EGM 722: “Programming for GIS and Remote Sensing“
Teresa Bornschlegl
Student number: B00863107
24.05.2022

1. Introduction

This code analyses changes in artisanal small-scale (ASM) mining activity at an illegal mining site in Bisie in the DR Congo (DRC) between the years 2016 and 2022 (Kranz et al. 2017). While ASM provides a livelihood to more than 45m peoples globally, it also is associated with financing military conflict groups and environmental degradation, such as deforestation and mercury pollution (Nayameyke et al.2021).

This code calculates the NDVI for Sentinel 2A imagery as well as the change of the NDVI between the years 2016 and 2022. Prior to the NDVI calculation, the code first loads and reads the imagery into memory, inspects several key characteristics of the imagery, and displays the False Colour Composite of the imagery for both years.

2. Setup and installation

a) Download my repository at my Github. You can find my repository at the following link: https://github.com/tbornschlegl/egm722_assignment.git

The repository includes:

- The Jupyter script for the code “Assignment_Script”
- The data files “sub_16.tif”, “sub_22-tif”)
- The Environment as environment.yml
- Readme
- License
- Gitignore
- Assignment Part 1.pdf

This code is written as a Jupyter notebook. To run this code, you best install Anaconda and the Jupyter notebook.

b) Download from my repository:

- The Jupyter script for the code “Assignment_Script”
- The data files “sub_16.tif”, “sub_22-tif”)
- The Environment as environment.yml

c) Install Anaconda – if you do not have installed it yet. To install follow the instructions provided at the following link: https://docs.anaconda.com/anaconda/install/

d) Once you have Anaconda installed, you need to set up the appropriate Environment to run this code. To do this, you have to import the Environment that you downloaded from my repository. To do so, click on import, then navigate under “Local Drive” to the location where you have saved the environment from my repository. This will create an environment with python version 3.9.

The packages you need to run this code are the following:

- Python (version 3.9)
- Cartopy (any version)

- Jupyter notebook (any version)
- Rasterio (any version)
- NumPy (any version)
- Matplotlib (any version)

e) Open Jupyter Notebook, upload and open the script “Assignment_Script”

3. Methods and data

This code analyses changes in artisanal small-scale (ASM) mining activity at an illegal mining site in Bisie in the DR Congo (DRC) between the years 2016 and 2022. The Bisie mining site is located in North Kivu, at 1°2’ 0´´S, 27°44’ 0´´ E. The main mineral is cassiterite which is used to produce tin (Kranz et al. 2017).

The data used are two Sentinel 2A multispectral imageries with band 2 (blue), 3(green), 4(red) and 8(NIR) with a 10m spatial resolution from Tile T35MNU, acquired the 05.04.2022 and 06.04.2022, respectively. Both images were cropped to a subset around the mining site prior to their use in the code.

Parts of the code were adapted from Dr. McNabb 2022 EGM 722 “Programming for GIS and Remote Sensing“ and from the Tutorials at: https://geobgu.xyz/py/rasterio.html#calculating-indices.

First, the code opens the datasets and load the imageries from 2016 and 2022 into the memory. We load all bands of the imagery, that is, band1(blue), band2(green), band3(red) and band8(NIR). If you were to load only one particular band, you would need to specify the number of the band in the read() method. For instance, for loading only band8(NIR), you would need to modify the code as, e.g. img16 = dataset16.read(3). Be careful: in the read() method, indices start from 1, not from 0.

The loaded images consist of arrays with the pixel values of the particular bands. Similar to lists and tuples, these arrays can be indexed. If there are more than just one band, the first index corresponds to the band, the second to the row (y) location, and the third to the column (x). If there is only one band, the first would correspond to the row (y) location, the second to the column (x) location.

The code then inspects several key characteristics of the imagery. First, it inspects the metadata of the imagery, such as, the datatype, the number of rows and columns, the geographic reference system, and the number of bands. The code also verifies the Bounding Box of the imagery, that is, the corners of the image at the left, top, bottom, and right.

The code also verifies the shape of the loaded imagery and verify if the shapes are the same for both bands. This is important because if the shapes were not the same, we could not perform calculations between the arrays. Finally, the code checks the maximum and minimum pixel value of all bands for both images.

Subsequently, the code inspects both images visually by displaying the images as False Colour Composite (NIR, red, green) and saves the created figure. The function does two important things: First, it re-orders the indices by applying the tranpose() method. This is important because the read()methods loads the raster in a way in which: bands = index 1, rows = index 2, columns = index 3. However, imshow () needs the indices to be order as: rows = index 1, columns = index 2, bands = index 3. Secondly, the function re-scales the image to have values between 0-1 (or 0-255) by dividing the image´s pixel values / 10000 and transforming the dtype into float.

