Exploring Images (F1.1)

Author
Jeff Howarth

Overview

Satellite images are at the heart of Google Earth Engine's power. This chapter teaches you how to inspect and visualize data stored in image bands. We first visualize individual bands as separate map layers and then explore a method to visualize three different bands in a single composite layer. We compare different kinds of composites for satellite bands that measure electromagnetic radiation in the visible and non-visible spectrum. We then explore images that represent more abstract attributes of locations, and create a composite layer to visualize change over time.

Learning Outcomes

- Using the Code Editor to load an image
- Using code to select image bands and visualize them as map layers
- Understanding true- and false-color composites of images
- Constructing new multiband images.
- Understanding how additive color works and how to interpret RGB composites.

Assumes you know how to:

- Sign up for an Earth Engine account, open the Code Editor, and save your script (Chap. F1.0).

Practicum

Section 1. Accessing an Image

If you have not already done so, you can add the book's code repository to the Code Editor by entering https://code.earthengine.google.com/?accept_repo=projects/gee-edu/book (or the short URL bit.ly/EEFA-repo) into your browser. The book's scripts will then be available in the script manager panel to view, run, or modify. If you have trouble finding the repo, you can visit bit.ly/EEFA-repo-help for help.

In the bottom left corner of the image (Fig. F1.1.5, area C), rivers and lakes appear very dark, which means that the pixel value in all three bands is low. However, sediment plumes fanning from the river into the sea appear with blue and cyan tints (Fig. F1.1.5, area D). If they look like primary blue, then the pixel value for the second band (B3) is likely higher than the first (B4) and third (B2) bands. If they appear more like cyan, an additive color, it means that the pixel values of the second and third bands are both greater than the first.

In total, the false-color composite provides more contrast than the true-color image for understanding differences across the scene. This suggests that other bands might contain more useful information as well. We saw earlier that our satellite image consisted of 19 bands. Six of these represent different portions of the electromagnetic spectrum, including three beyond the visible spectrum, that can be used to make different false-color composites. Use the code below to explore a composite that shows shortwave infrared, near infrared, and visible green (Fig. F1.1.7).

```
Map.addLayer(
    first_image,
    {
        bands: ['SR_B5', 'SR_B4', 'SR_B2'],
        min: 8000,
        max: 17000
    },
    'Short wave false color');
```

