Lab 1 – Introduction to Multispectral Imagery

Fall 2025
Geography 413: Remote Sensing of the Environment
Instructor: Dr. Hannah Herrero
Lab Instructor: Dryver Finch

Authored by Dryver Finch August 2025

Outline of Objectives

1.1 GeoTIFFs and QGIS

A. Visualize local multispectral imagery in QGIS.
B. Manipulate images within QGIS (merging, visualization parameters)
C. Understand bands and metadata.

1.2 Google Earth Engine and Colab

A. Search for datasets within GEE and understand data types.
B. Visualize Cloud-Optimized GeoTIFFs (COGs) within the Code Editor.
C. Create Python Notebook (.ipynb) files and run script in Google Colab.

1.1: GeoTIFFs and QGIS

If you have used any GIS software up to this point, it's likely been an Esri product (ArcGIS Pro, Online, Story Maps, etc.). While this is the industry standard, Quantum GIS (QGIS) is an excellent open-source option that is more flexible with multispectral imagery visualization and analysis. Also, since it's open-source, it's free to use and learning this software allows you to do GIS work without having to worry about a pay wall when your UTK access expires after graduation!

While most of your lab work and final projects will be done using Google Earth Engine, QGIS is a great way to get introduced to the data you will be working with across platforms. Also, depending on the project and data needs, it could make more sense to use QGIS than Earth Engine.

A. Visualizing Multispectral Images

Most multispectral images you encounter will be GeoTIFF (.tif) files. These are very similar to the JPEG/PNG format you may be familiar with in traditional photography. Images are broken down into pixels, with each pixel representing a different color hue. Together, these pixels mosaic an image we can recognize. The smaller the pixels are, the more detailed our image is. For example, if you zoom in all the way on your phone's camera, you may notice that the image quality deteriorates, and pixels become more noticeable. GeoTIFFs also store data in pixels, but the basic difference is that these images are georeferenced to a specific place on Earth. So, when we load the image, it automatically covers the correct area for us.

Let's try it out.

1) First, launch QGIS on your device and create a New Project (Ctrl+N).

2) Add a basemap by navigating to 'XYZ Tiles' in the Browser tab on the left-hand side and double clicking "OpenStreetMap".

3) In your Lab1Data folder, you should see four different "1_1_A_bandX.tif" files. Load those into QGIS by dragging and dropping from File Explorer.

4) Save your project somewhere safe! (OneDrive is preferred – if you save on a desktop computer in the lab, you will likely lose the entire project and have to start over.)

ANSWER QUESTION 1

These images were taken from the Landsat-9 satellite in May of 2025. The sensor aboard the satellite is able to record multiple spectral signatures (where we get the term 'Multispectral') from the Earth's Surface Reflectance in a given area. While these four images are from the exact same place and time, they each hold much different information. Refer to the Landsat9_FactSheet in your Lab 1 folder for more information on bands. For the purposes of this lab, we will be looking at the four most used bands: Red, Green, Blue, and Near-Infrared. You will discuss the science behind this more in the lecture. For now, all you need to know about the bands is provided in the Fact Sheet.

Images are recorded in grayscale; black being the lowest reflectance values and white being the highest reflectance values. For example, let's say you took a multispectral image of a highway that had been plowed following a blizzard. The black asphalt road will not reflect much light, but the white snow beside it reflects a lot of light. When we look at our grayscale images, it may be hard to tell it's not colorized because the black road would be black (low reflectance values) and the snow would be white (high reflectance values).

ANSWER QUESTION 2

B. Manipulating Images

While we can get a lot of good information from each band individually, it's more helpful and intuitive to have all bands in a single image. When we have an image with Red, Green, and Blue (RGB) bands, we can view a True-Color image.

Let's practice:

1) Within your QGIS project, look to the top ribbon and navigate to Raster > Miscellaneous > Merge.

2) For the Input Layers parameter, select your four images.

3) Check the box beside "Place each input file into a separate band"

4) Instead of [Save to Temporary File] select Save File and save the merged image to your project folder. Make sure you save your files as a .tif

5) Click Run.

