



Google

Import, Export, & Render

Chris Herwig, Google

Agenda

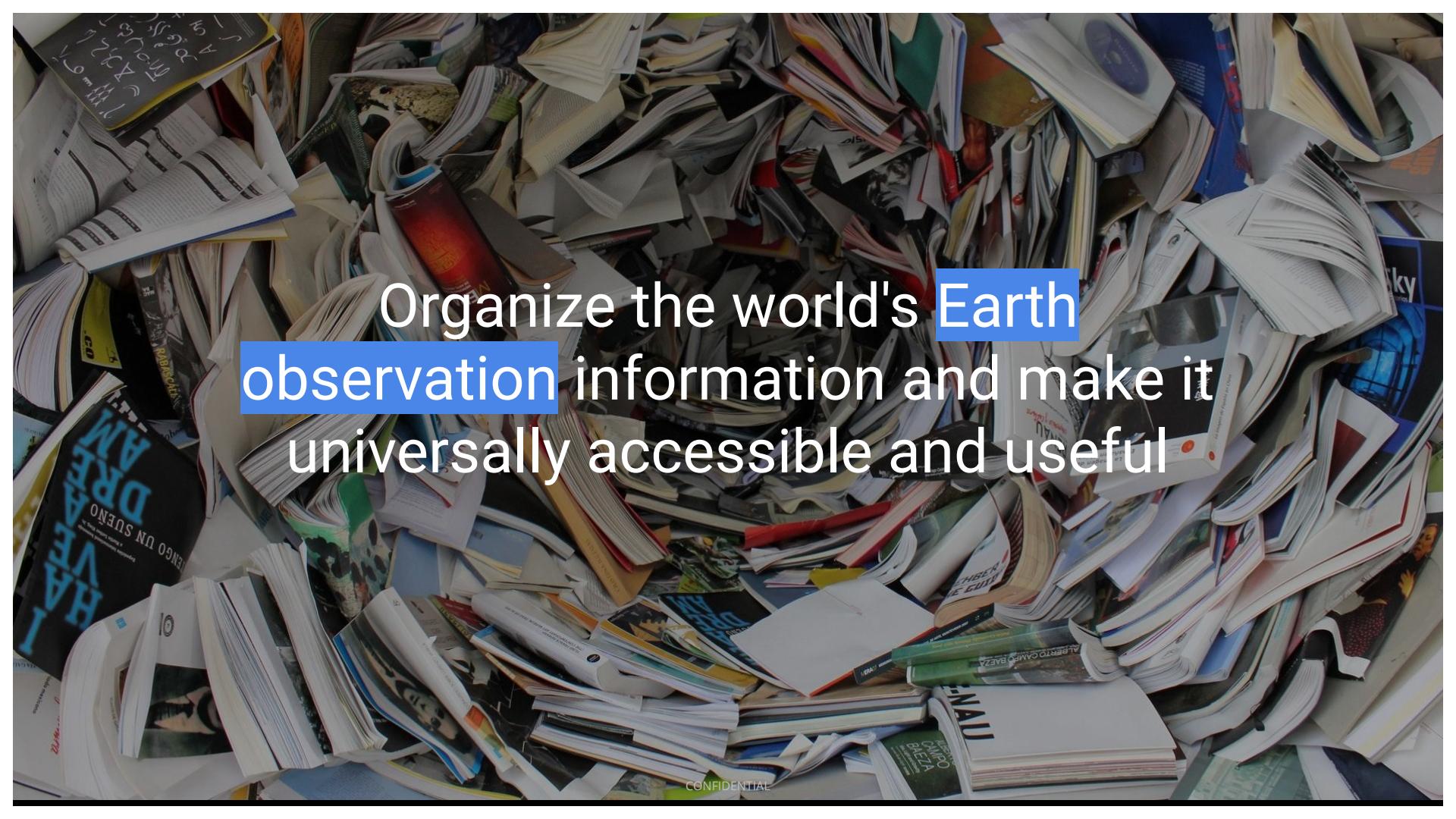
1. Working with data in public data catalog
2. Types of data in Earth Engine
3. Analyzing your own data in EE
4. Taking your results out of EE



What's new?

1. CSVs are now importable from the Code Editor
and via command line utility.
2. Export array images as assets





Organize the world's Earth
observation information and make it
universally accessible and useful



Chart



Image



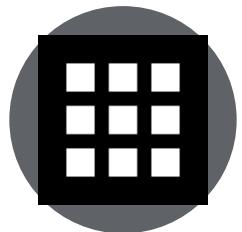
Map



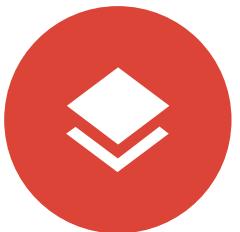
Video



Table



App

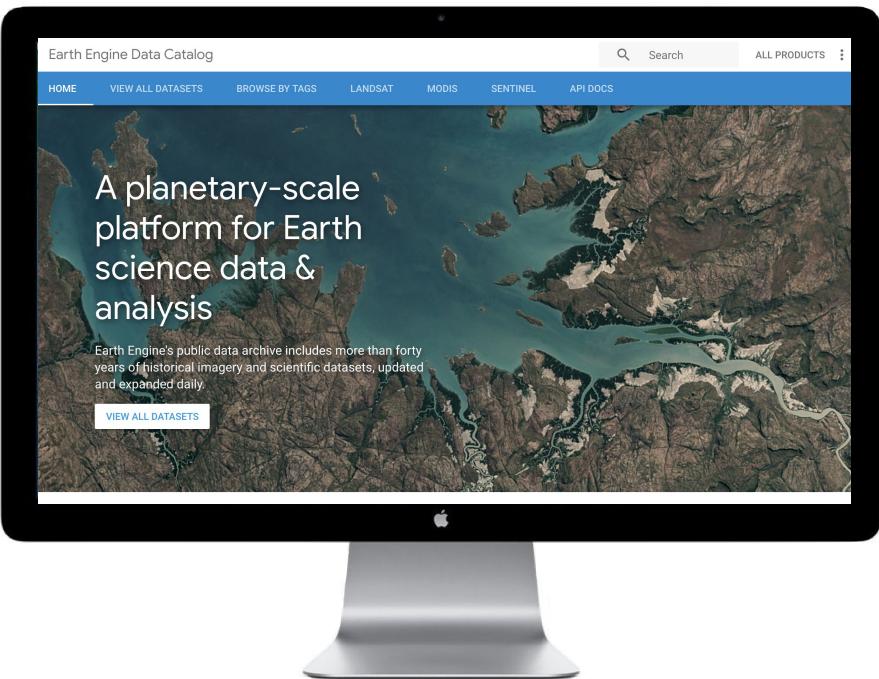


Earth Engine Asset



Public Data Catalog

[developers.google.com/
earth-engine/datasets/](https://developers.google.com/earth-engine/datasets/)

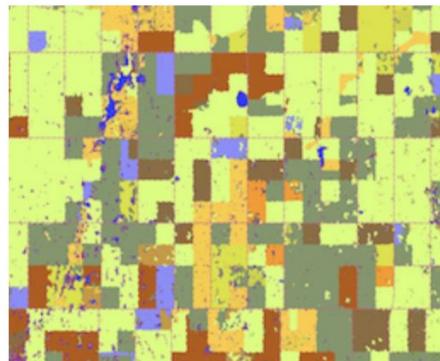


Earth Engine Data Catalog

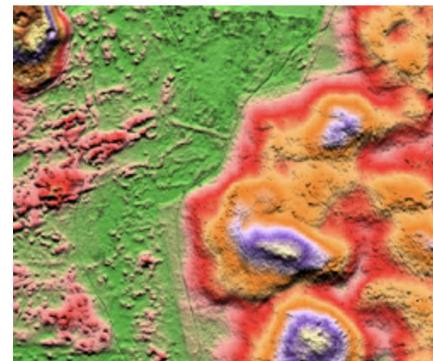
Earth Engine's public data catalog includes a variety of standard Earth science raster datasets. You can import these datasets into your script environment with a single click. You can also upload your own [raster data](#) or vector data for private use or sharing in your scripts.

Filter list of datasets

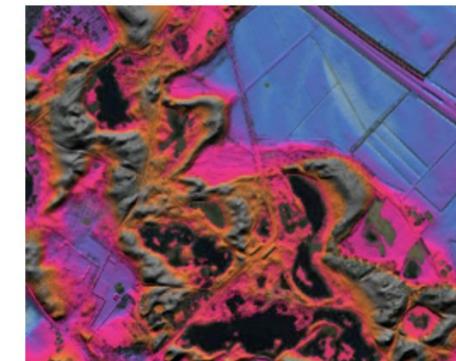
[Canada AAFC Annual Crop Inventory](#)



[AHN Netherlands 0.5m DEM, Interpolated](#)



[AHN Netherlands 0.5m DEM, Non-Interpolated](#)



HOME

VIEW ALL DATASETS

BROWSE BY TAGS

LANDSAT

MODIS

SENTINEL

API DOCS

Browse by Dataset Tags

 Filter dataset tags[16-day](#)

1 datasets

[16day](#)

9 datasets

[3-hourly](#)

5 datasets

[30-year](#)

1 datasets

[4day](#)

1 datasets

[8-day](#)

1 datasets

[8day](#)

12 datasets

[aafc](#)

1 datasets

[aboveground](#)

1 datasets

[accessibility](#)

2 datasets

[accumulation](#)

2 datasets

[aerial](#)

1 datasets

[aerosol](#)

1 datasets

[agriculture](#)

1 datasets

[ahn](#)

3 datasets

[air-temperature](#)

1 datasets

[albedo](#)

7 datasets

[alos](#)

4 datasets

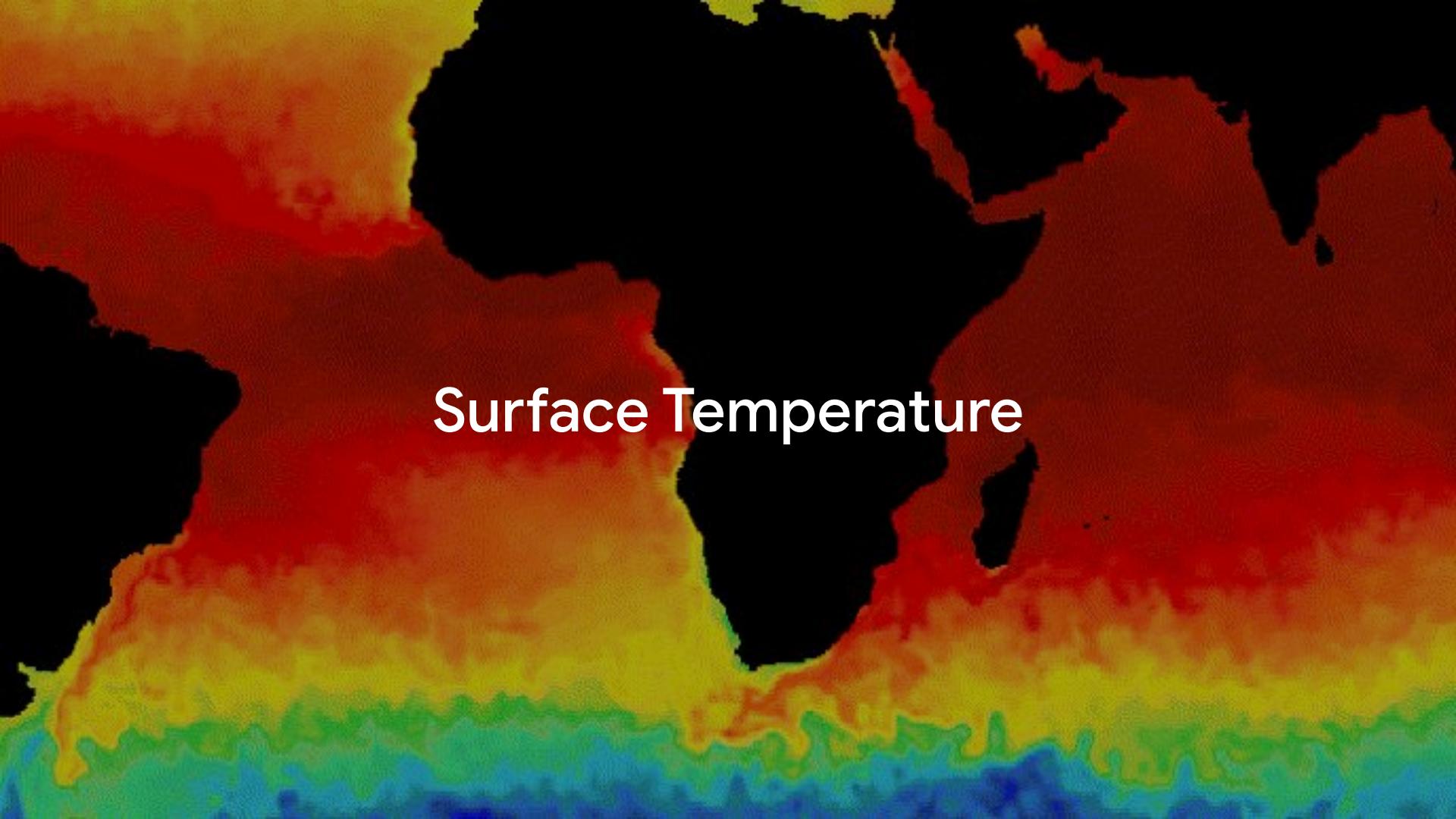
[alos2](#)

2 datasets

[annual](#)

3 datasets

[antarctica](#)[aot](#)[aqua](#)[arctic](#)



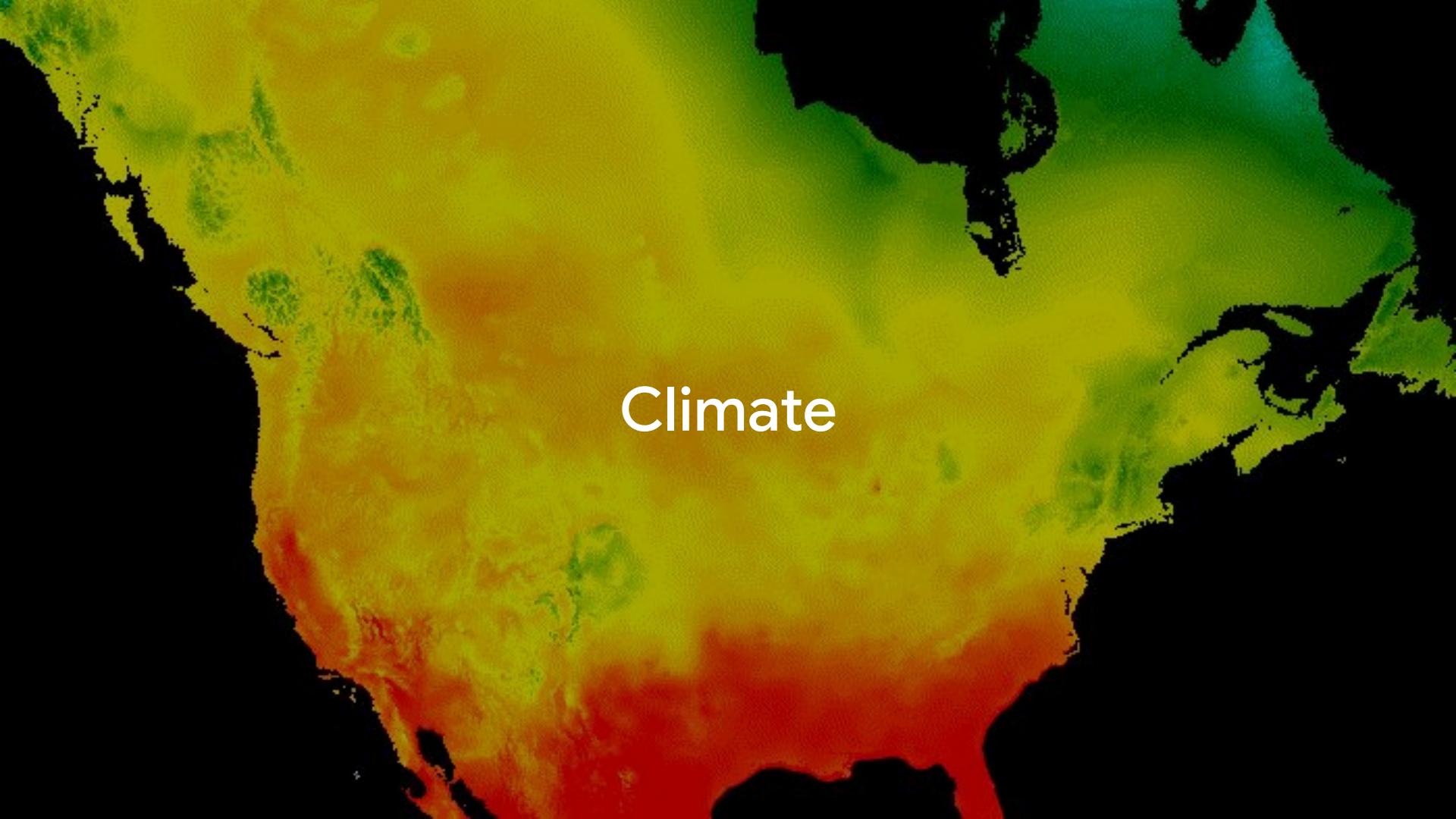
Surface Temperature

An aerial photograph of a park area. The central feature is a large grassy field with several baseball diamonds. There are also basketball courts and playground equipment. The park is surrounded by trees and is adjacent to a road and some buildings.