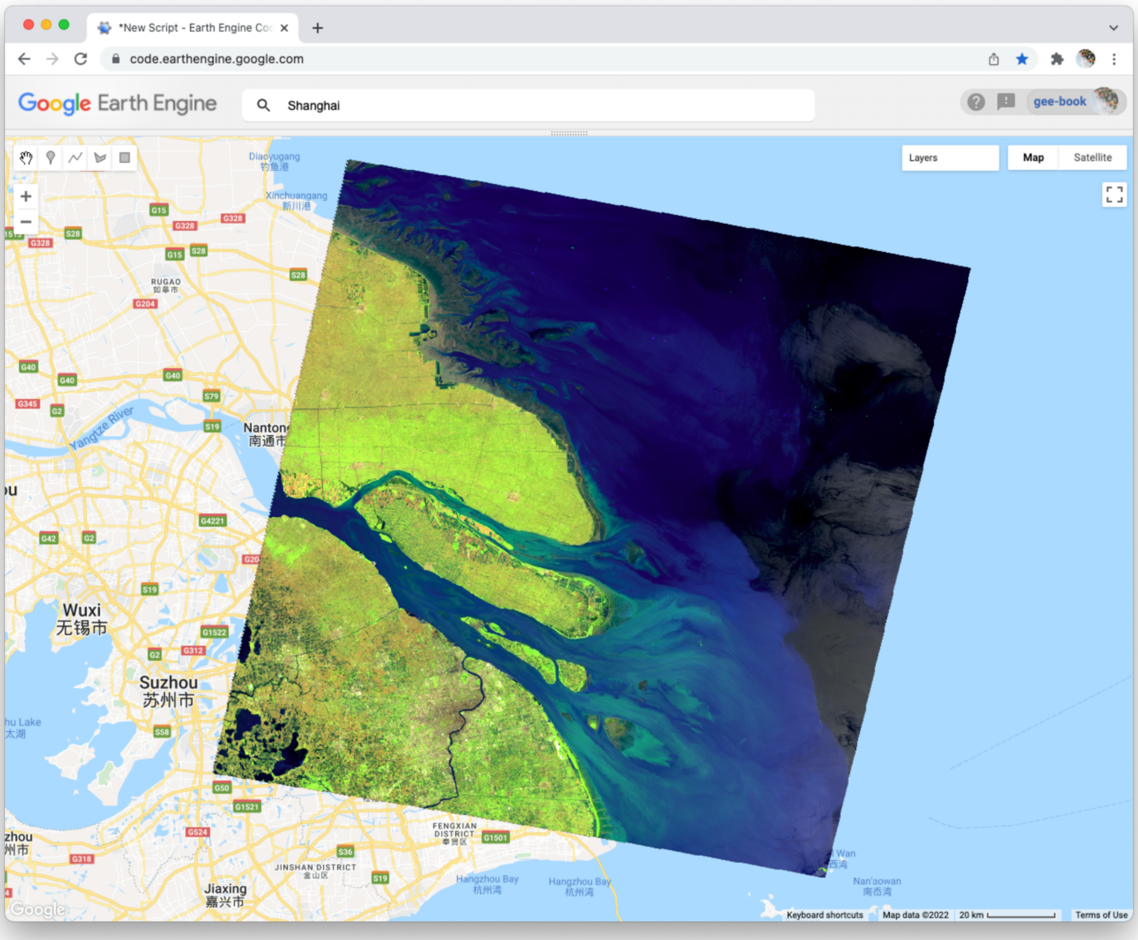

Fig. F1.1.7 Shortwave infrared false-color composite

To compare the two false-color composites, zoom into the area shown in the two pictures of Fig. F1.1.8. You should notice that bright red locations in the left composite appear bright green in the right composite. Why do you think that is? Does the image on the right show new distinctions not seen in the image on the left? If so, what do you think they are?

Fig. F1.1.8 Near-infrared versus shortwave infrared false-color composites

Code Checkpoint F11b. The book's repository contains a script that shows what your code should look like at this point.

Section 5. Additive Color System

Thus far, we have used RGB composites to make a true-color image, in which the colors on the screen match the colors in our everyday world. We also used the same principles to draw two false-color combinations of optical bands collected by the satellite. To be able to read and interpret information from composite images generally, it is useful to understand the *additive color system*. Views of data in Earth Engine, and indeed everything drawn on a computer screen, use three channels for display (red, green, and blue). The order of the bands in a composite layer determines the *color channel* used to display the DN of pixels. When the DN is higher in one band relative to the other two bands, the pixel will appear tinted with the color channel used to display that band. For example, when the first band is higher relative to the other two bands, the pixel will appear reddish. The intensity of the pixel color will express the magnitude of difference between the DN quantities.

