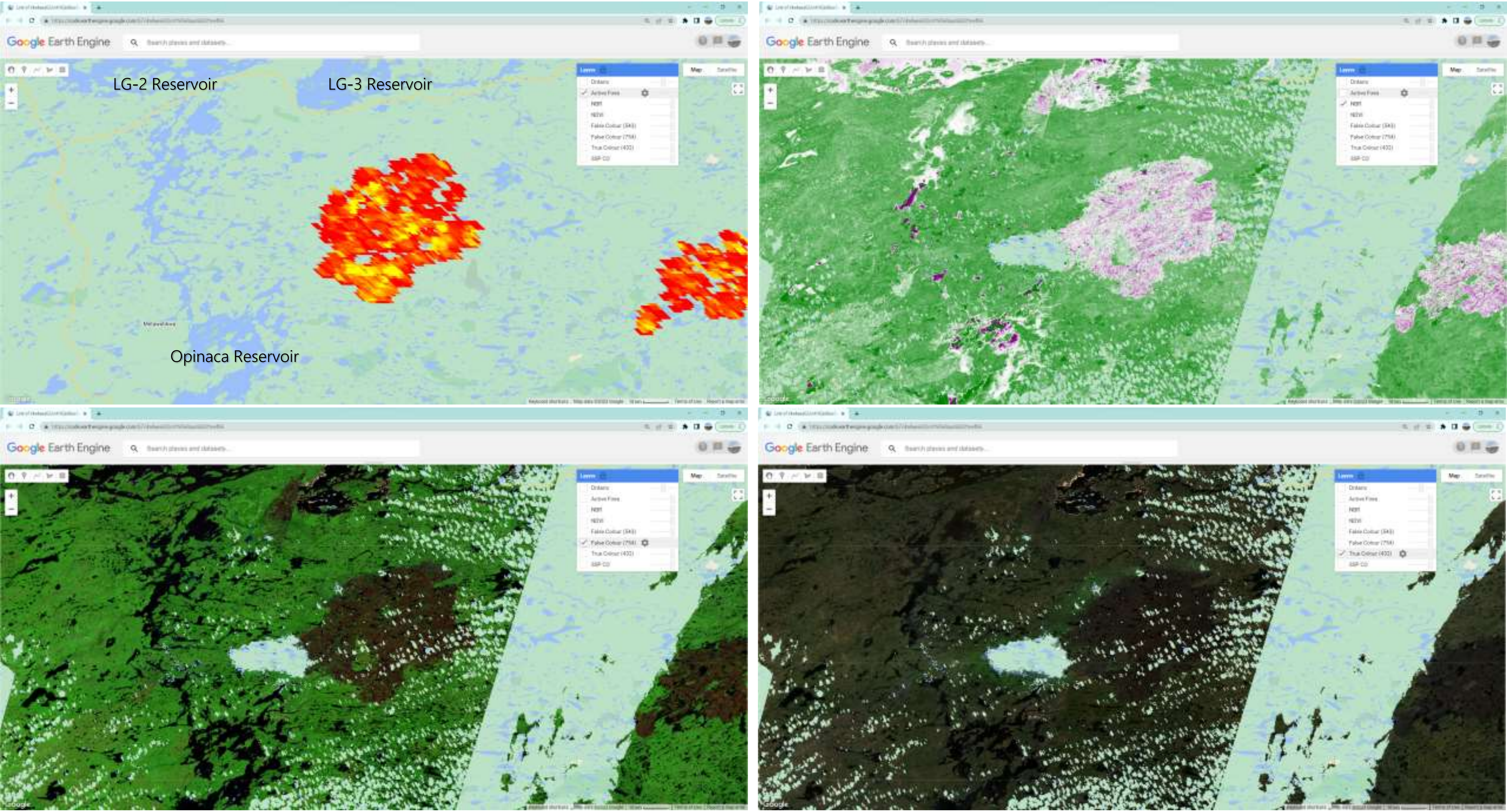
https://tinyurl.com/GEEFireCOLandsat

https://tinyurl.com/GEEFireCOLandsat



https://tinyurl.com/GEEFireCOLandsat




LG-2 Reservoir
LG-3 Reservoir
Opinaca Reservoir
Google Earth Engine

Explore:
https://tinyurl.com/GEEFireCOLandsat

Note: don't be afraid of 'breaking' the code! If you encounter an error, you can always start over from the link!

Suggestions:

- Look at different regions across Canada or even the World (Zoom in/out)
- Look at different date ranges
    - Is the atmospheric Carbon Monoxide different from the same time last year (May – June)?
    - When does fire typically occur in different areas of the world?
    - Are there more fires this year than last year?
    - Landsat 9 was launched in Sept. 2021, is there an alternative surface reflectance dataset we could use for previous years? Search the GEE data catalogue at: https://developers.google.com/earth-engine/datasets
    - What happens to the Landsat Imagery if the date range only includes a few days?
    - For how long after a fire is a burn scar detectable with Landsat?
    - How might we detect change over time using NDVI or NBR?
- Adjust the Visualization Settings for the different layers. Try a different color palette, or adjusting max and min values
- What happens to the Landsat Imagery if you don't mask bad pixels?
- How could you update the code to monitor this summer's fire season as it progresses?

Wrap - Up

- Select a SMALL area (a single fire) to export a Landsat 9 image
  - Use the rectangular geometry tool.
- You will use this data for <u>Assignment 4</u> – Change detection and 3D visualization of Canada's 2023 Wildfire Season in ArcGIS Pro

```
Export.image.toDrive({
  image: Landsat9,
  folder: "yourAssignment4Folder",
  fileNamePrefix: "yourFileName",
  region: geometry,
  crsTransform: Landsat9.select('red').projection().getInfo().transform,
  crs: Landsat9.select('red').projection().getInfo().crs
  });
```

Summary:

- We applied the theoretical concepts from previous lectures using Satellite Datasets and Google Earth Engine:
  - Thermal: MODIS
  - Atmospheric: TROPOMI (Sentinel 5P)
  - Optical: Landsat
- We can now perform the following tasks in Google Earth Engine:
  - Import data for analysis
  - Filter image collections and feature collections by criteria such as:
    - Properties
    - Date
    - Image bands
  - Mask poor quality pixels
  - Calculate Vegetation Indices
  - Customize image visualizations
  - Export images to Google Drive for offline analysis (e.g. in ArcGIS Pro)