High-res Imagery



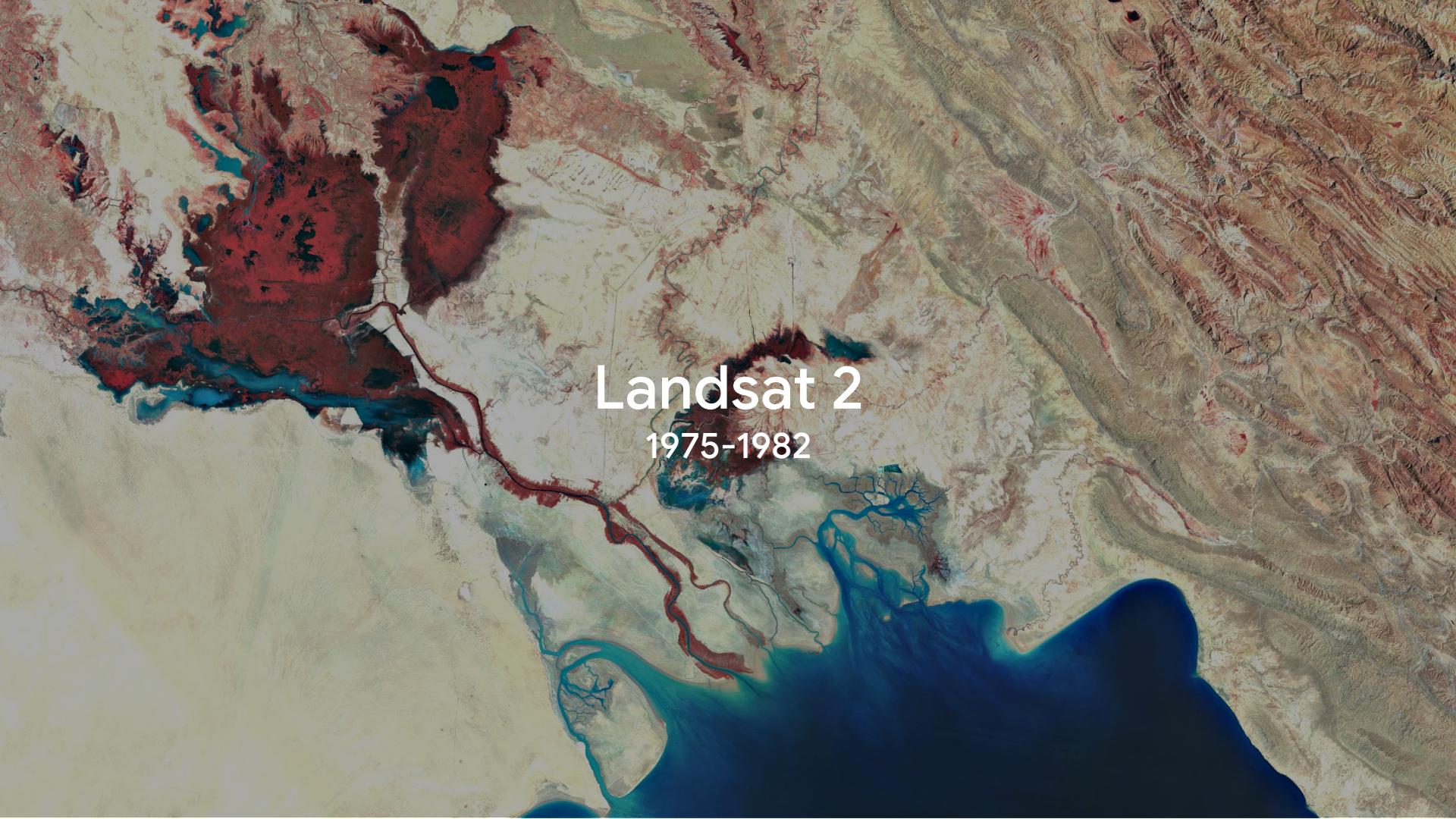
Terrain



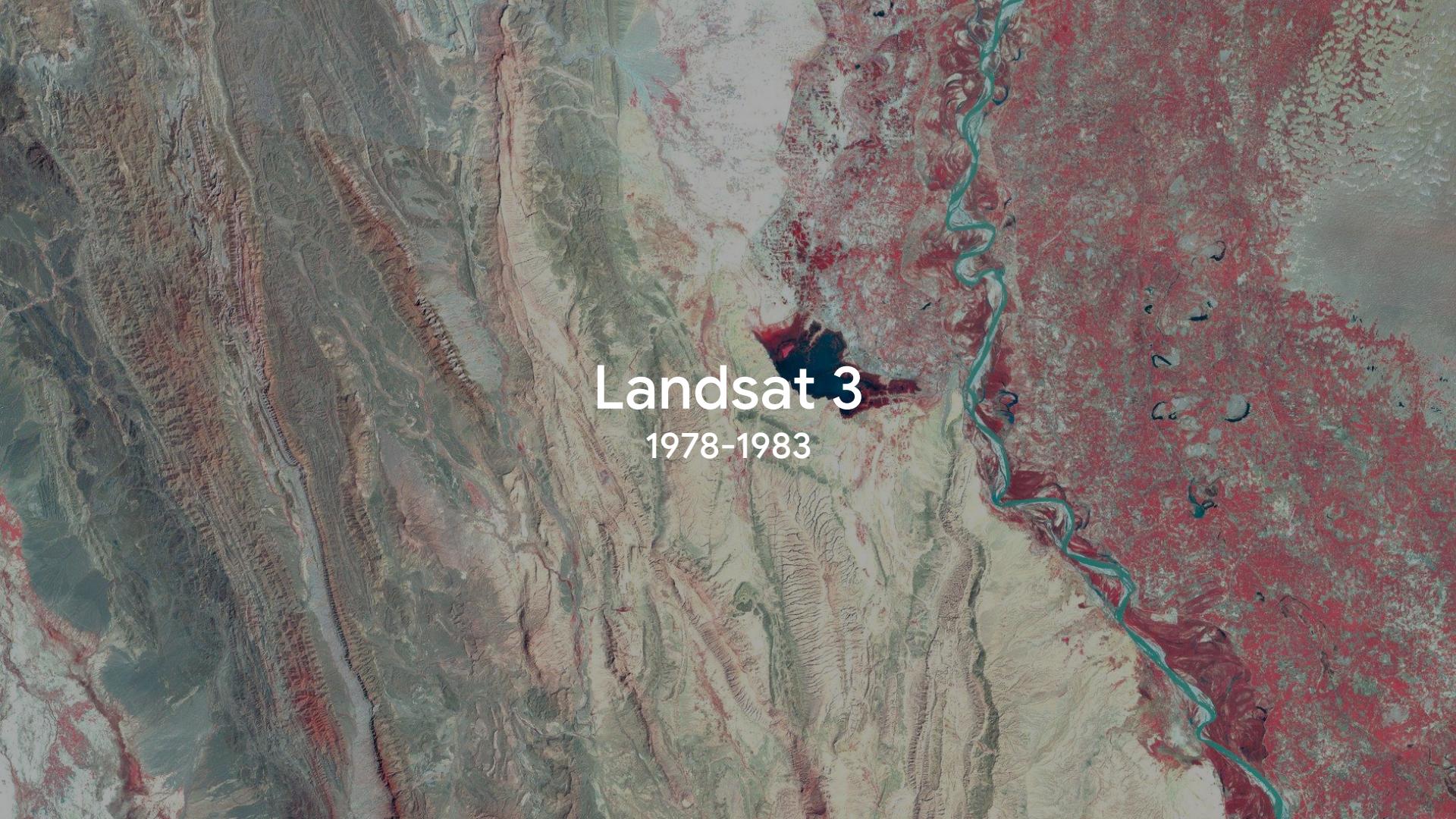
Climate



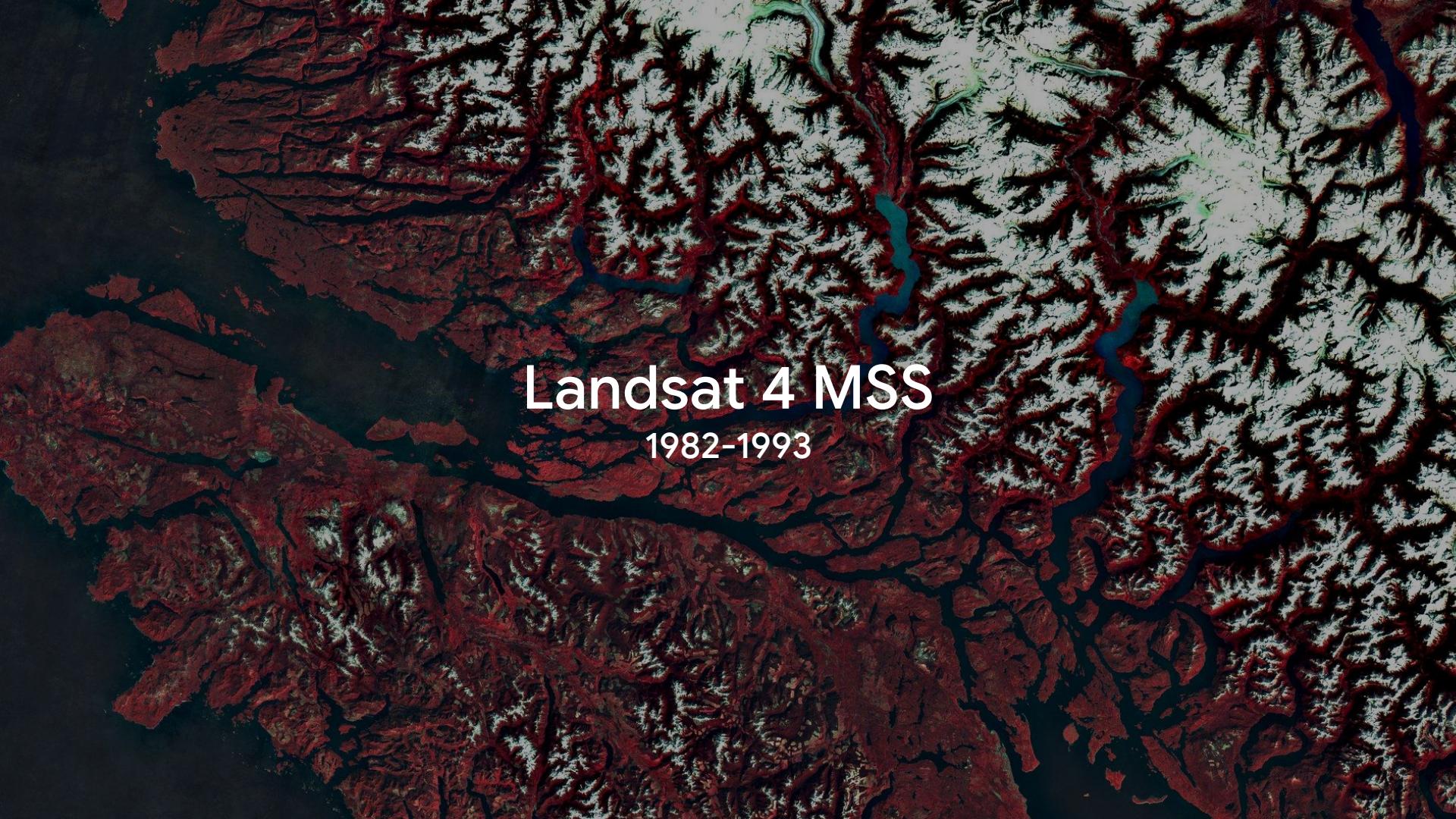
Landsat 1
1972-1978



Landsat 2
1975-1982



Landsat 3
1978-1983



Landsat 4 MSS

1982-1993

A satellite map of Lake Urmia, Iran, showing its dramatic shrinkage between 1982 and 1993. The lake, which was once the world's fourth largest, has lost over 90% of its surface area. The remaining dark blue-green water is fragmented into several large, irregular patches. The surrounding land is a mix of brown, tan, and yellowish-brown colors, indicating dry, cracked earth and exposed salt flats. The text 'Landsat 4 TM' and '1982-1993' is overlaid in the center-left.