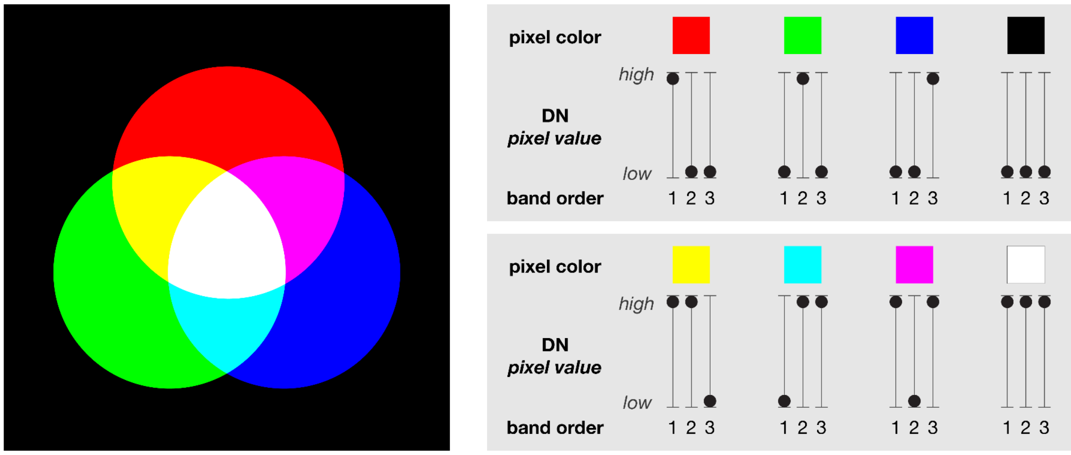

Fig. F1.1.9 Additive color system

The way that primary colors combine to make new colors in an additive color system can be confusing at first, especially if you learned how to mix colors by painting or printing. When using an additive color system, red combined with green makes yellow, green combined with blue makes cyan, and red combined with blue makes magenta (Fig. F1.1.9). Combining all three primary colors makes white. The absence of all primary colors makes black. For RGB composites, this means that if the pixel value of two bands are greater than that of the third band, the pixel color will appear tinted as a combined color. For example, when the pixel value of the first and second bands of a composite are higher than that of the third band, the pixel will appear yellowish.

Section 6. Attributes of Locations

So far, we have explored bands as a method for storing data about slices of the electromagnetic spectrum that can be measured by satellites. Now we will work towards applying the additive color system to bands that store non-optical and more abstract *attributes* of geographic locations.

To begin, add this code to your script and run it.

```
var lights93 = ee.Image('NOAA/DMSP-OLS/NIGHTTIME_LIGHTS/F101993');
print('Nighttime lights', lights93);
```

```
Map.addLayer(
    lights93,
    {
        bands: ['stable_lights'],
        min: 0,
        max: 63
    },
    'Lights');
```

This code loads an image of global nighttime lights and adds a new layer to the map. Please look at the metadata that we printed to the Console panel. You should see that the image consists of four bands. The code selects the “stable_lights” band to display as a layer to the map. The range of values for display (0–63) represent the minimum and maximum pixel values in this image. As mentioned earlier, you can find this range in the Earth Engine Data Datalog or with other Earth Engine methods. These will be described in more detail in the next few chapters.

The global nighttime lights image represents the average brightness of nighttime lights at each pixel for a calendar year. For those of us who have sat by a window in an airplane as it descends to a destination at night, the scene may look vaguely familiar. But the image is very much an abstraction. It provides us a view of the planet that we would never be able to see from an airplane or even from space. Night blankets the entire planet in darkness. There are no clouds. In the “stable lights” band, there are no ephemeral sources of light. Lightning strikes, wildfires, and other transient lights have been removed. It is a layer that aims to answer one question about our planet at one point in time: In 1993, how bright were Earth’s stable, artificial sources of light?

With the zoom controls on the map, you can zoom out to see the bright spot of Shanghai, the large blob of Seoul to the north and east, the darkness of North Korea except for the small dot of Pyongyang, and the dense strips of lights of Japan and the west coast of Taiwan (Fig. F1.1.10).