This is needed because supported array values for more than three bands must be either between 0-1 float or 0-255int.

As a next step, the code performs a percentile stretch. While the displaying of the images we just performed serves a first impression, both images turn out to be quite dark. To do so, the code uses a function that calculates a percentile stretch for the imagery. That function assures that the minimum and maximum of the pixel values are between 0, 100 and that the image is only 2-dimensional. Secondly, the function makes sure that all values calculated in the stretch below the minimum are set to 0 and the ones above the maximum value to 1.

After having inspected both images visually, the code performs the NDVI calculation. As a first step, we re-scale both images to make sure the pixel values are between 0 and 1 as well as floats (fractions). Then, the code calculates the NDVI for both years, applying the formula NDVI = (NIR) -(Red) / (NIR) + (Red) for our code. Subsequently, the code displays the results from the NDVI calculation of both years.

Finally, the code calculates and displays the difference in the NDVI between year 2016 and 2022.

4. (Expected) results of running my code

The code will first show the information pertaining to attribute and metadata information.
Results from displaying the percentile stretched imageries show that artisanal small-scale mining intensified over the years. In the 2022 imagery, we can see that "centre" of the mining site persisted over the years. New mining sites opened up on the north of the mining site while the road to the mining site to the south as well as some "patches" regrow some vegetation (see Figure 1 and 2).

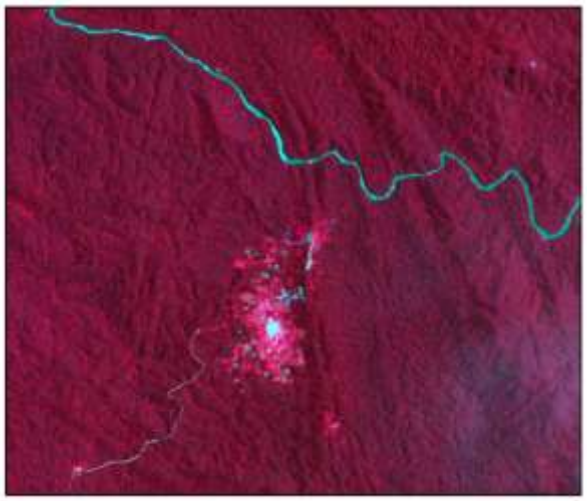

Figure 1: False Colour Composite 2016

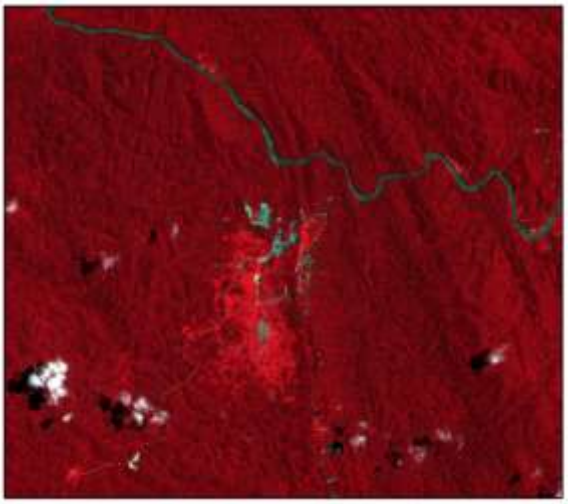

Figure 2: False Colour Composite Satellite 2022

Results from calculating the NDVI for year 2016 and 2022 show the above mentioned results even more clearly. The red patches in year 2022 show the expansion of the mining site to the north and to the south-east as well as some re-vegetation of the road in the south (see Figure 3 & 4). The river as well as clouds are also shown in red, especially in the year 2022 – which would need to be masked in further developing the code.