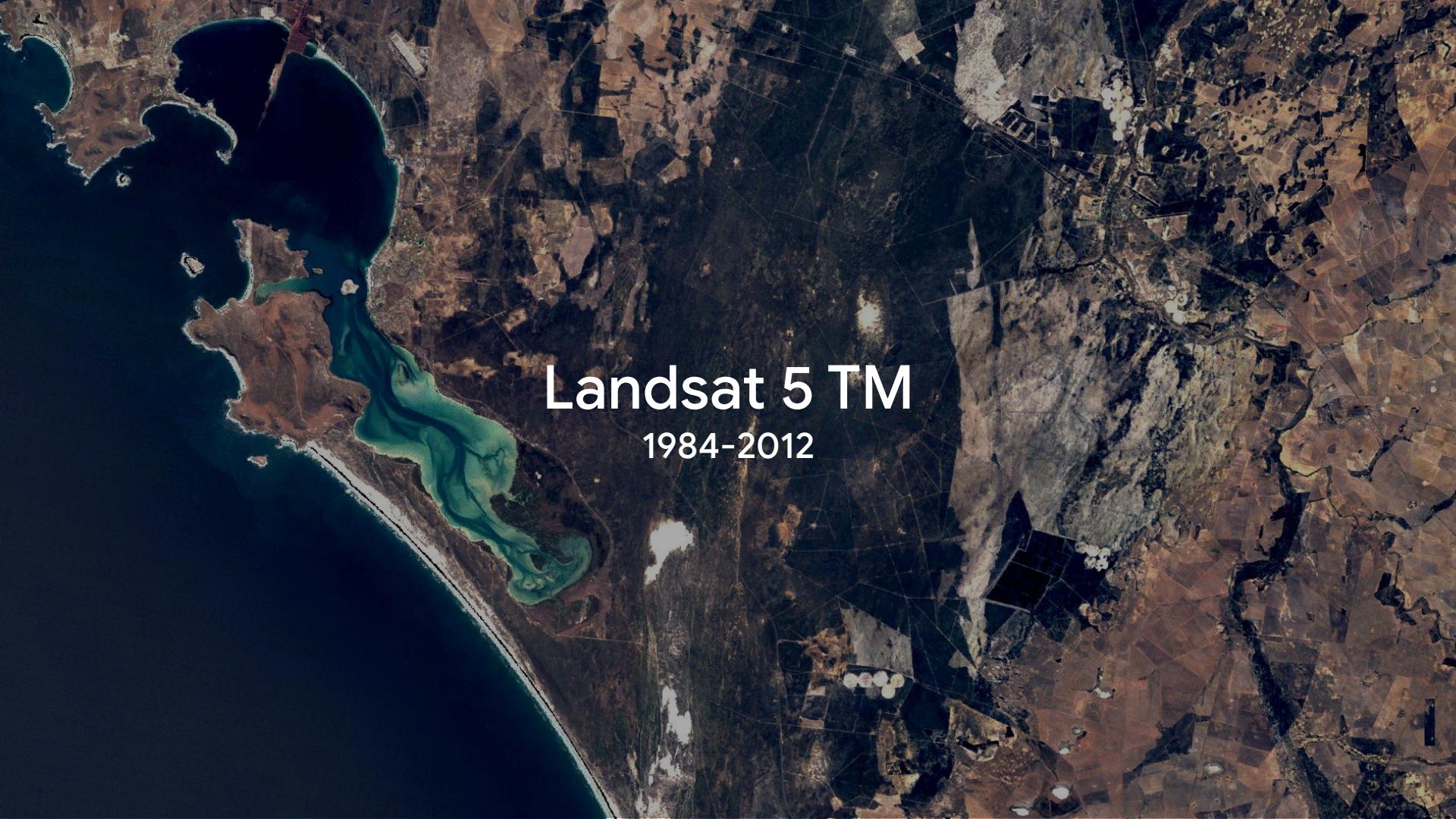
Landsat 4 TM

1982-1993

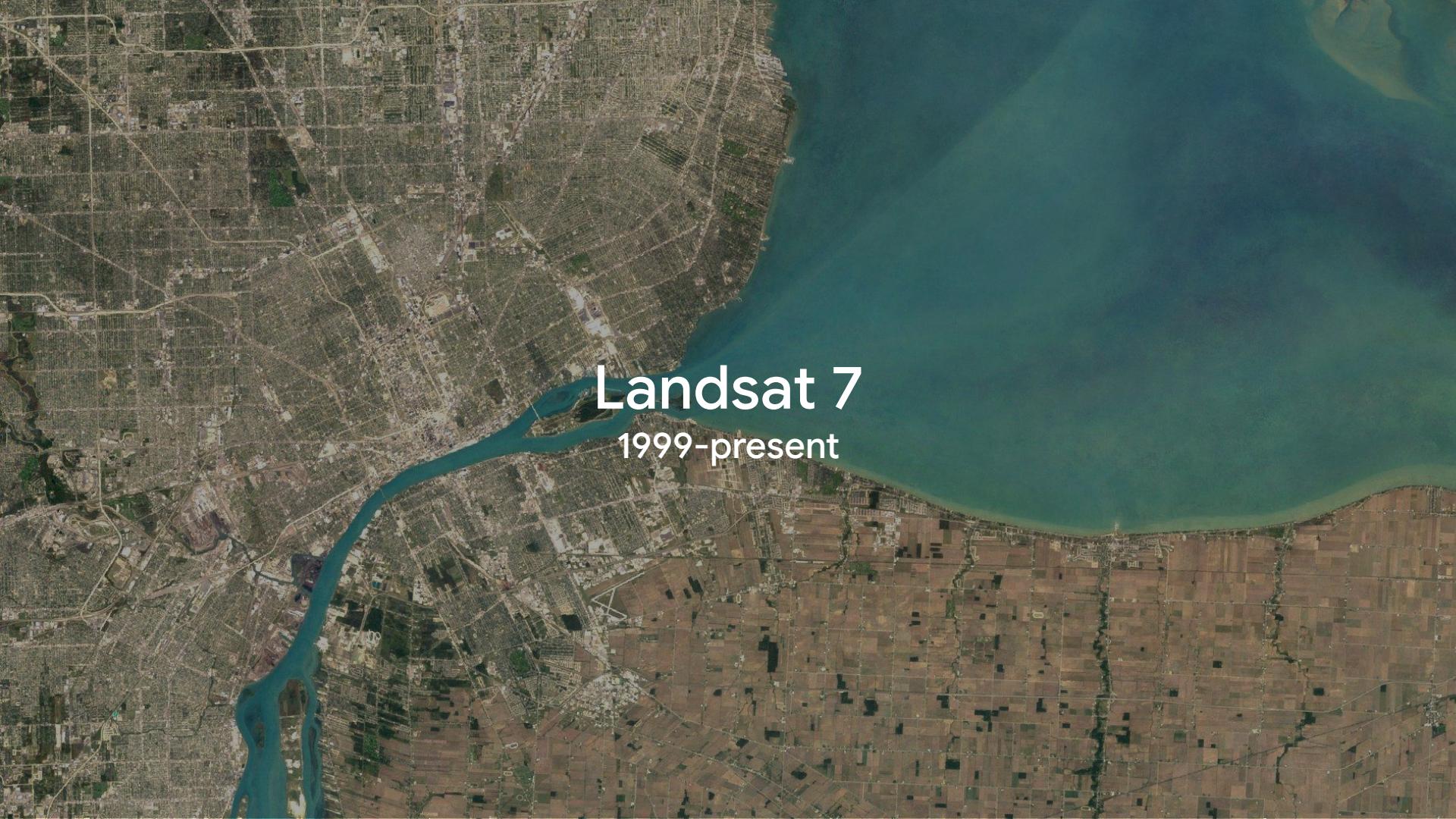


Landsat 5 MSS

1984-2012



Landsat 5 TM
1984-2012

A satellite image of the Detroit metropolitan area, showing a dense grid of urban development along the coast of Lake Michigan. A large river, likely the Detroit River, flows through the center of the city. To the east, there is a mix of urban areas and large agricultural fields in a grid pattern.

Landsat 7
1999-present

A satellite image showing a hilly terrain with various land cover types. A winding river or stream is visible in the lower-left quadrant. The surrounding land is a mix of reddish-brown and greenish-brown colors, suggesting different soil types or vegetation. Some darker, more forested areas are scattered throughout.

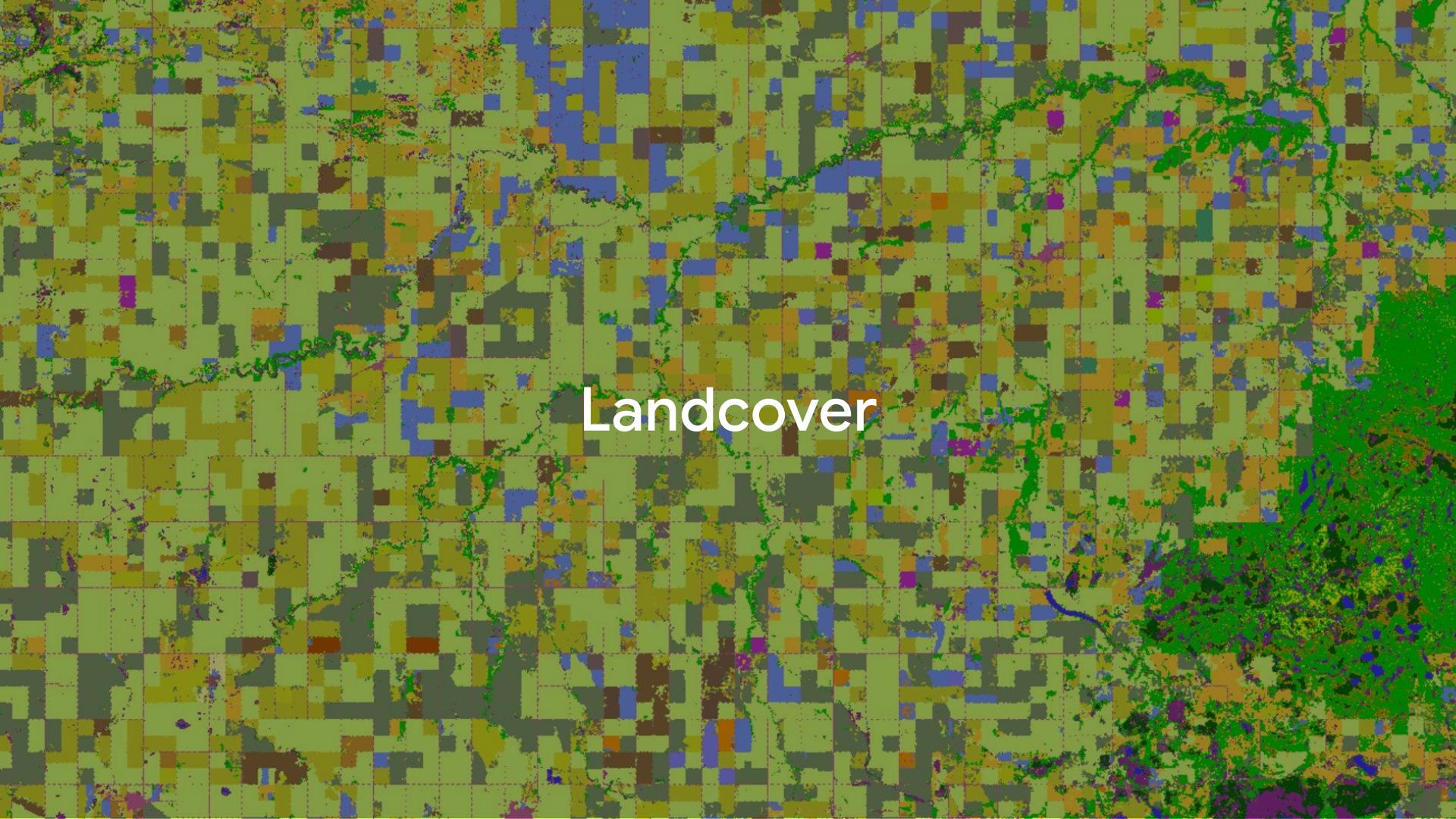
Landsat 8
2013-present



Landsat 4, 1980s



Landsat 8, 2015~

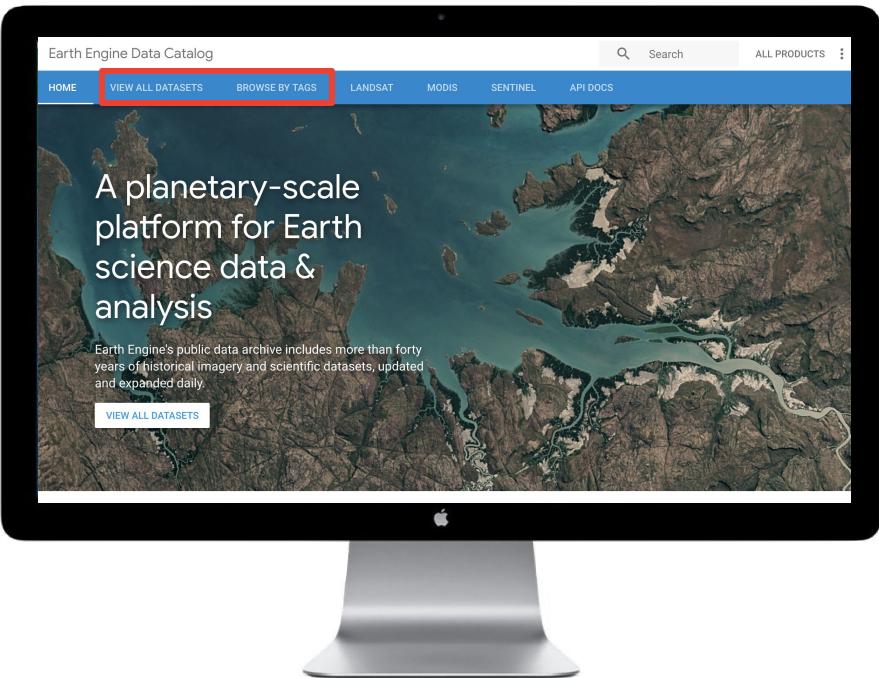


A detailed satellite map of a rural landscape, likely a agricultural area. The map is composed of a grid of colored pixels representing different land cover types. The colors include various shades of green for vegetation, brown and tan for agricultural fields, and blue for water bodies like rivers and lakes. A prominent feature is a winding river or stream that cuts through the landscape. The word "Landcover" is overlaid on the map in a large, white, sans-serif font, centered in the middle-left portion of the image.

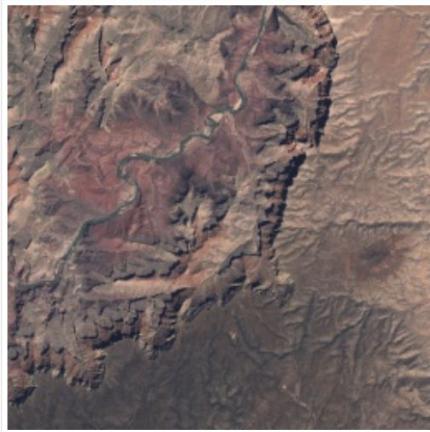
Landcover

[developers.google.com/
earth-engine/datasets/](https://developers.google.com/earth-engine/datasets/)

Hands-on: Find and use a dataset



USGS Landsat 8 Surface Reflectance Tier 1



Dataset Provider

USGS

Earth Engine Snippet

```
ee.ImageCollection("LANDSAT/LC08/C01/T1_SR")
```



Tags

global

sr

reflectance

l8sr

cloud

fmask

cfmask

lc08

landsat

usgs

DESCRIPTION

BANDS

IMAGE PROPERTIES

This dataset is the atmospherically corrected surface reflectance from the Landsat 8 OLI/TIRS sensors. These images contain 5 visible and near-infrared (VNIR) bands and 2 shortwave infrared (SWIR) bands, resulting in 7 surface reflectance bands. It includes the Operational Land Imager (OLI) bands (0.43, 0.45, 0.54, 0.64, 0.86, 0.88, 0.94 micrometers) and the Thermal Infrared Sensor (TIRS) bands (10.6, 11.5 micrometers).

LANDSAT_LC08_C01_T1_SR.js

Get Link

Save ▾

Run

Reset ▾



Inspector

Console

Tasks

```
1  /**
2   * Function to mask clouds based on the pixel_qa band of Landsat 8 SR data.
3   * @param {ee.Image} image input Landsat 8 SR image
4   * @return {ee.Image} cloudmasked Landsat 8 image
5   */
6  function maskL8sr(image) {
7    // Bits 3 and 5 are cloud shadow and cloud, respectively.
8    var cloudShadowBitMask = (1 << 3);
9    var cloudsBitMask = (1 << 5);
10   // Get the pixel QA band.
11   var qa = image.select('pixel_qa');
12   // Both flags should be set to zero, indicating clear conditions.
13   var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
14     .and(qa.bitwiseAnd(cloudsBitMask).eq(0));
```

Use print(...) to write to this console.