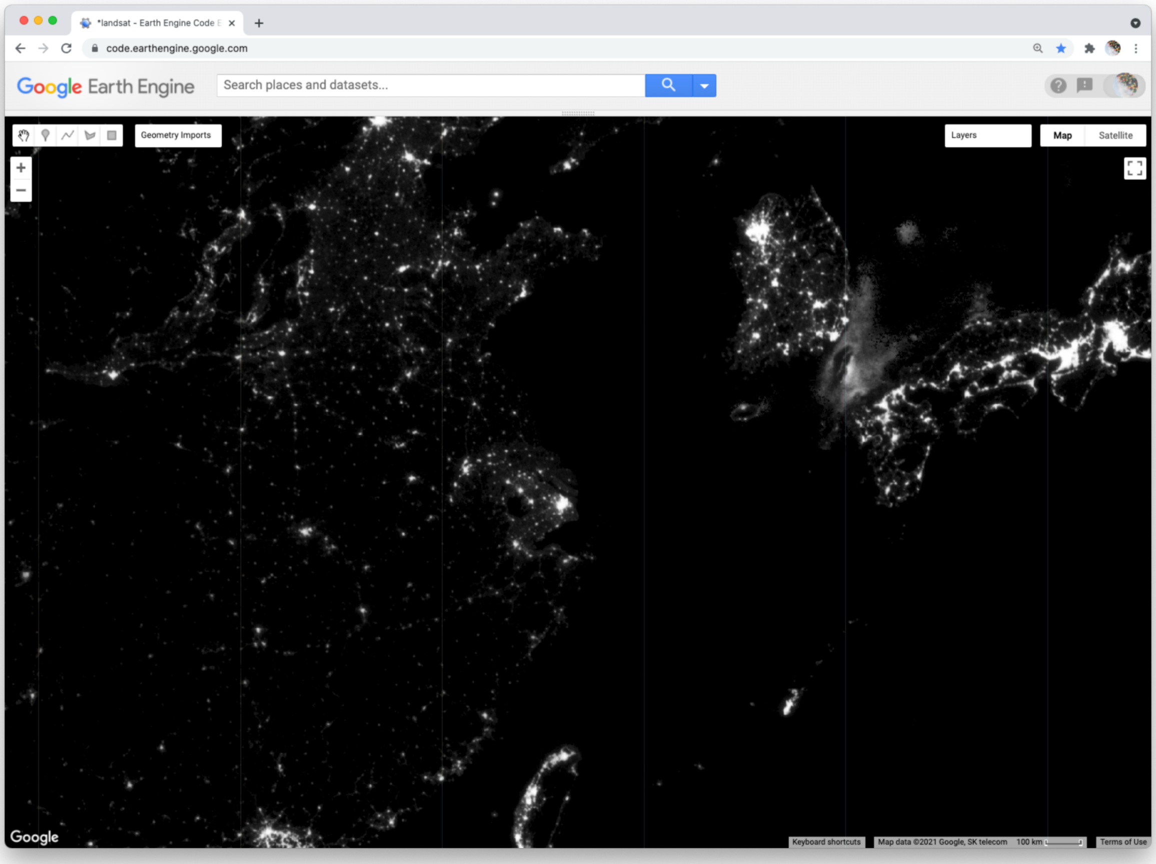

Fig. F1.1.10 Stable nighttime lights in 1993

Section 7. Abstract RGB Composites

Now we can use the additive color system to make an RGB composite that compares stable nighttime lights at three different slices of time. Add the code below to your script and run it.

```
var lights03 = ee.Image('NOAA/DMSP-OLS/NIGHTTIME_LIGHTS/F152003')
    .select('stable_lights').rename('2003');

var lights13 = ee.Image('NOAA/DMSP-OLS/NIGHTTIME_LIGHTS/F182013')
    .select('stable_lights').rename('2013');

var changeImage = lights13.addBands(lights03)
    .addBands(lights93.select('stable_lights').rename('1993'));
```

```
print('change image', changeImage);

Map.addLayer(
    changeImage,
    {
        min: 0,
        max: 63
    },
    'Change composite');
```

This code does a few things. First, it creates two new images, each representing a different slice of time. For both, we use the `select` method to select a band ("stable_lights") and the `rename` method to change the band name to indicate the year it represents.

Next, the code uses the `addBands` method to create a new, three-band image that we name "changeImage". It does this by taking one image (lights13) as the first band, using another image (lights03) as the second band, and the lights93 image seen earlier as the third band. The third band is given the name "1993" as it is placed into the image.

