Life Sciences and Facility Management
Institute of Natural Resource Sciences

Julian Kraft UI22
Remote Sensing
April 8, 2024

NDVI calculation using Sentinel-2 data and Python

Introduction

The Normalized Difference Vegetation Index (NDVI) is a widely used index to monitor vegetation health and growth. It is calculated from the red and near-infrared bands of satellite imagery. The NDVI values range from -1 to 1, where values close to 1 indicate healthy vegetation and values close to -1 indicate no vegetation. In this report, the calculation using Sentinel-2 satellite data and Python is described. The data processing was done for the area of the Kanton of Schaffhausen in Switzerland. The area is defined by the bounding box (West: 8.3, South: 47.53, East: 8.9, North: 47.84, ESPG:4326). The temporal extent covers the summer months (June, July, August) of the years 2017 to 2023. To solve this task, three different Remote Sensing products are used: Sentinel-2 Level-2A data bands B04 (Red) and B08 (NIR) and the preprocessed Scene Classification Layer (SCL). In this report the focus is on how the data was obtained and processed and how the created NDVI product looks like for a single time frame.

For the next step, namely task 2, the idea is to analyze the data over time. The NDVI for itself provides limited information about the actual vegetation health and growth, since the values are relative. Therefore, the plan is to calculate the deviation from the mean NDVI over the whole time analyzed for each pixel on every time frame. This needs to be further elaborated and tested.

Methods

To process and download the data, the OpenEO platform was used. There is a great advantage in using platforms like this, since they provide a lot of functionalities to process date in the cloud before downloading only the needed result. The platform provides a Python API, which was used to access the data and process it. It is important to split the processing into smaller steps, since the data is too large to be processed at once. In this case it was done month by month. The code is available on GitHub. Only frames with less than 85% cloud cover were used.

The NDVI was calculated using the following formula (Sinergise 2024):

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{1}$$

After calculating the NDVI, the values were masked using the SCL layer. All pixels not classified as vegetation were set to not available (nan) in order for them not to be considered in the analysis. This solves the problem of having NDVI values for clouds as well as NDVI values for water bodies, urban areas, etc. The final NDVI product was then downloaded as Data Cube with the dimensions time, x, y and the coordinate system ESPG:32632. The data cube was saved as a NetCDF file.

Results

In Figure 1 the NDVI for the Kanton of Schaffhausen for a single time frame is shown. There are some plain white areas where no values are available. The values for the NDVI range from 0.2 to 0.9. This represents only a sample of the generated data. For every Image of Sentinel-2 with a cloud coverage of less than 85% during the defined spatial and temporal extent - data is available and ready for further analysis.

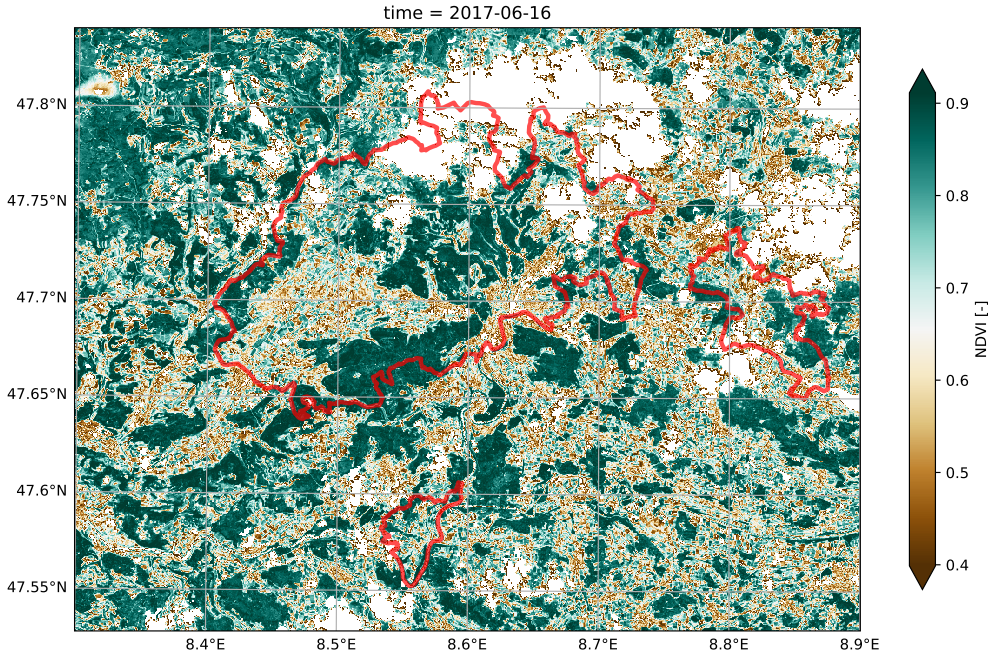

Figure 1: Plot of the NDVI for the Kanton of Schaffhausen for a single time frame. The red line represents the boarder of the Kanton.

Discussion

The dark green areas in Figure 1 represent especially dense and healthy vegetation. The area more or less corresponds to the forested area. The values range only from 0.2 to 0.9, wich is most likely due to the fact that only vegetation pixels are considered and the conditions for plant growth must have been good. Anyhow it is difficult to explain with no further investigation. This is something that will be analyzed in more detail before taking the task further.

References

Sinergise, Sentinel-Hub by (2024). *Normalized difference vegetation index*. Sentinel Hub custom scripts. URL: https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/ndvi/ (visited on 04/07/2024).

Declaration of AI Assistance

ChatGPT was used to answer a broad variety of questions concerning Python and LaTeX code
GitHub copilot was used to assist while writing the report and coding

Code

All code is available on https://github.com/juliankraft/RemoteSensing_Task01