Layers

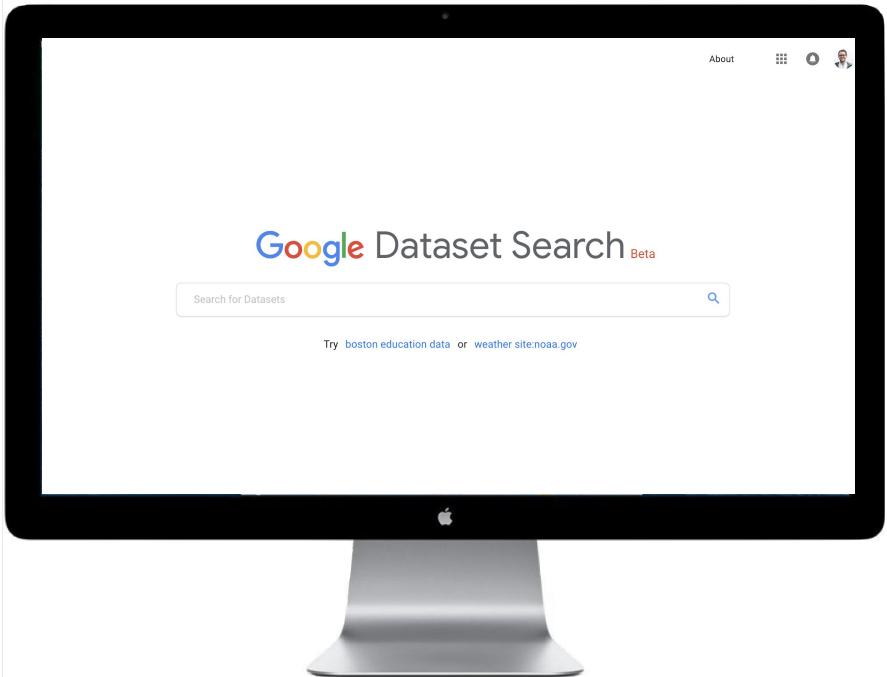
Map

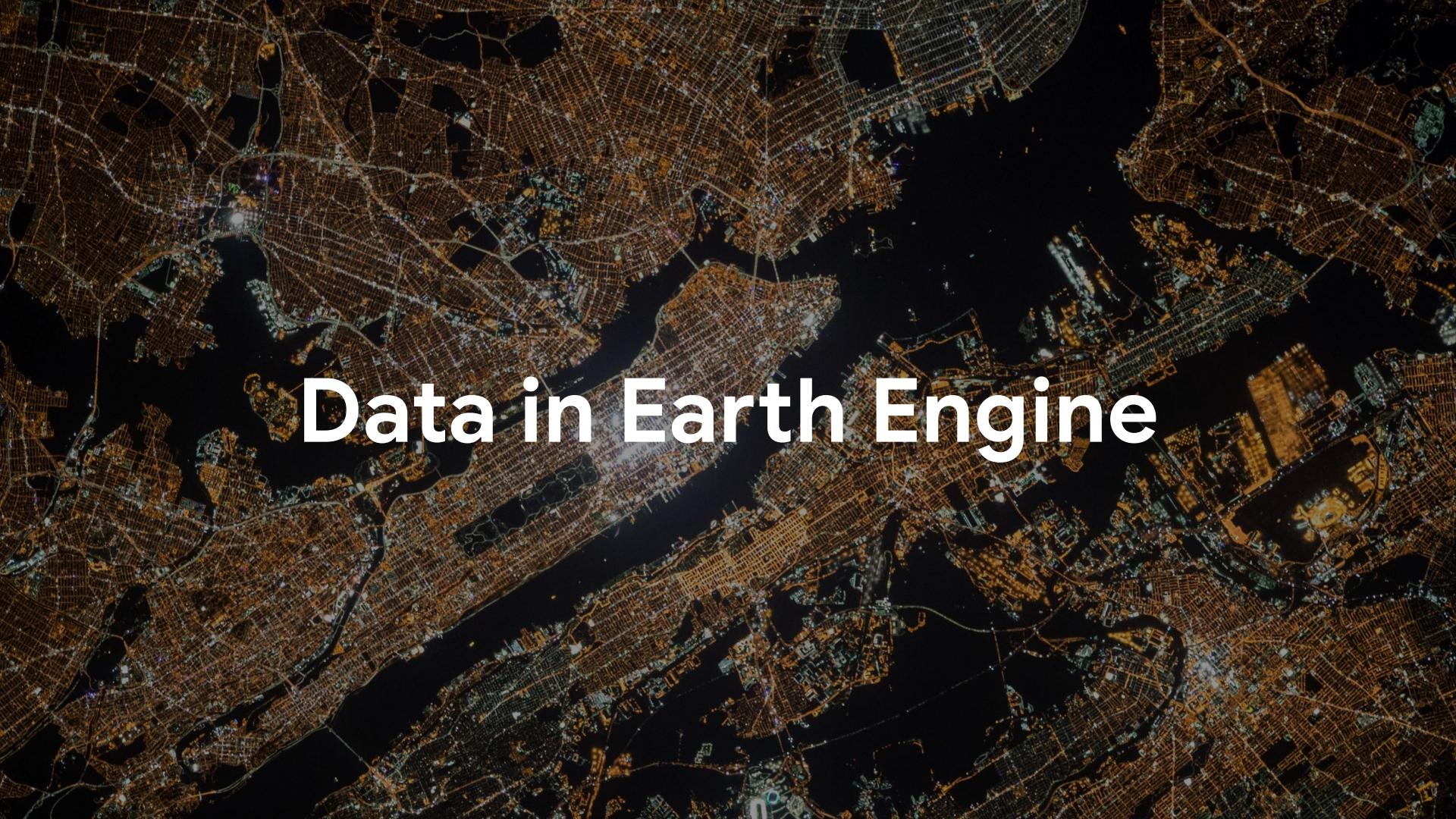
Satellite



[toolbox.google.com/
datasetsearch](https://toolbox.google.com/datasetsearch)

Hands-on: Find and use a dataset with dataset search



A nighttime satellite map of New York City, showing the illuminated city grid against the dark water of the Hudson River and East River.

Data in Earth Engine

Vector / Tables

- Geometries
 - Point
 - LineString
 - LinearRing
 - Polygon
 - MultiPoint
 - MultiLineString
 - MultiPolygon
- Geometry Collections
 - **Collection** of geometries
- Features
 - Geometry + **attributes**
- Feature Collections
 - **Collection** of features

Raster / Image

Image

- Arbitrary number of bands + band designations
- Metadata properties

Image Collection

- **Collection** of images
- Can have collection-level metadata attributes

A white fluffy puppy, possibly a Samoyed or similar breed, is lying on its stomach on a sandy beach. The puppy has light-colored eyes and a pink nose. It is wearing a dark collar with some small red and blue details. The background shows the ocean waves under a clear blue sky.

Asset Management

Asset Manager

- Code Editor assets tab
- Upload rasters
- Manage uploaded sources
- Edit asset metadata
- Create/edit image collections and folders
- Change sharing settings on images or collections
- View quota usage

Upload a new asset X

Source files ?
SELECT Please select source files for this asset.

Asset ID ?
users/username/ **assetid**

Properties ?

system:time_start	yyyy-mm-dd hh:mm:ss
stringProperty	foo
numericProperty	42
asStringProperty	13 (string)

+ Add property

Advanced

Pyramiding policy ? Masking mode ?

Mean	No-data value	No-data value, e.g. 1
------	---------------	-----------------------

[Learn more](#) about how uploaded files are processed.

OK **Cancel**

Manage asset properties

- Add property to define a property name and value.
- Values can be a string or a number.
 - Enter strings without quotes.
 - To format a number as a string, enter a single quote (') before the number.
- Click the delete icon to remove a property.

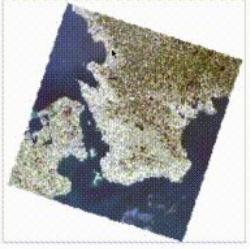
Image: denmark_landsat8

Image ID
users/herwig/eeus/denmark_landsat8

Date
Not specified

Size
99.62MB

Last modified
2016-06-08 20:43:59 UTC



Properties (0)
[+ Add property](#)

Bands (3)

Index	Name	Type	Dimensions
0	vis-red	unsigned int8	8078x8202 px
1	vis-green	unsigned int8	8078x8202 px
2	vis-blue	unsigned int8	8078x8202 px

Done **Import** **Share** **Delete**

Cube

Command-line utility

Install

See [instructions for installing in our docs](#)

Authenticate

After installing, you'll need to authenticate via the earthengine authenticate command

Use

Once authenticated, CLI supports:
Create/manage assets and collections, ACL setting
moving/deleting/copying assets, uploading raster data
And [more...](#)

Command-line asset property management

```
# update/set date of an asset
$ earthengine asset set --time_start
2016-05-01T12:34:56 \
    users/username/asset_id;

# add properties to asset
$ earthengine asset set -p
'@(number)clouds=42' \
    users/username/asset_id

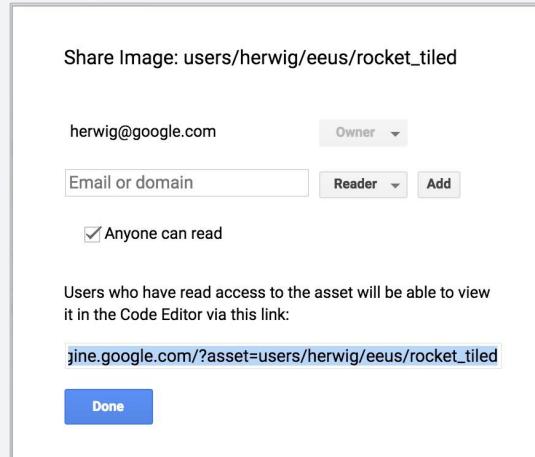
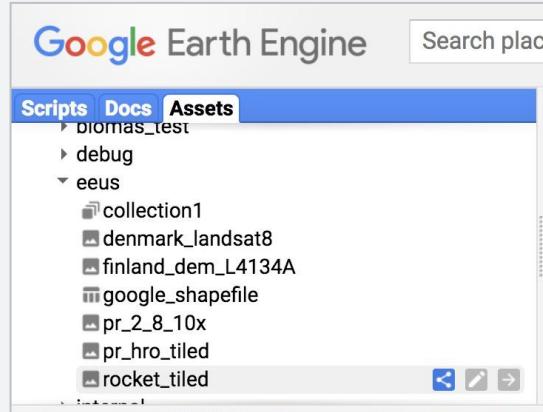
# add string property to asset
$ earthengine asset set -p
'@(string)station=Alaska' \
    users/username/asset_id
```

Share

Click the Share button to configure access to your private assets. You can also share by hovering over the asset and clicking the share icon.

The sharing dialog (Figure 4) lets you configure read or write access for individuals or groups (specified by a domain).

To make an asset public, check the 'Anyone can read' box. Any user with write access to a folder will be able to delete assets from it.

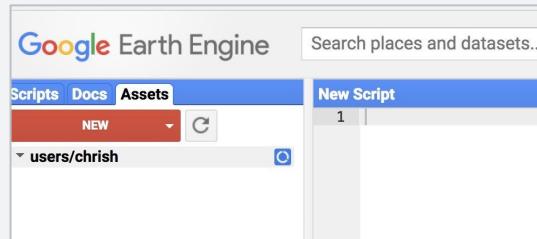
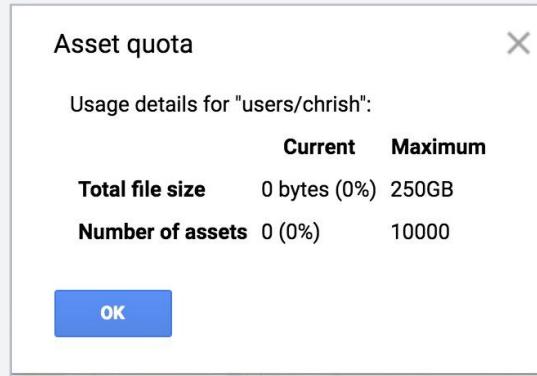


Quota

Earth Engine accounts come with 250GB of storage quota, which is independent of any Drive quota you might have.

Quota usage is measured by the total number and the compressed size of stored assets.

You can view your usage by clicking the data usage icon next to `users/username` in the Asset Manager.

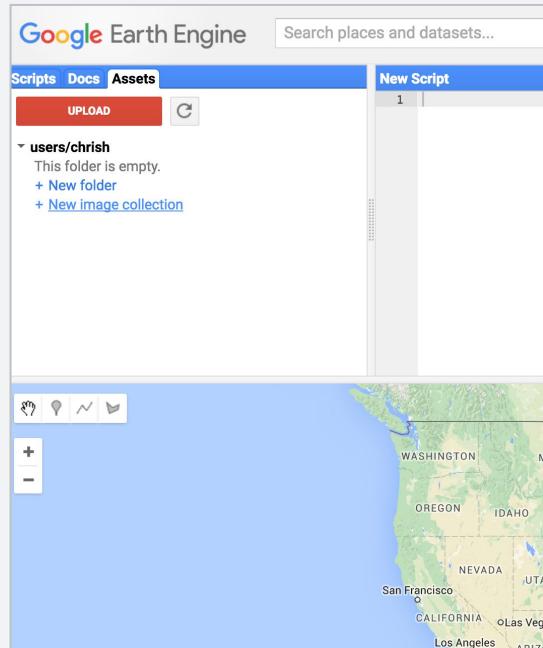


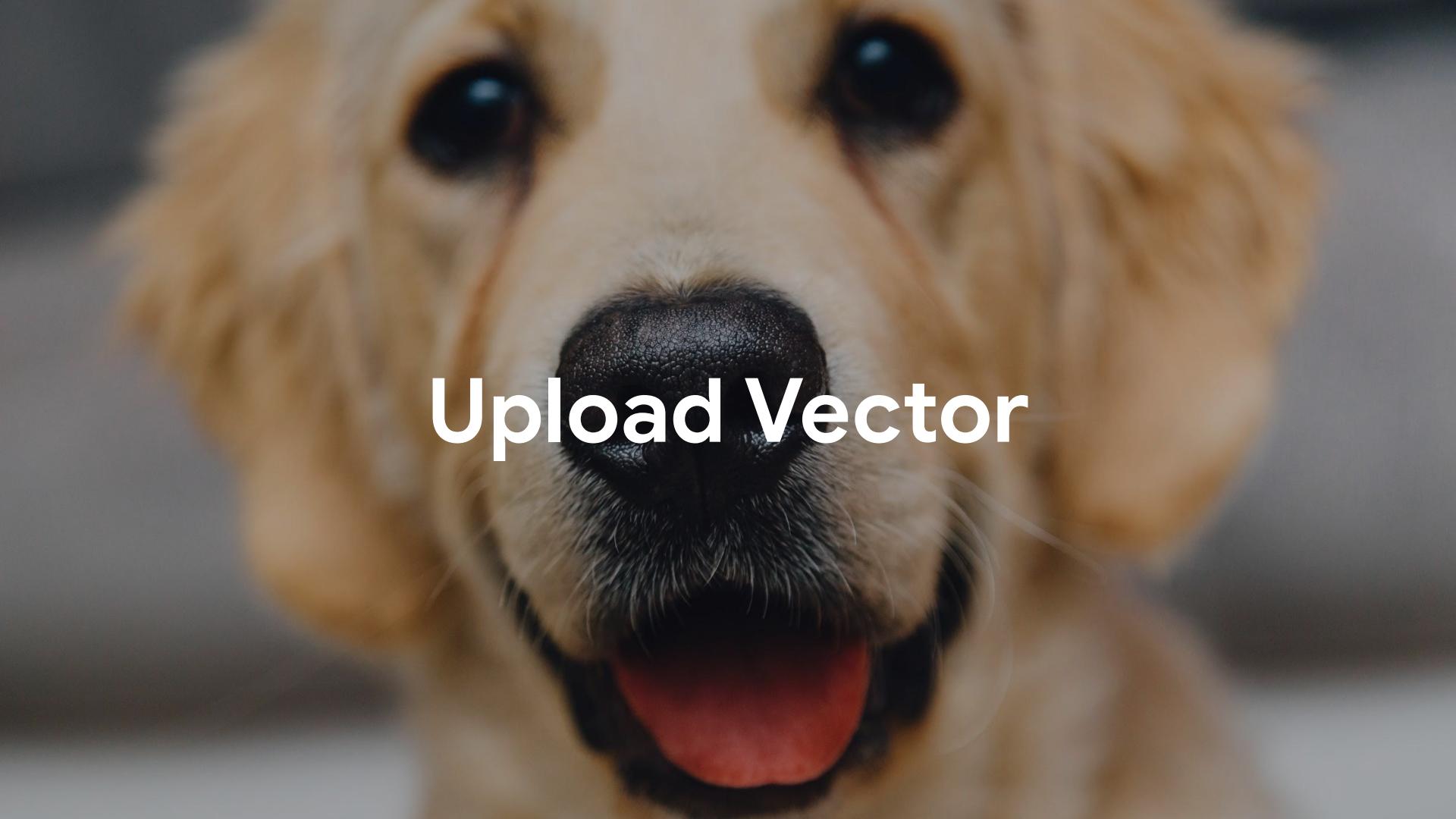
Create image collections

Move images and collections into folders with drag and drop.