Finally, the code prints metadata to the **Console** and adds the layer to the map as an RGB composite using `Map.addLayer`. If you look at the printed metadata, you should see under the label "change image" that our image is composed of three bands, with each band named after a year. You should also notice the order of the bands in the image: 2013, 2003, 1993. This order determines the color channels used to represent each slice of time in the composite: 2013 as red, 2003 as green, and 1993 as blue (Fig. F1.1.11).

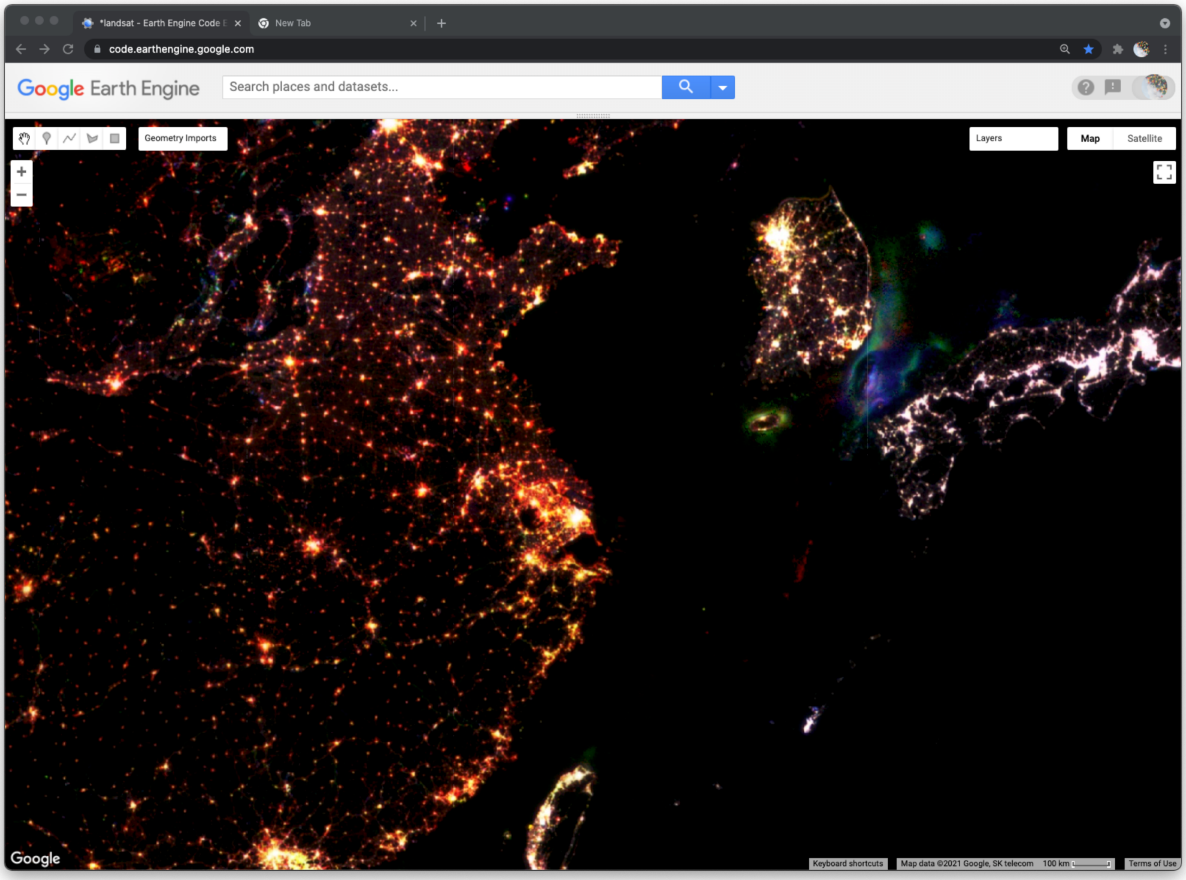

Fig. F1.1.11 RGB composite of stable nighttime lights (2013, 2003, 1993)

We can now read the colors displayed on the layer to interpret different kinds of changes in nighttime lights across the planet over two decades. Pixels that appear white have high brightness in all three years. You can use the **Inspector** panel to confirm this. Click on the **Inspector** panel to change the cursor to a crosshair and then click on a pixel that appears white. Look under the *Pixel* category of the **Inspector** panel for the “Change composite” layer. The pixel value for each band should be high (at or near 63).