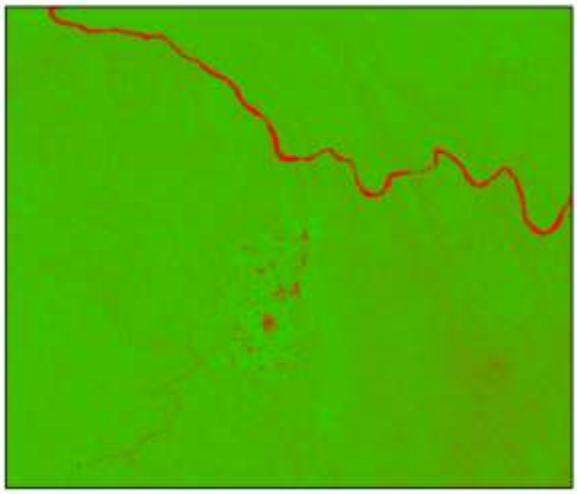

Figure 3: NDVI 2016

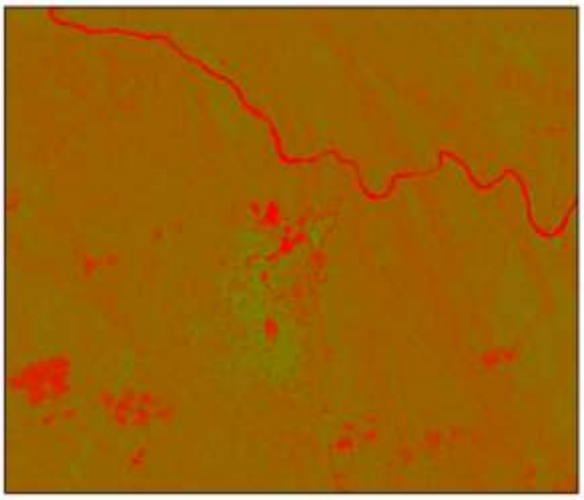

Figure 4: NDVI 2022

Results from calculating and displaying the NDVI difference of year 2016 and 2022 shows the extension of the mining site´s expansion re-vegation even more clearly. The extension of mining site in the north (as well as the clouds) is shown in red (see Figure 5). The sites of re-vegetations are shown in light turquoise. I must further investigate why the river also shows a difference between both images. Further developing the code, I would need to include a title and legend in the displayed imagery.

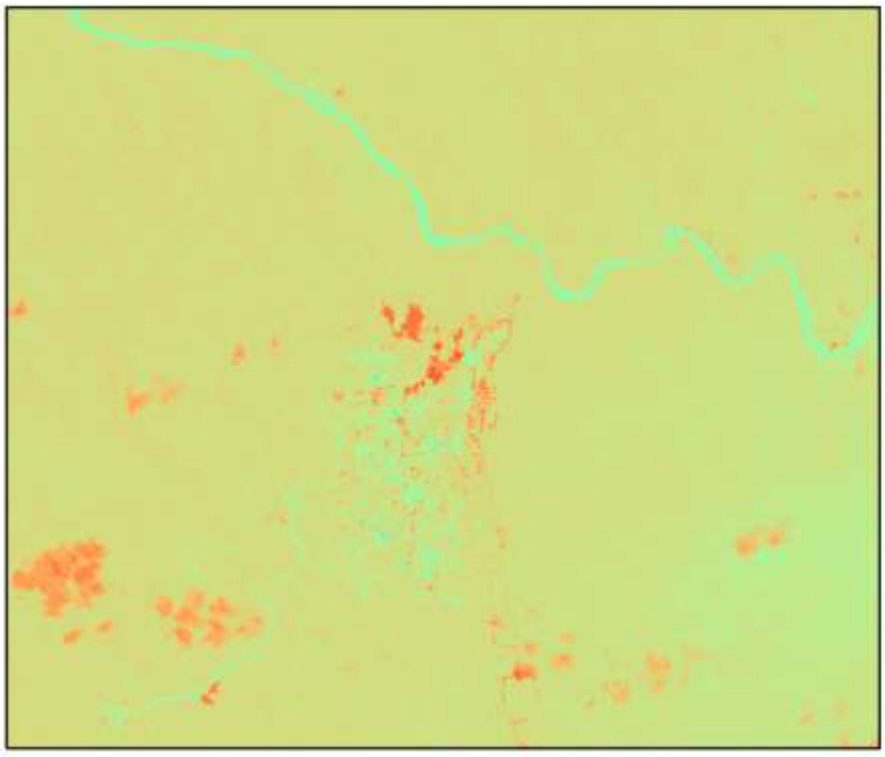

Figure 5: NDVI Difference

5. Troubleshooting advice

If the code throws a NameError, indicating that a certain name or variable is not defined, you must likely run again the cells above, specifically the cell in which the name/variable is defined.

If the code throws as RasterioIOError, indicating that the file or directory cannot be found, likely the path to your file is not correct. In that case, copy the path again of the location in which you have stored your files (downloaded from my Github) and insert that path again the the (“ “).

6. References

Kranz, O., Lang, S., & Schoepfer, E. (2017). Int J Appl Earth Obs Geoinformation 2 . 5D change detection from satellite imagery to monitor small-scale mining activities in the Democratic Republic of the Congo. Int J Appl Earth Obs Geoinformation, 61, 81–91.

Nyamekye, C., Ghansah, B., Agyapong, E., Obuobie, E., Awuah, A., & Kwofie, S. (2021). Remote Sensing Applications : Society and Environment Examining the performances of true color RGB bands from Landsat-8 , Sentinel-2 and UAV as stand-alone data for mapping artisanal and Small-Scale Mining (ASM). Remote Sensing Applications: Society and Environment, 24