To remove images from the collection, click on the collection to open dialog.

Can also use Earth Engine command line utility to create and manage image collections.



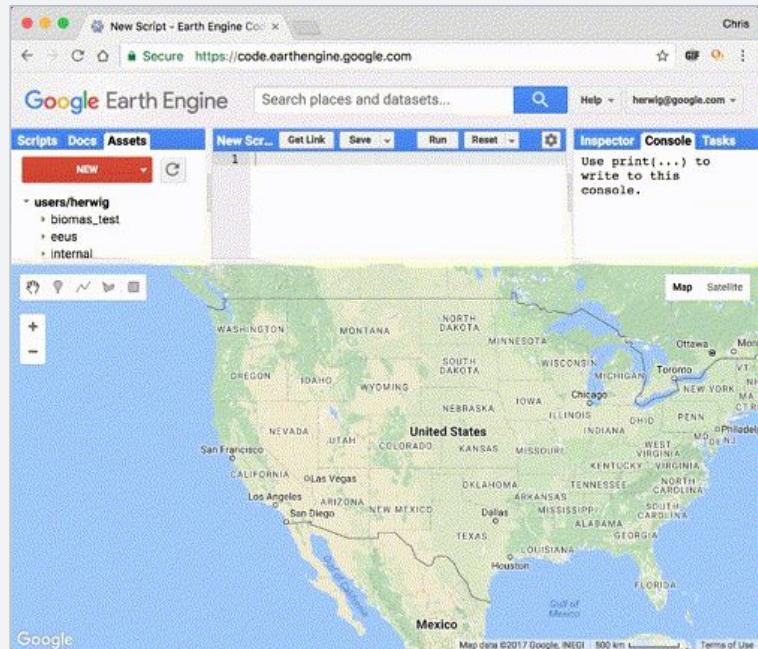
A close-up, slightly blurred photograph of a dog's face, showing its eyes, nose, and tongue.

Upload Vector

Code Editor: Upload table

Upload shapefile with sidecar files or zip file

If prj file is omitted, WGS84 is assumed.



Scripts Docs Assets

New Script Get Link Save Run Reset Inspector Console Tasks

NEW hong-kong-internat... ✓ 5m

users/herwig port-of-hong-kong... ✓ 6m

biomas_test Asset ingestion: us... ✓ 4m

eeus

internal

Older tasks

Task details: Asset ingestion: users/herwig/eeus/google_shapefile

State: Completed

Started: **16m ago** (2017-06-12 11:31:23 -0700)

Runtime: **4m**

Id: **75MBKWL573CE2PPLMBHYHTYI**

View: [Output link](#)

OK

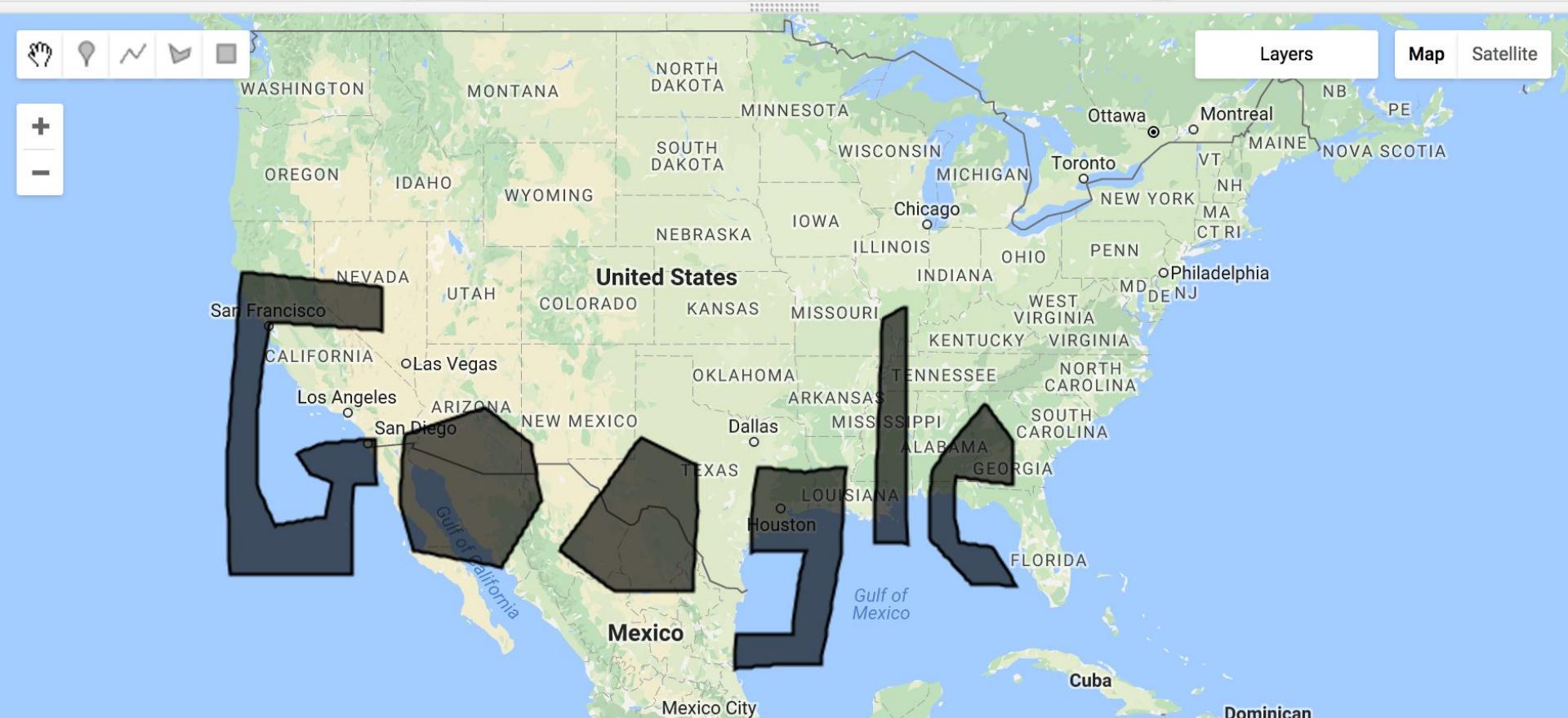
Var table. Table users/nierwig/eeus/google_smap... to this console.

▼ Private

- ▶ Watermask modis land
- ▶ aesthetics
- ▶ bugs
- ▶ external

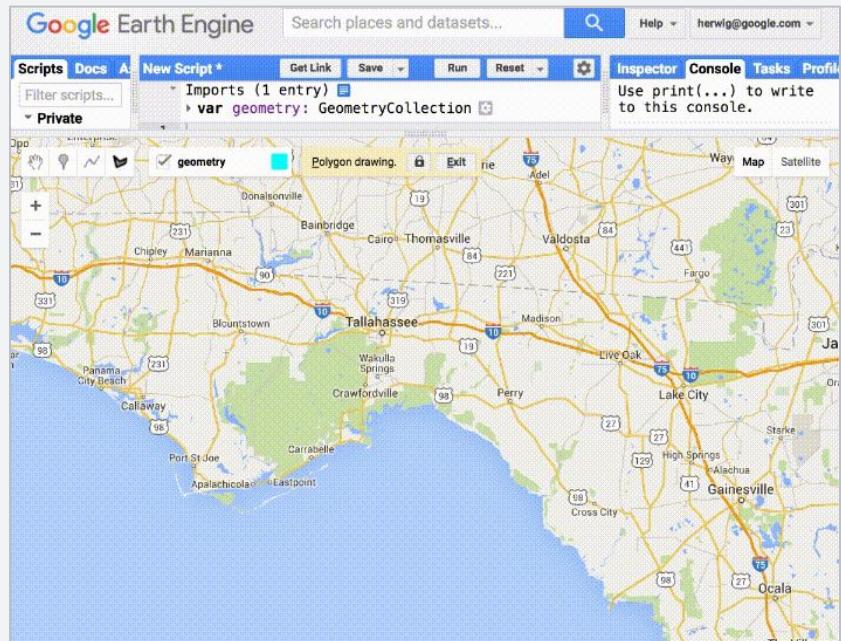
i 1 Map.addLayer(table)

2 |



Hand-drawn geometries in code editor

- Points (+Multi)
- Polygons (+Multi)
- LineStrings (+Multi)
- Geometry
- Feature
- FeatureCollection



Copy/paste GeoJSON

```
var json = {  
    "type": "FeatureCollection",  
    "columns": {  
        "system:index": "String"  
    },  
    "features": [  
        {  
            "type": "Feature",  
            "geometry": {  
                "geodesic": true,  
                "type": "Point",  
                "coordinates": [  
                    -71.015625,  
                    25.244695951306042  
                ]  
            },  
            "id": "0",  
            "properties": {}  
        }  
    ]  
};  
  
var feats =  
ee.FeatureCollection(json.features);
```

Code Editor: Upload CSV

- A row for each feature
- as many columns as there are properties or variables for the feature set.
- If features are geospatial, must have a geolocation defined by either a geometry string (GeoJSON, WKT) or x and y position properties.
- Alternatively, two columns for x and y coordinates representing point locations can be defined in a spreadsheet application and exported as CSV format along with any other variables.
- Projection assumed to be WGS84 unless otherwise specified

Upload a new table asset

Source files ⓘ
SELECT Please drag and drop or select files for this asset.

Asset ID ⓘ
users/username/

Advanced

Character Encoding ⓘ
UTF-8

CRS ⓘ
EPSG:4326

Geodesic ⓘ
 Infer from CRS

Primary Geometry Column ⓘ
geometry

X Column ⓘ Y Column ⓘ
longitude latitude

Date Format ⓘ
yyyy-MM-dd'T'HH:mm:ss.SSSZ

CSV Delimiter ⓘ CSV Qualifier ⓘ
. *

Maximum Error ⓘ
1.0

Maximum Vertices ⓘ
1000000

[Learn more](#) ⓘ about how uploaded files are processed.

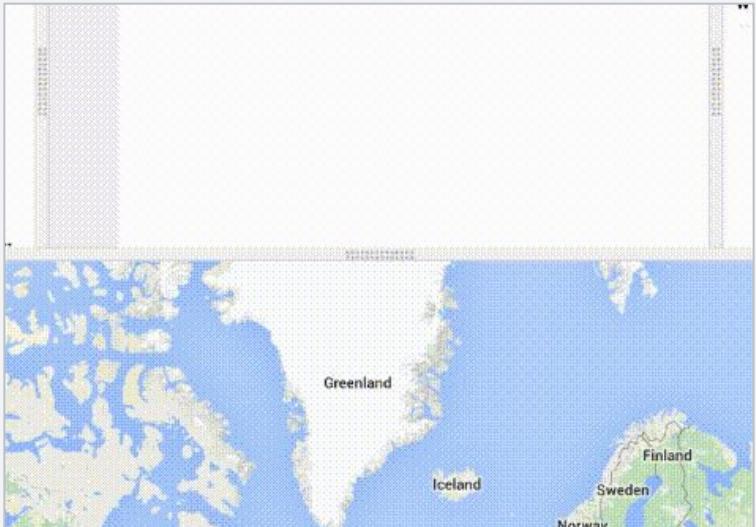
CANCEL **OK**

A close-up photograph of a small, fluffy white puppy with dark brown spots on its head and back. The puppy is lying down, facing the camera, and is positioned next to a light-colored, textured wall, possibly made of concrete or stone. The lighting is soft, creating a warm atmosphere.

Upload Raster

Upload raster

- Geotiff
- 10GB per-file maximum
- Can set properties
- Choose pyramiding policy
- Set optional no data value
- See Advanced image upload options
- Tiled uploads supported, see docs



Command-line: Upload raster

- Geotiff(s) on Google Cloud Storage
- 10GB per-file maximum
- (optional) Set pyramiding policy
- (optional) Set no data value
- Can also set metadata properties at time of upload.
- See upload section of Command Line Usage docs

```
assetid="users/herwig/eeus/mytestupload";
earthengine upload image \
--asset_id="${assetid}" \
--pyramiding_policy=sample \
--nodata_value=0 \
gs://bucket/image.tif \
gs://bucket/image2.tif
```

Try it out

```
var importedAsset =  
ee.Image('users/herwig/eeus/pr_hro_  
tiled');  
Map.setCenter(-66.090923,  
18.375517, 16);  
Map.addLayer(importedAsset);
```

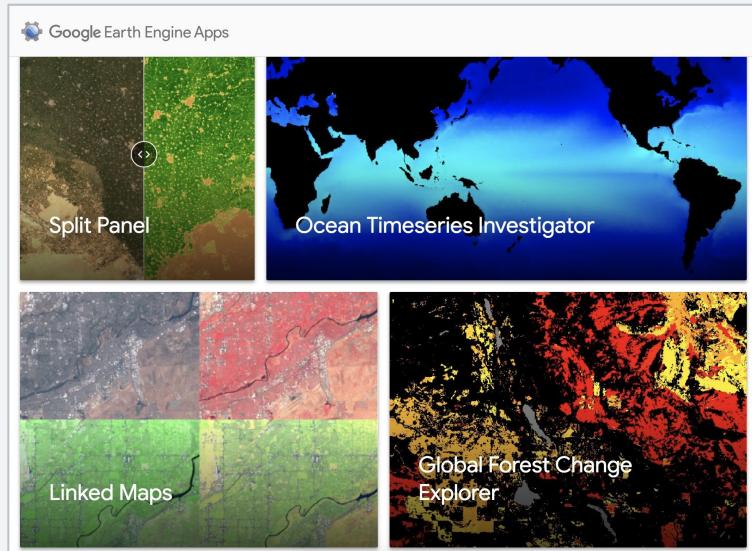
Questions

A black and white dog with a white patch on its chest is swimming in water, creating a large, dynamic splash of water droplets around it. The background is a warm, out-of-focus yellowish-brown.