6) Once your image loads, right click the layer and select "Properties" at the bottom, then choose "Symbology"

7) The Render Type should be Multiband color – if it's Singleband Gray, you did not merge properly.

8) Once you merge, your band order should be Blue (1), Green (2), Red (3), NIR (4) – to view a True-Color image, readjust your band combination to be a 3-2-1 visualization.

9) Click Apply.

ANSWER QUESTION 3

C. Bands and Metadata

It's easier for us to look at a True-Color composite because that's what our eyes see: only the visible light (RGB). Sometimes, though, we need to look at different arrangements in our research that can highlight things our eyes cannot see. One of the most common arrangements in Remote Sensing is NIR-Green-Red. Each image will go into a separate "color gun": red, green, and blue, respectively. This shows pixels with high NIR reflectance as red, Green pixels remain green, and (oddly enough) Red pixels show up as blue. Refer to your Fact Sheet to remind yourself what you're looking at – it can get confusing!

Let's view our image in a False-Color composite:

1) Go back to the Symbology tab in Properties.

2) Place your NIR band in the red color gun, your Green band in the green color gun, and Red band in the blue color gun. (4-2-3 visualization)

3) Click Apply.

ANSWER QUESTION 4

ANSWER CHALLENGE QUESTION

1.2: Google Earth Engine and Colab

Google Earth Engine (GEE) hosts petabytes of spatial data that we can access once we have a Cloud Account registered – which you all should have by now. The two main products/interfaces that Google offers within GEE are the Data Catalog and Code Editor. Google also offers a cloud-based coding platform called Colab that we will be using throughout the semester. Keeping everything on Google makes the most sense for us in this case because that's where our data is hosted, but there are many ways to work with remotely sensed data (like QGIS!) depending on the scenario.

A. Data Catalog

Navigate to the Data Catalog if you have not already. Explore the catalog and investigate any datasets that interest you! The home page has different imager highlighted or you can go to the Categories tab to find something more specific. There are two main data types you will encounter in GEE: *Features* and *Images*.

Features = Vector (points, lines, polygons)
Images = Raster (temperature, precipitation, multispectral)

When you have a set of Feature or Images, you have a *FeatureCollection* or *ImageCollection*, respectively. You cannot have a collection that contains both Features and Images. Knowing what data type you are working with will help you later when we start running functions on our data – you can't visualize a polygon in Mulitband Color.

The table below gives a breakdown of datatypes for two example datasets:

Dataset	Single Object	Object Set
TIGER: US Census States	Object Type: Feature - Each state in this dataset is an individual Feature	Set Type: FeatureCollection - All the states and territories together in one FeatureCollection
MOD44B.061 Terra Vegetation Continuous Fields Yearly Global	Object Type: Image - Each year in this dataset is an individual Image	Set Type: ImageCollection - Every year from 2000 to 2024 together in one ImageCollection

ANSWER QUESTION 5

Let's look a little closer at the TIGER dataset:

1) It's always a good idea to look at your metadata before you do anything. Read the Description and then go to Table Schema.

2) Table Schema is where you can find the Properties for a FeatureCollection. Tables can be set up differently depending on the data provider. TIGER US Census data is pretty much always going to be formatted the same way.

3) The 'NAME' column contains the name of each property. When we are working with data in Python, we will call on these names to filter our data.

4) 'TYPE' tells us the data type (string, double, int, etc.) and 'DESCRIPTION' tells us a bit more about that field.

ANSWER QUESTION 6

Next, let's check out the MODIS dataset:

1) First thing to do when you are looking at ImageCollections is to check 'Dataset Availability' and 'Cadence'. These tell us more about the temporal (time) elements of the data.
    a. Dataset Availability: 2000-03-05 – 2024-03-05 = First image in 2000 and latest image in 2024
    b. Cadence: 1 year – New image added every year = Yearly Imagery

2) Read the Description. You may notice there is no 'Table Schema'. This is because images do not have Properties, they have Bands.