Many clumps of white pixels represent urban cores. If you zoom into Shanghai, you will notice that the periphery of the white-colored core appears yellowish and the terminal edges appear reddish. Yellow represents locations that were bright in 2013 and 2003 but dark in 1993. Red represents locations that appear bright in 2013 but dark in 2003 and 1993. If you zoom out, you will see this gradient of white core to yellow periphery to red edge occurs around many cities across the planet, and shows the global pattern of urban sprawl over the 20-year period.

When you zoom out from Shanghai, you will likely notice that each map layer redraws every time you change the zoom level. In order to explore the change composite layer more efficiently, use the **Layer** manager panel to not show (uncheck) all of the layers except for “Change composite.” Now the map will respond faster when you zoom and pan because it will only refresh the single displayed shown layer.

In addition to urban change, the layer also shows changes in resource extraction activities that produce bright lights. Often, these activities produce lights that are stable over the span of a year (and therefore included in the “stable lights” band), but are not sustained over the span of a decade or more. For example, in the Korea Strait (between South Korea and Japan), you can see geographic shifts of fishing fleets that use bright halogen lights to attract squid and other sea creatures towards the water surface and into their nets. Bluish pixels were likely fished more heavily in 1993 and became used less frequently by 2003, while greenish pixels were likely fished more heavily in 2003 and less frequently by 2013 (Fig. F1.1.11).