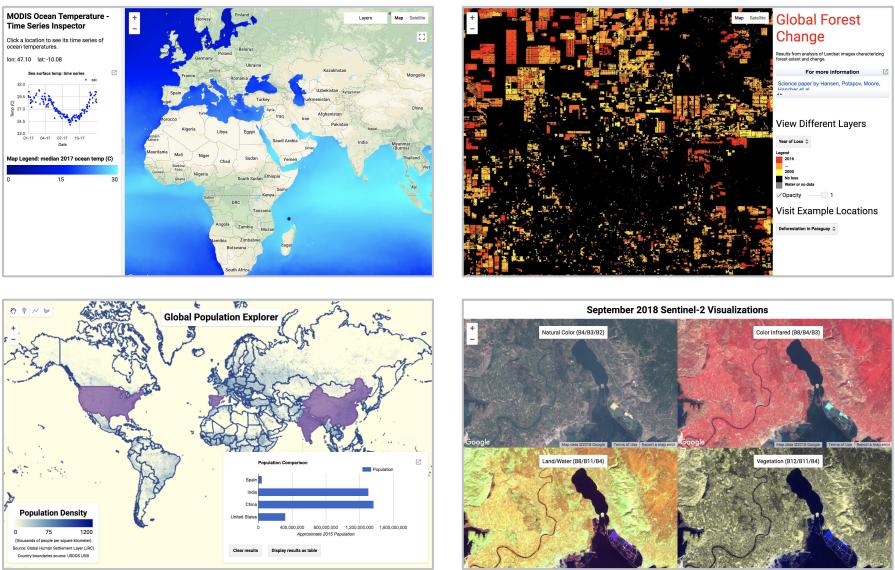
Earth Engine Apps

Earth Engine Apps

- Earth Engine Apps are **dynamic, publicly accessible user interfaces** for your analysis.
- You can share with whomever you like – **no Earth Engine account required**.
- You control what that viewport looks like, thanks to the broad suite of widgets, UI components, and interactivity tools built into the User Interface API.



<https://earthengine.app>



Earth Engine App

developers.google.com/earth-engine/apps

Publish New App

Owner

users/abcd ▾

App Name ?

Burning Bright

Your App ID will be burning-bright [Edit ▾](#)

Google Cloud Project API Key ?

[Click here](#) to create an API Key, and paste it below.

AlzaSyDTCBb8cLdtFgnpR2D2OzKufhPcp30d7Z4

URL: <https://abcd.users.earthengine.app/view/burning-bright>

Feature this app in your [Apps Gallery](#)

[x](#) Reset Thumbnail

Description (Optional)



This app was created by me using Google Earth Engine. It maps changes on the Earth's surface at a global scale.

Source Code ?

Current contents of editor

Repository script path

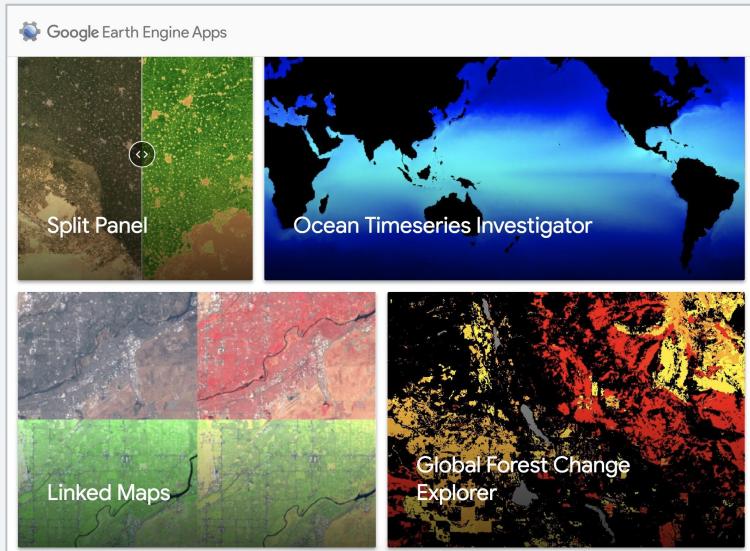
When an app is published, it's public, and anyone can view it. The published source code will be publicly readable. All assets must also be shared publicly for the app to display properly.

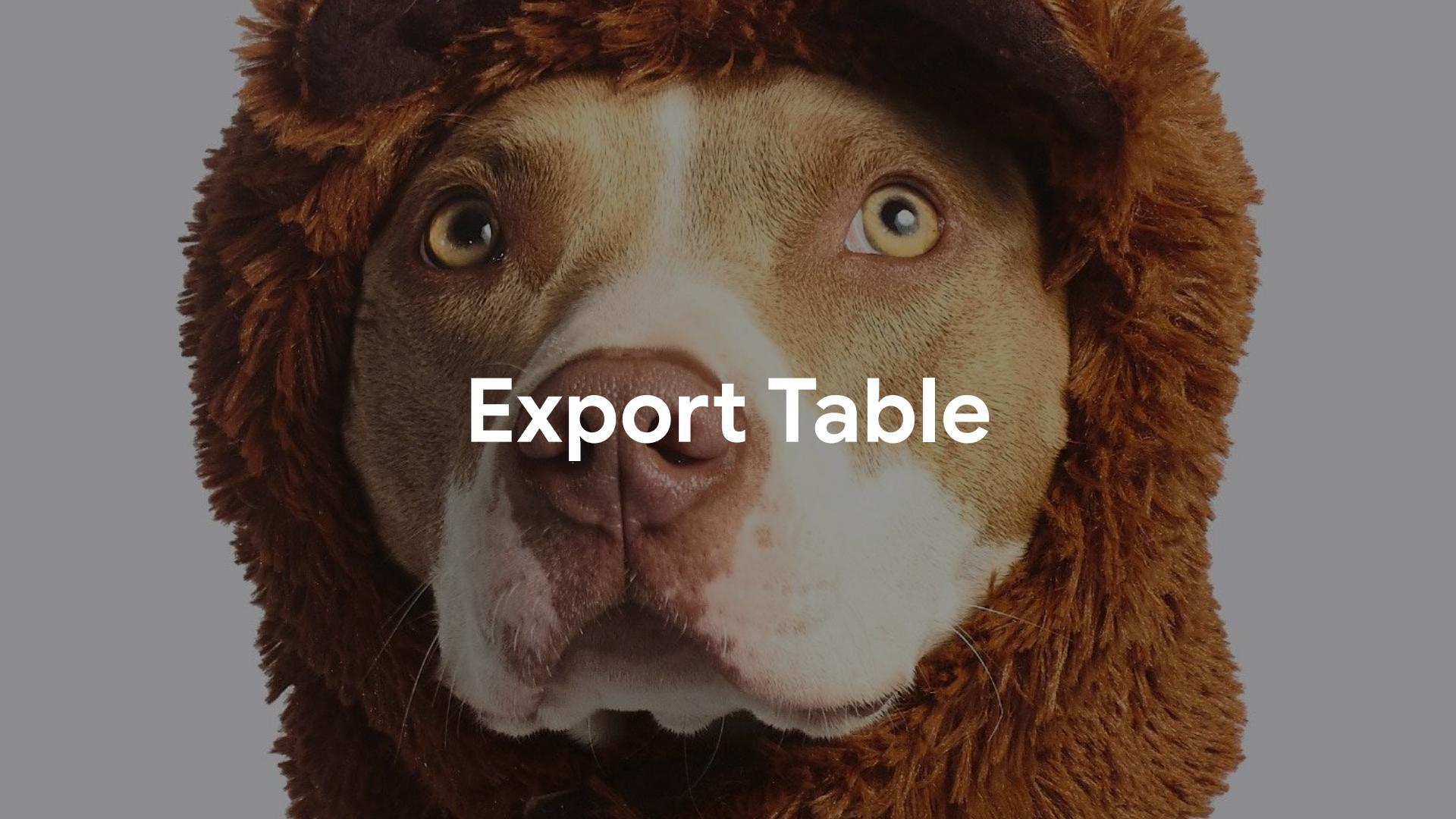
CANCEL

PUBLISH

Earth Engine Apps

- Apps work best for research and scientific collaboration specialized audiences, so please keep in mind that they may not scale well to wide audiences.
- Similar to Earth Engine's per-user quota, Apps have usage quotas on concurrent queries.
- App performance will depend on the computational intensity of the specific app, with simpler apps able to serve more concurrent users than complex ones.
- For more on quota and tips on how to optimize your App, check out the [Apps FAQ](#).





Export Table

Export table to Drive or CloudStorage

You can export a FeatureCollection as CSV, SHP (shapefile), GeoJSON, KML, or KMZ using Export.table.

Format-specific constraints:

- KML: output FeatureCollection will have all the geometries transformed to unprojected (WGS84) coordinates.
- SHP: output FeatureCollection must contain features with the same geometry type and projection and must fit within the Shapefile size limits (2Gb). Column names are truncated to 10 characters or fewer, and this must not create duplicate column names.

Export table to Asset

There are several limitations on the size and shape of Earth Engine table assets:

- Maximum of 100 million features
- Maximum of 1000 properties (columns)
- Maximum of 100,000 vertices for each row's geometry
- Maximum of 100,000 characters per string value

```
// Export a table to Earth Engine table
asset
Export.table.toAsset({
  collection: features,
  description: 'exportToTableAssetExample',
  assetId: 'exampleAssetId',
});
```



On-the-fly
GIFs

Export table to Asset

There are several limitations on the size and shape of Earth Engine table assets:

- Maximum of 100 million features
- Maximum of 1000 properties (columns)
- Maximum of 100,000 vertices for each row's geometry
- Maximum of 100,000 characters per string value

```
// Export a table to Earth Engine table
asset
Export.table.toAsset({
  collection: features,
  description: 'exportToTableAssetExample',
  assetId: 'exampleAssetId',
});
```



Export Image

Export image

- Specify region to export or use native image bounds
- Scale (can only specify one): nominal scale of desired bands
- All bands must be of same pixelType - so cast to best fit before export.
- Ex: Cast image of bands [int16 and int32] to int32.

```
var vis = {  
  min: 500,  
  max: 2400,  
  gamma: 1.4,  
  bands: ['B4', 'B3', 'B2']  
};  
var denmark = ee.Image('COPERNICUS/S2/20160512T102029_20160512T160424_T33UUB');  
// use image extents as region argument given to export  
// obtain native scale  
  
var scale = denmark.select(vis.bands).projection()  
  .nominalScale(). getInfo();  
  
var aoi = denmark.geometry(scale);  
// add rgb styled S2 image to map to preview.  
  
Map.addLayer(denmark, vis);  
  
// export raw S2 image with all original bands  
Export.image.toDrive({  
  image: denmark.toInt32(),  
  description: 'denmark-s2-source-data',  
  region: aoi,  
  scale: scale,  
});
```

Cloud-optimized GeoTIFF

You can specify image output format (if the destination is not `toAsset()`) with the `fileFormat` parameter (currently only the default, 'GeoTIFF', is enabled).

Other configuration options are set with the `formatOptions` parameter, which should be a dictionary keyed by other format options (currently only `cloudOptimized`, to export a [cloud-optimized GeoTIFF](#), is enabled and is **false** by default).

```
// Export a cloud-optimized GeoTIFF.  
Export.image.toDrive({  
  image: landsat,  
  description:  
  'imageToCOGeoTiffExample',  
  scale: 30,  
  region: geometry,  
  fileFormat: 'GeoTIFF',  
  formatOptions: {  
    cloudOptimized: true  
  }  
});
```

Exporting images as they appear in the Code Editor

- `Map.addLayer(image, visParams);`
- The `image.visualize()` method converts an image into an 8-bit RGB image for display or export.
- Image displayed in code editor is EPSG:3857.
- See [visualization parameters in docs](#).

```
var denmark = ee.Image('LANDSAT/LC8_SR/LC81940212014072');

var vis = {
  min: 100,
  max: 2000,
  gamma: 1.5,
  bands: ['B4', 'B3', 'B2']
};
// visualize image using visOpts above, return an 8bit RGB image
denmark = denmark.visualize(vis);

// obtain native scale of RGB bands
var scale = denmark.projection().nominalScale(). getInfo();
// use image extents as region argument given to export
var aoi = denmark.geometry(scale).bounds();

// add an alpha channel as 4th band to mask no-data regions
var mask = denmark.mask().reduce(ee.Reducer.min())
  .multiply(255).toByte();

denmark = denmark.addBands(mask);

Export.image.toDrive({
  image: denmark,
  description: 'denmark-l8-rgb',
  region: aoi,
  scale: scale,
});
```

Export Image to Cloud Storage

To use Google Cloud Storage, you'll need to set up a project, enable billing for the project, and create a storage bucket.

See the [Cloud Storage Quickstart page](#) for instructions.

Data exported to a Cloud Storage bucket will have the bucket's default object Access Control List (ACL).

You must have write permission for the specified bucket.

```
// Export a cloud-optimized GeoTIFF.  
Export.image.toDrive({  
  image: landsat,  
  description:  
  'imageToCOGeoTiffExample',  
  scale: 30,  
  region: geometry,  
  fileFormat: 'GeoTIFF',  
  formatOptions: {  
    cloudOptimized: true  
  }  
});
```

Export Image to Earth Engine Asset

- To export an image to an asset in your Earth Engine assets folder, use Export.image.toAsset().
- Use the Asset Manager in the code editor to monitor your storage quota, and manage your Earth Engine assets.
- New parameter: assetId
- pyramidingPolicy

```
var denmark = ee.Image('LANDSAT/LC8_SR/LC81940212014072');

var vis = {
  min: 100,
  max: 2000,
  gamma: 1.5,
  bands: ['B4', 'B3', 'B2']
};
// visualize image using visOpts above
// turning it into 8-bit RGB image.
denmark = denmark.visualize(vis);

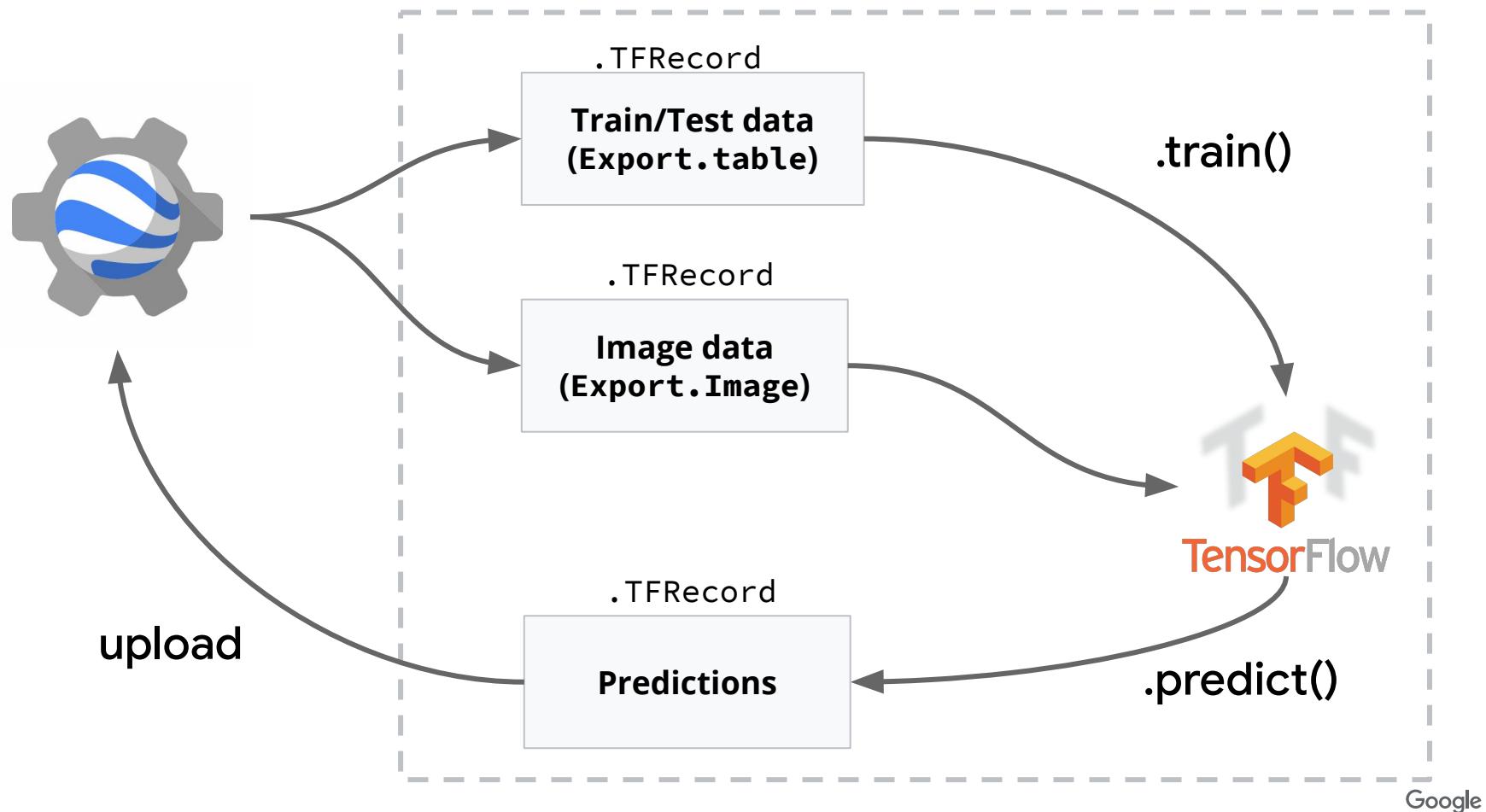
// obtain native scale of RGB bands
var scale = denmark.projection().nominalScale(). getInfo();

// use image extents as region argument given to export
var aoi = denmark.geometry(scale).bounds();

Export.image.toAsset({
  image: denmark,
  assetId: 'users/herwig/eeus/denmark_landsat8',
  region: aoi,
  scale: scale,
  description: 'exportToAsset-Denmark-l8',
  pyramidingPolicy: {'default': 'sample'}
});
```

A photograph of two young golden retriever puppies sitting in a field of small, colorful flowers. One puppy is facing the camera with its tongue slightly out, while the other is facing right with its mouth open. The background is a soft-focus green field.