3) In the Bands tab, the first thing you will see is 'Pixel Size' or our image's resolution. MODIS imagery has 250-meter pixels, which is quite large/coarse (the larger the pixel, the lower the image resolution).

4) The bands for MODIS are different than the Landsat image we worked with in QGIS. That's because MODIS is a product derived from multispectral imagery – it's still raster data but it is not multispectral (or spectral at all).

5) There are still 'NAME' and 'DESCRIPTION' columns that work the same as they do with FeatureCollections. The main difference you will notice are the UNITS, MIN, and MAX.
    a. Units: In this instance, our units are % for each band, meaning each pixel will have a single percentage value for the 250m x 250m are covered by the given pixel.
    b. Min/Max: The bounds for the band in the given unit. In our example, each band ranges from 0% to 100%.

ANSWER QUESTION 7

B. Code Editor

The GEE Code Editor is a Java-powered interface that has become a pillar of remote sensing cloud computing. While we won't be using the Code Editor much directly, our next two labs use packages built off the Code Editor.

Let’s go ahead and visualize the vector data we’ve been working with:

1) Navigate back to the TIGER data.

2) Scroll down and click “Open with Code Editor”.

3) The script should be copied into the terminal. Click “Run”.

4) In the line that starts with **var image** change ‘ALAND’ to ‘AWATER’ and Run.
    a. This changes what Property we are visualizing. Instead of categorizing states by Land Area, we are categorizing by Water Area. Since we did not change our min/max, all of our values are lower, making them only purple and blue (our bottom two classes).

5) In the **var visParams** line, set min: 5e+8 and max: 5e+9 and Run.

ANSWER QUESTION 8

Now, let’s look at the raster data:

1) Navigate back to the MODIS data.

2) Click “Open with Code Editor”.

3) The script should be copied into the terminal. Click “Run”.

4) In **var visualization** change [‘Percent_Tree_Cover’] to [‘Percent_NonTree_Vegetation’] and Run.

C. Google Colab

Google Colab is your new best friend this semester – it makes cloud computing accessible for people at all levels of programming ability. This week, we are just going to get comfortable with the interface and notebook sharing. No coding… yet!

1) Open this Colab Notebook that I made.

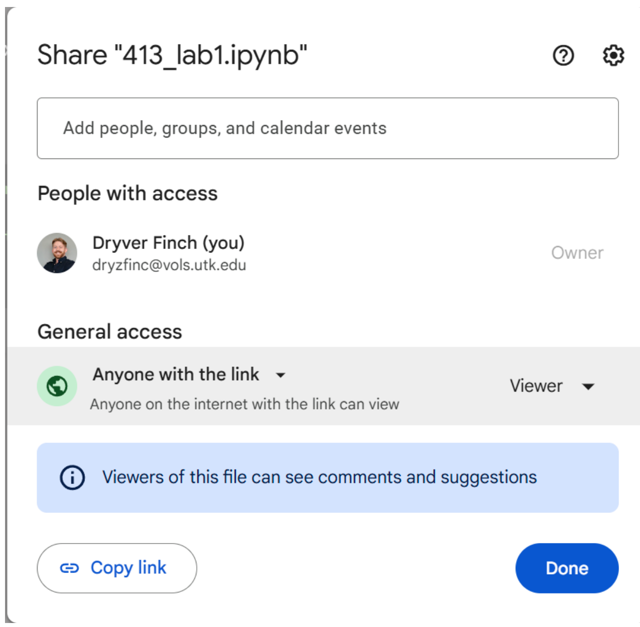
2) Upon opening, you are viewing my version of the notebook. You are not given permission to edit, so you will need to save a copy to your Drive.

3) Navigate to File > Save a Copy in Drive

4) Rename your file from Copy of 413_lab1.ipynb to LASTNAME_413lab1.ipynb

5) Add your name to the Text cell and create a new Code cell that prints a supportive message.

6) Click Share in the top right. Make sure your settings match those in the image below and Copy link.
    a. If your settings are not the same, I will not be able to open your link and will not be able to grade your lab!



ANSWER CHALLENGE QUESTION 2