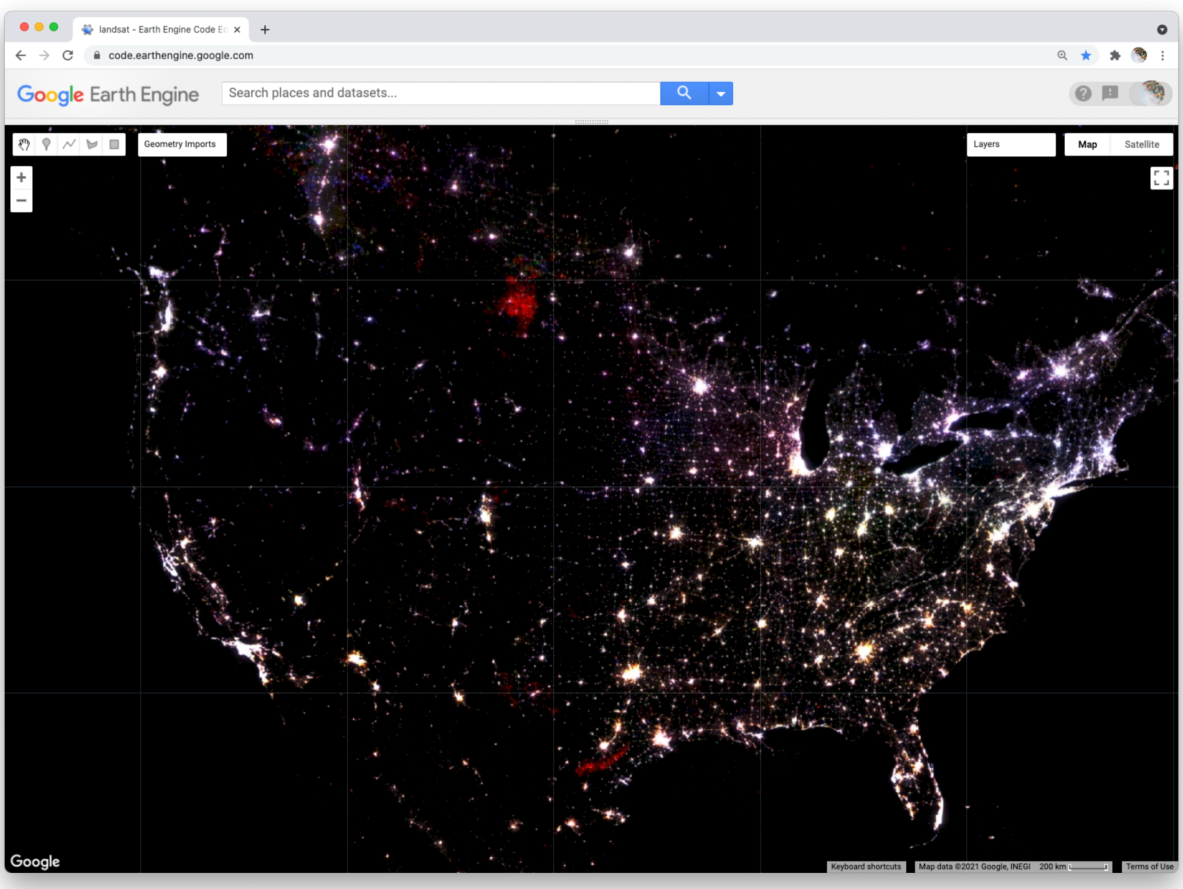

Fig. F1.1.12 Large red blobs in North Dakota and Texas from fossil fuel extraction in specific years

Similarly, fossil fuel extraction produces nighttime lights through gas flaring. If you pan to North America (Fig. F1.1.12), red blobs in Alberta and North Dakota and a red swath in southeastern Texas all represent places where oil and gas extraction were absent in 1993 and 2003 but booming by 2013. Pan over to the Persian Gulf and you will see changes that look like holiday lights with dots of white, red, green, and blue appearing near each other; these distinguish stable and shifting locations of oil production. Blue lights in Syria near the border with Iraq signify the abandonment of oil fields after 1993 (Fig. F1.1.13). Pan further north and you will see another “holiday lights” display from oil and gas extraction around Surgut, Russia. In many of these places, you can check for oil and gas infrastructure by zooming in to a colored spot, making the lights layer not visible, and selecting the **Satellite** base layer (upper right).