Export TFRecord



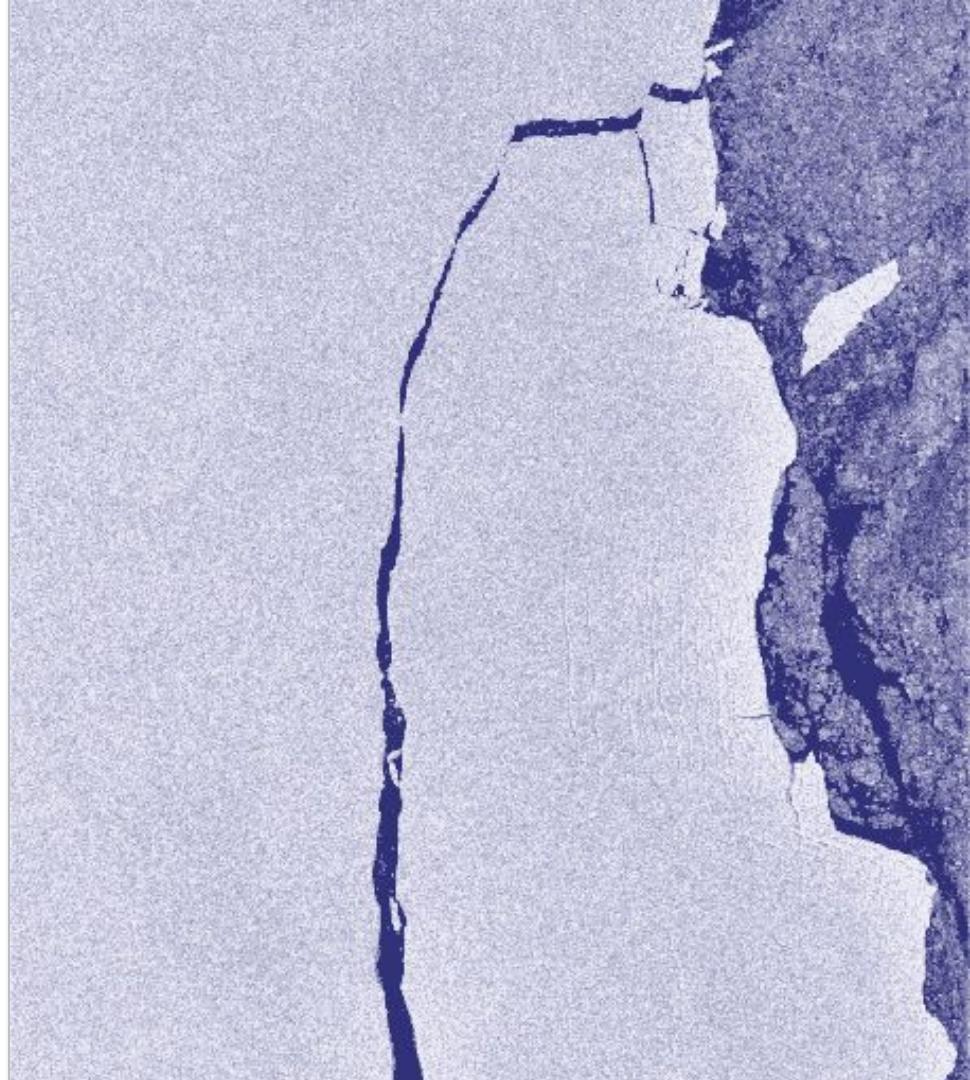
Property	Description
<code>patchDimensions</code>	Patch dimensions tiled over the export area, covering every pixel in the bounding box exactly once.
<code>kernelSize</code>	The size of the kernel, in pixels, as applied over a patch in the first layer of a CNN.
<code>compressed</code>	If true, compresses the .tfrecord files with gzip and appends the ".gz" suffix
<code>maxFileSize</code>	Maximum size, in bytes, for an exported .tfrecord (before compression). A smaller file size will result in greater sharding (and, thus, more output files).
<code>defaultValue</code>	The value set in each band of a pixel that is partially or completely masked, and the value set at each value in an output 3D feature made from an array band where the array length at the source pixel is less than the depth of the feature value
<code>tensorDepths</code>	List corresponding to the array bands of the input that specifies the desired depth of the 3d tensors.
<code>sequenceData</code>	If true, each pixel is output as a SequenceExample mapping scalar bands to the context and array bands to the example's sequences
<code>collapseBands</code>	If true, all bands will be combined into a single 3D tensor, taking on the name of the first band in the image.
<code>maskedThreshold</code>	Maximum allowed proportion of masked pixels in a patch.



Export Video

Export Video to Drive

- Export ordered image collections as video, where frames are defined by images in the collection
- Specify frame rate, scale (or) dimensions.
- MP4.
- Images in ImageCollection must be 8-bit RGB images.
- Can take a significant amount of time to complete.



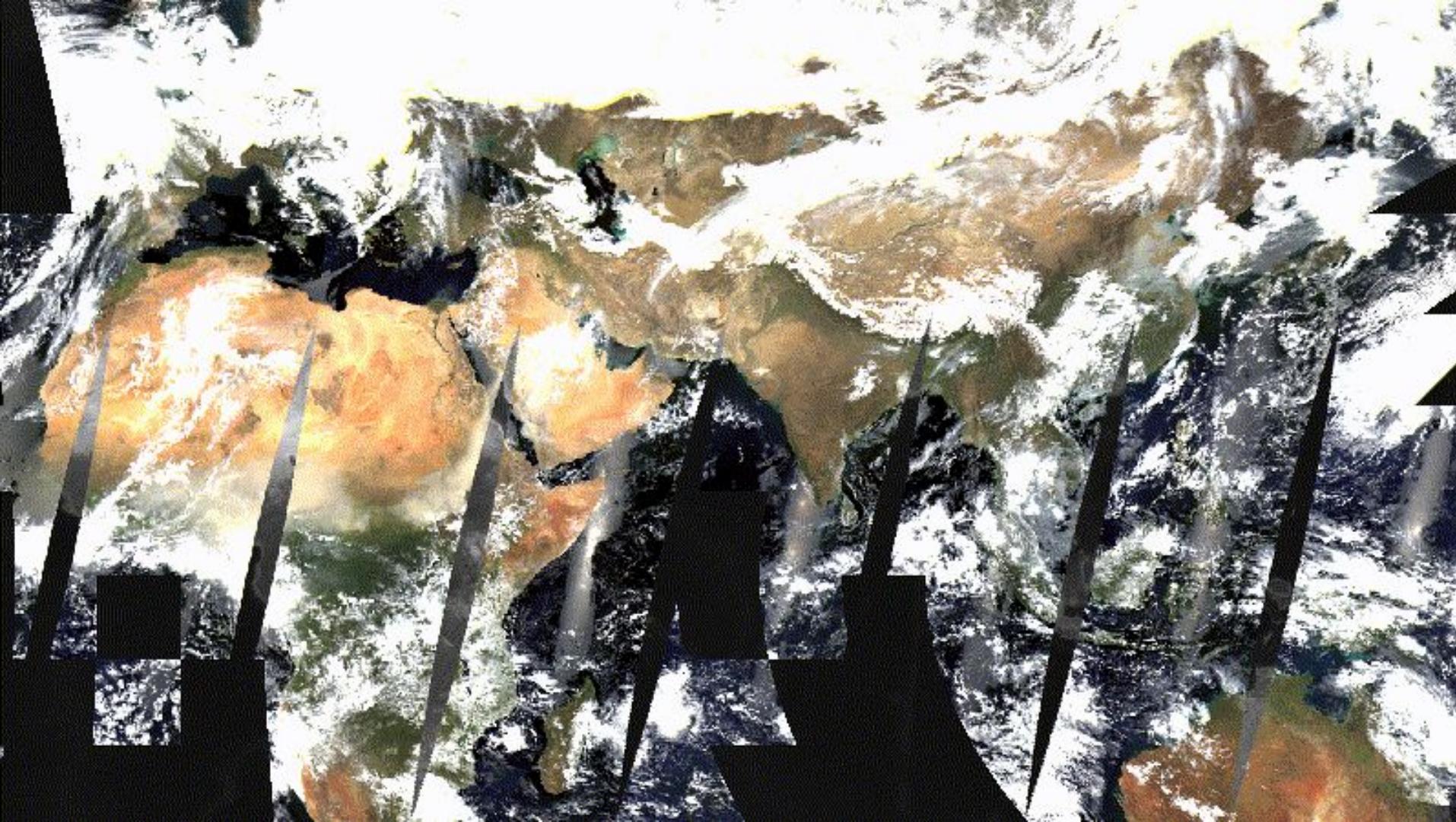


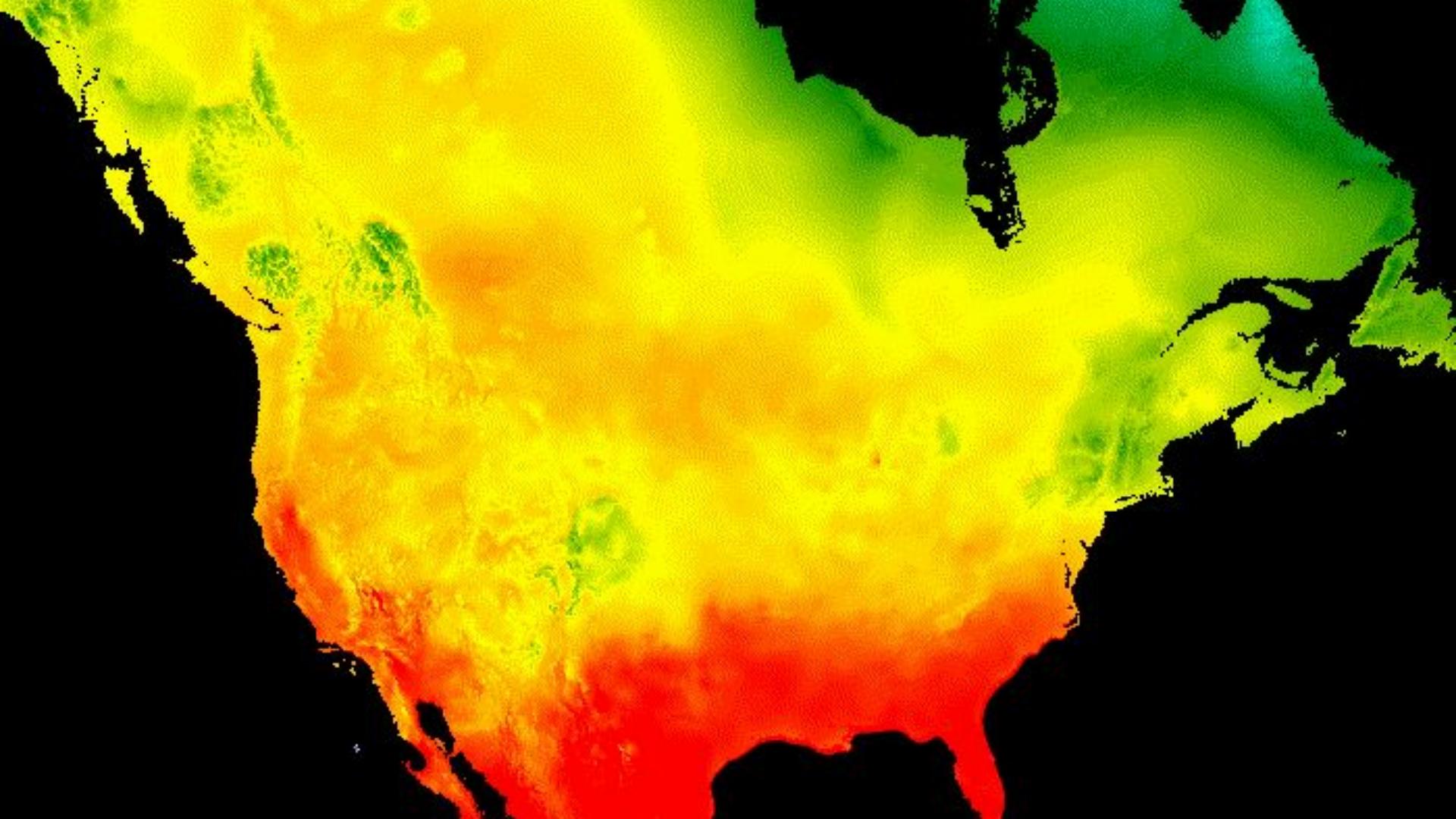
```
var region = ee.Geometry.Point([151.213, -33.87]);

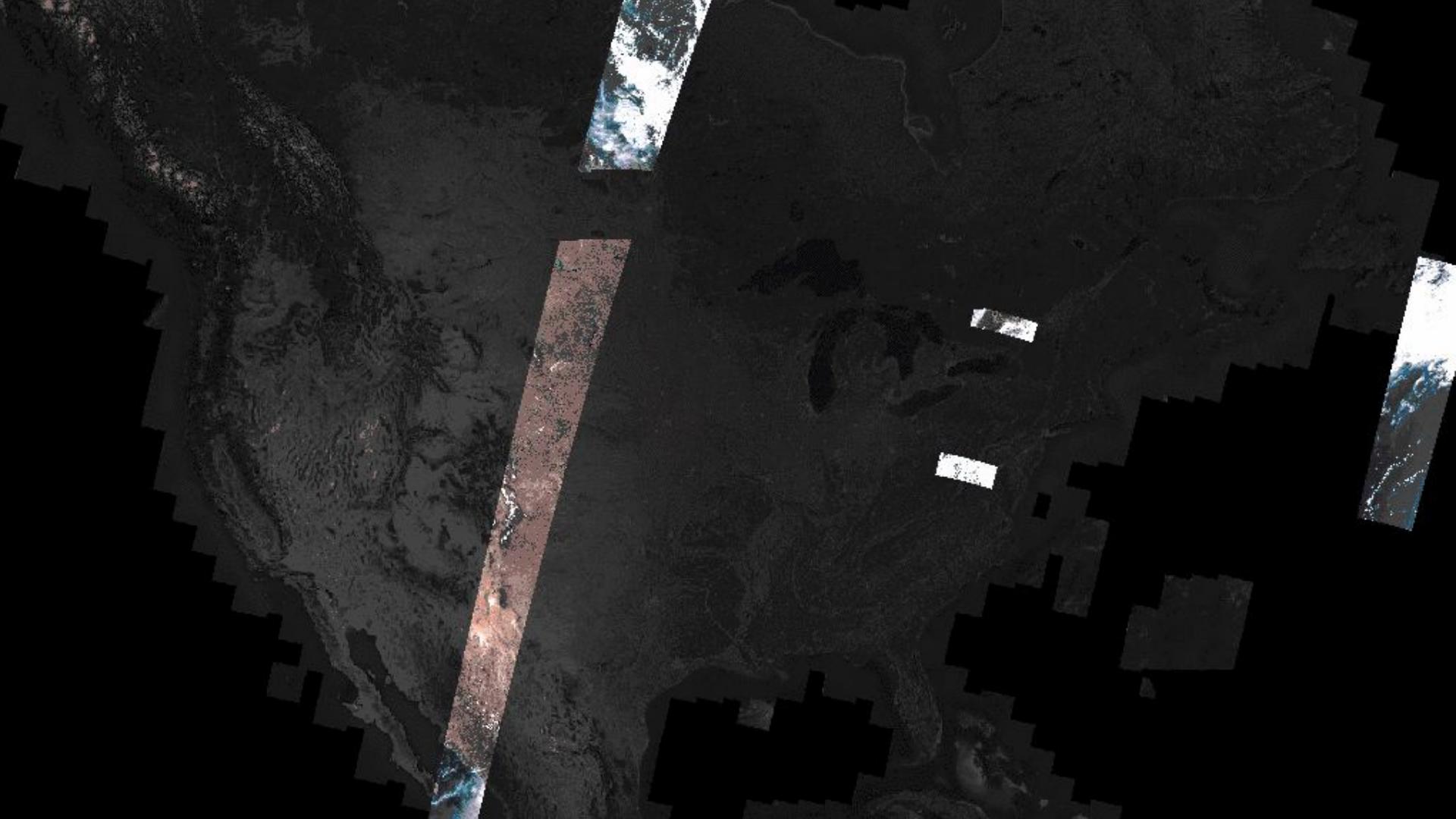
var vis = {
  min:500,
  max:3000,
  gamma:1.4,
  bands: ['B4', 'B3', 'B2']
}

var collection = ee.ImageCollection('COPERNICUS/S2')
  .filterBounds(region)
  .sort('system:time_start')
  // Need to make the data 8-bit.
  .map(function(image) {
    return image.visualize(vis);
});
```

```
Export.video.toDrive({
  collection: collection,
  description: 'sydney-5fps',
  dimensions: 1080,
  framesPerSecond: 5,
  region: region
});
```







Export Video to Cloud Storage

- To use Google Cloud Storage, you'll need to set up a project, enable billing for the project, and create a storage bucket.
- See the [Cloud Storage Quickstart page for instructions](#).

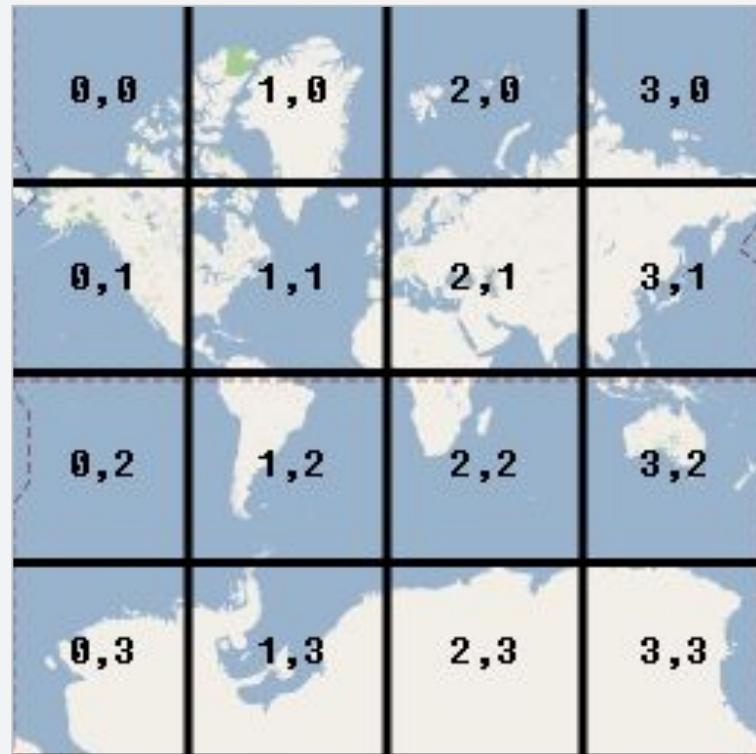
```
Export.video.toCloudStorage({  
  collection: collection,  
  description: 'videoToGCS',  
  bucket: bucketName,  
  dimensions: 1080,  
  framesPerSecond: 1,  
  fileNamePrefix: 'myvideos/',  
  region: region  
});
```

A bulldog with a tan and white coat and a black harness stands in shallow, foamy ocean water. Waves are crashing behind it, creating white foam. The background shows more of the ocean and sky.

Export Map Tiles

Export Map Tiles

- `Export.map.toCloudStorage()` exports map tiles suitable for display using the Google Maps API and within Google Earth
- Exported image must have either 1 or 3-bands.
- Output Map is
 - publicly viewable*
 - can be embedded on other pages
 - does not require the viewer to be a registered Earth Engine user.
- User exporting must be writer or owner of bucket.



*Unless you choose not to write public tiles (`writePublicTiles=false`), in which case the tiles inherit the bucket's default object ACL

Export Map Tiles

- fileFormat: map tiles' format jpg/png/auto ["auto"]
- path: The string used as the output tiles prefix[task description].
- writePublicTiles: write public tiles or use bucket's default ACL. [true]
- maxZoom: maximum zoom level tiles to export
- scale: The max image resolution in meters per pixel, as an alternative to maxZoom
- minZoom: min map tile level to export. [zero]
- See [scale and zoom docs](#)

Export Map Tiles

Zoom Level	Pixel Size (at equator)
0	156 km
1	78 km
2	39 km
3	20 km
4	10 km
5	4.9 km
6	2.4 km
7	1.2 km
8	611 m
9	305 m
10	152 m
11	76 m
12	38 m
13	19 m
14	9.6 m
15	4.8 m

Export Map Tiles

15/10367/14681



15/10368/14681

15/10367/14682



15/10368/14682

Export Map Tiles

```
var image =
ee.Image("users/herwig/eeus/rocket_
tiled");
Map.addLayer(image);

var southBounds =
ee.Geometry.Polygon(
[-180, 85, 0, 85,
 179.998, 85, 179.998, -85,
 0, -85, -180, -85
], null, false);

Export.map.toCloudStorage({
  image: image,
  bucket: 'herwig-export-tests',
  description: 'export-rocket',
  path: 'maps/20180118/rocket',
  minZoom: 0,
  maxZoom: 6,
  region: southBounds,
});
```

```
1 var image = ee.Image("users/herwig/eeus/rocket_tiled");
2 Map.addLayer(image)
3
4 var southBounds = ee.Geometry.Polygon([-180, 85, 0, 85, 179.998, 85, 179.998, -85, 0, -85])
5
6 Export.map.toCloudStorage(
7   {
8     image: image,
9     bucket: 'herwig-export-tests',
10    description: 'export-rocket',
11    path: 'maps/20180118/rocket',
12    minZoom: 0,
13    maxZoom: 6,
14    region: southBounds,
15  })
})
```

export-rocket

RUN

export-rocket

✓ 8m

Task details: export-rocket



State: Completed

Started: **10m ago** (2018-06-07 14:15:00 -0700)

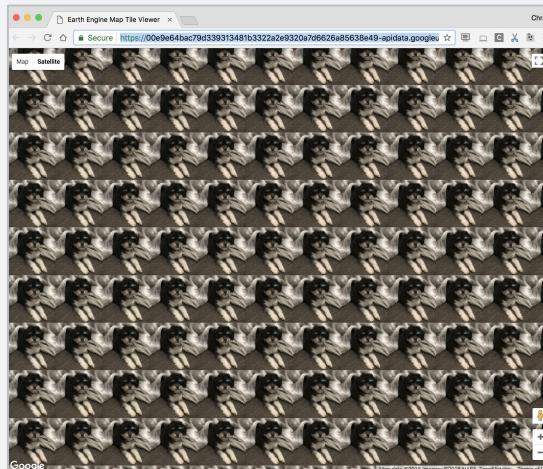
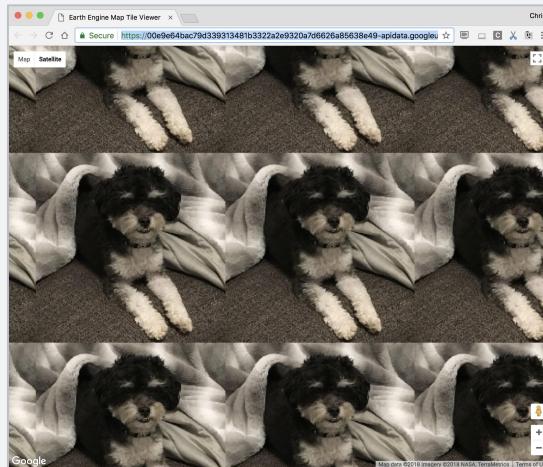
Runtime: **8m**

Id: **RGBUV5QVQA3GX7IOWRDUAHCQ**

[View task](#) | [Source script](#) [Open in GCS](#) [Open in Google Earth](#)

OK

It's a map!



Google

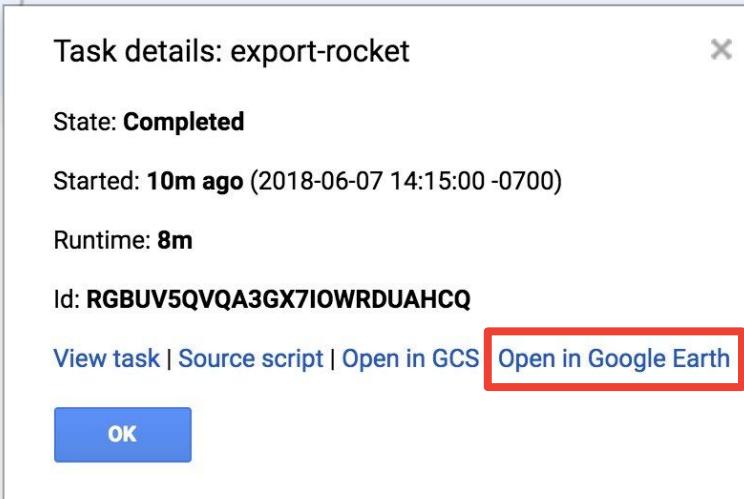
```
1 var image = ee.Image("users/herwig/eeus/rocket_tiled");
2 Map.addLayer(image)
3
4 var southBounds = ee.Geometry.Polygon([-180, 85, 0, 85, 179.998, 85, 179.998, -85, 0, -85]);
5
6 Export.map.toCloudStorage(
7   {
8     image: image,
9     bucket: 'herwig-export-tests',
10    description: 'export-rocket',
11    path: 'maps/20180118/rocket',
12    minZoom: 0,
13    maxZoom: 6,
14    region: southBounds,
15  })
})
```

export-rocket

RUN

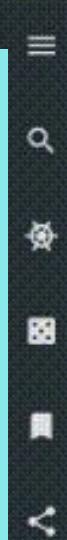
export-rocket

✓ 8m

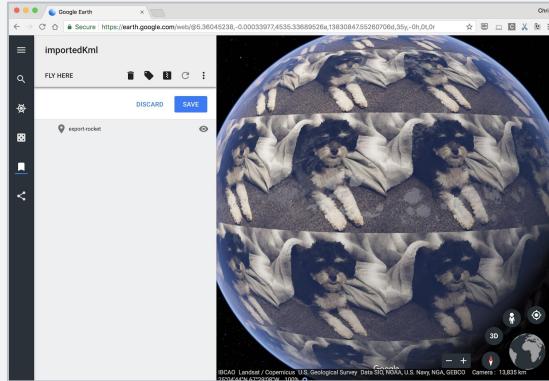
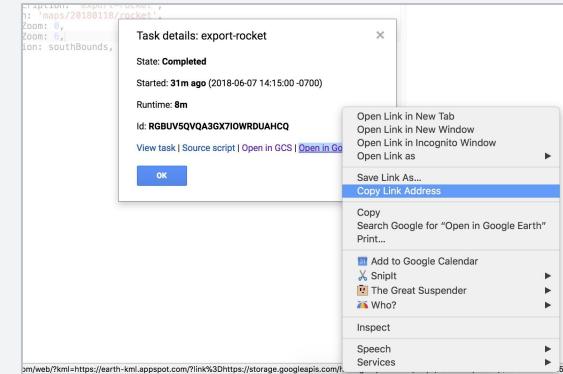




Globe



Share Earth link



Cloud + Maps Platform Credits Programs

Cloud: Geo for Good Cloud Credits Program:

g.co/earth/cloud-credits

Maps: Eligible nonprofits, startups, crisis response, and news media organizations may apply for Google Maps Platform credits through the following programs:

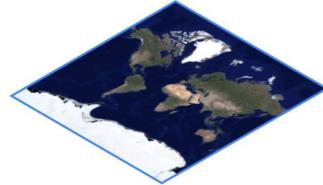
- Nonprofits can apply for [Google for Nonprofits](#)
- Startup organizations can apply to the [Google Cloud Startup Program](#)
- Crisis response organizations can learn more about our [Crisis Response efforts](#)
- News media organizations can learn more about the [Google News Initiative](#)

Google

A black and white dog, possibly a Border Collie, is captured mid-splash in a body of water. The dog's head is above the surface, with water droplets flying around it. Its front paws are partially submerged. The background is dark and out of focus.

Key Concepts

Key Concepts



Pyramiding Policy

How each pixel at a given level of the pyramid is computed from the aggregation of a 4x4 block of pixels at the next lower level.

Continuous valued images: mean

Discrete valued images: sample (usually the top left pixel) of pixels at the next lower level.

Key Concepts

Access Control Lists (ACL)

By default, assets users upload to their Earth Engine account are private -- accessible only to that user. Users can change access control settings on individual assets or collections. Update ACLs using Asset Manager in Code Editor or via command line `earthengine acl` command

scale

(in meters) Ex. 30

dimensions

dimensions (in pixels): width and height for output image, Ex: '1024x768'

Key Concepts

crs

Coordinate Reference System. Ex:
'EPSG:4326'

crsTransform

A list of affine transformation parameters in row-major order
[xScale, xShearing,
xTranslation, yShearing,
yScale, yTranslation]
If not explicitly specified, the CRS of the output will be taken from the first band of the image to be exported.
crsTransform, scale, and dimensions are mutually exclusive.

region

(ee.Geometry) region of image to export

maxPixels

The `maxPixels` parameter is intended to prevent very large exports from inadvertently being created.

Increase `maxPixels` if the default value is too low for your intended output image.

If the output image file is large, it will be exported as multiple tiles, with each named `baseFilename-yMin-xMin`. `maxPixels` default is `1e8`.