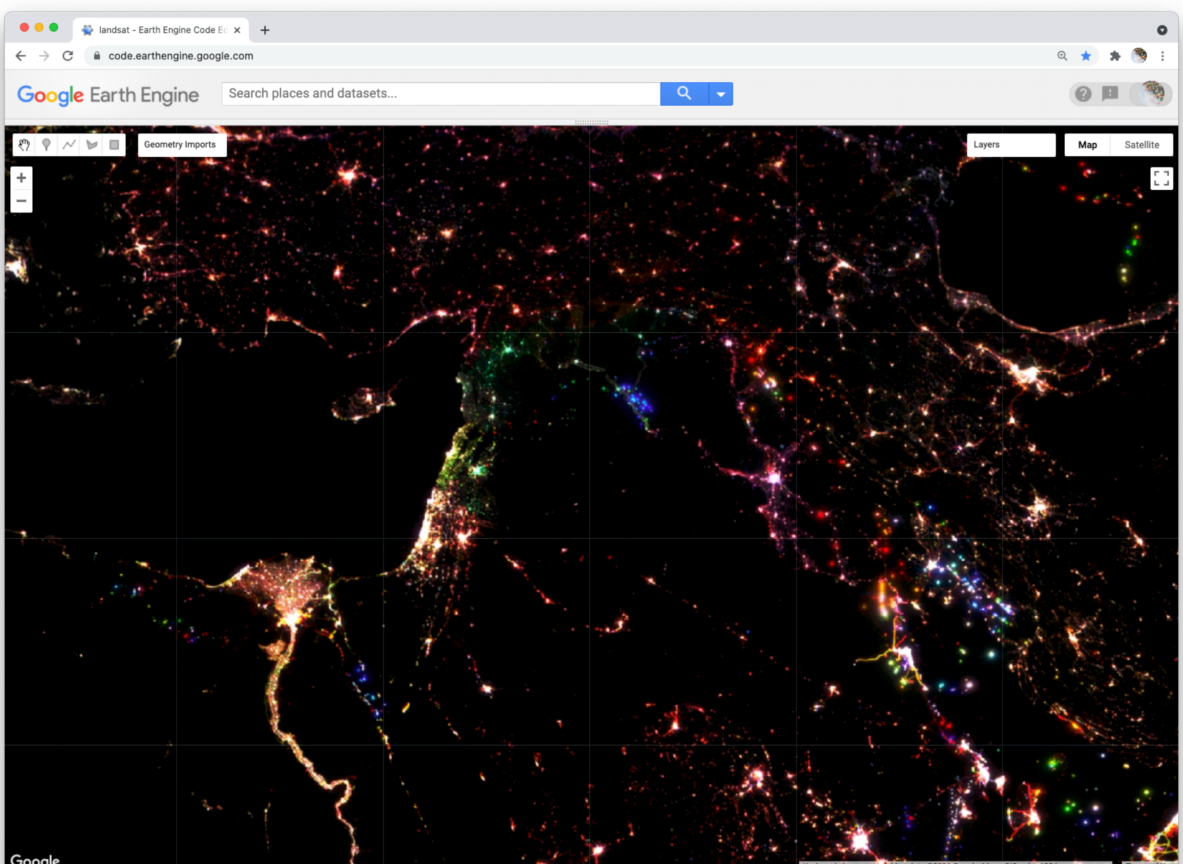

Fig. F1.1.13 Nighttime light changes in the Middle East

As you explore this image, remember to check your interpretations with the **Inspector** panel by clicking on a pixel and reading the pixel value for each band. Refer back to the additive color figure to remember how the color system works. If you practice this, you should be able to read any RGB composite by knowing how colors relate to the relative pixel value of each band. This will empower you to employ false-color composites as a flexible and powerful method to explore and interpret geographic patterns and changes on Earth's surface.

Code Checkpoint F11c. The book's repository contains a script that shows what your code should look like at this point.

Synthesis

Assignment 1. Compare and contrast the changes in nighttime lights around Damascus, Syria versus Amman, Jordan. How are the colors for the two cities similar and different? How do you interpret the differences?

Assignment 2. Look at the changes in nighttime lights in the region of Port Harcourt, Nigeria. What kinds of changes do you think these colors signify? What clues in the satellite basemap can you see to confirm your interpretation?

Assignment 3. In the nighttime lights change composite, we did not specify the three bands to use for our RGB composite. How do you think Earth Engine chose the three bands to display? How do you think Earth Engine determined which band should be shown with the red, green, and blue channels?

Assignment 4. Create a new script to make three composites (natural color, near infrared false color, and shortwave infrared false-color composites) for this image:

```
'LANDSAT/LT05/C02/T1_L2/LT05_022039_20050907'
```

What environmental event do you think the images show? Compare and contrast the natural and false-color composites. What do the false-color composites help you see that is more difficult to decipher in the natural color composite?

Assignment 5. Create a new script and run this code to view this image over Shanghai:

```
var image = ee.Image('LANDSAT/LT05/C02/T1_L2/LT05_118038_20000606');

Map.addLayer(
    image,
    {
        bands: ['SR_B1'],
        min: 8000,
        max: 17000
    },
    'Layer 1'
);

Map.addLayer(
    image.select('SR_B1'),
    {
        min: 8000,
        max: 17000
    },
    'Layer 2'
);
```

Inspect Layer 1 and Layer 2 with the **Inspector** panel**.** Describe how the two layers differ and explain why they differ.

Conclusion

In this chapter, we looked at how an image is composed of one or more bands, where each band stores data about geographic locations as pixel values. We explored different ways of visualizing these pixel values as map layers, including a grayscale display of single bands and RGB composites of three bands. We created natural and false-color composites that use additive color to display information in visible and non-visible portions of the spectrum. We examined additive color as a general system for visualizing pixel values across multiple bands. We then explored how bands and RGB composites can be used to represent more abstract phenomena, including different kinds of change over time.

Feedback

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