An interactive GEE application for information visualization and extraction tool using radar remote sensing data

Niklas JAGGY

Department of Geoinformatics – Z_GIS, University of Salzburg, Austria,
niklas.jaggyg@stud.sbg.ac.at

Abstract

The last 15 years have seen an increase of applications where geospatial technologies and Earth Observation data are used to support operations of humanitarian NGOs. The main benefit thereby is the possibility to obtain reliable information of crisis areas that are usually not or difficult to access. Currently, most information is extracted from optical satellite imagery although radar data can complement the information retrieval due to its inherent advantages concerning data acquisition in cloudy areas and during nighttime. Although more difficult to interpret, it has shown its benefits in application cases such as dwelling and flood detection. Because radar imagery is not as intuitively interpretable and more complex to process, an extra barrier is present that prevents non-experts from exploiting radar data for information extraction. Therefore, this project aims at creating an interactive Google Earth Engine App that guides the user through a radar imagery visualization and information extraction process to lower the entry barrier for non-experts to explore radar data in humanitarian contexts.

1 Introduction

The last 10 to 15 years have seen a tremendous increase in geospatial technologies and Earth Observation (EO) being taken up both by disaster responders and the humanitarian aid community (Lang et al., 2020; Voigt et al., 2016). This is driven on one side from the need for reliable and updated information on crisis areas which is difficult to obtain on the ground as these areas are usually difficult to access, if at all. Additionally, the urgency of the matter adds a time constraint. The critical information is vital for both the operation planning in the immediate response phase and the long-term monitoring and safeguarding (Lang et al., 2017). On the other side, the availability of EO data has dramatically increased while at the same time, access to large satellite data catalogs has become easier. Amongst the largest benefits of including EO data in an operational context is the objective, large-scale, consistent, continuous, and cost-effective data collection over crisis areas. However, in many cases, issues with cloud cover or other atmospheric obstructions on optical satellite imagery are reported that severely reduce data availability (Braun & Hochschild, 2017). Radar imagery therefore emerges as a complementary data source with its natural benefits in terms of data acquisition during nighttime and penetration of atmospheric obstructions. Although more complex to process and interpret due to radars inherent characteristics it provides a reliable source of EO data (Braun & Hochschild, 2017). Because the raw data does not contain any specific value for humanitarian aid operations, it needs thorough processing and analysis to extract information and derive products that are understandable by the user (Voigt et al., 2016). Additionally, educating potential users and promoting the uptake of radar data in operational contexts in general is necessary. Among various approaches, this can be achieved by creating tutorials or giving workshops. A different path is the development of tools that allow a hands-on experience by learning directly on the data. Tools of this sort can lower the entry barrier for exploiting radar capabilities for humanitarian applications.

This project follows the latter path by creating an interactive Google Earth Engine App that introduces radar imagery to non-expert users. The goal is to guide the user step-by-step through a process where radar imagery is displayed in different viewing modes so it can be visually explored as best as possible. By highlighting change through false color composites, the user gains a better feeling on how these changes appear in basic backscatter radar images. Finally, a digitization function allows to manually delineate the identified change and download it as file to integrate it in common GIS software.

2 Material and Methods

2.1 Data

The most important characteristic for data used in this project was that it is freely available as it should fall in line with the free and open access of the app itself. Therefore, all datasets included in the app come from the European Copernicus initiative (formerly GMES), an earth observation program to provide information on understanding and mitigating climate change effects, managing the environment and ensuring civil security (ESA, n.d.). With respect to the goal of the app to lower barriers for radar image analysis, the focus was set on data from the Sentinel-1 constellation. It comprises two polar-orbiting radar satellites that perform synthetic radar imaging in the C-band range of the electromagnetic spectrum. The combined repeat cycle is 6 days at the equator. Data is acquired in one of four different imaging modes. These allow the generation of images with a spatial resolution of down to 5m, swath widths with up to 400km and either single or dual-polarization (Sentinel Online, n.d.a). The data is available in four different processing levels, of which the Ground Range Detected (GRD) level is best suited for analysis by non-expert users as it requires no further special pre-processing. As it is analysis ready, the following pre-processing steps have been applied by the distributor before serving it out: Thermal noise removal, radiometric calibration, and terrain correction. The radar data is complemented by the Sentinel-2 constellation. The two satellites carry an optical instrument payload capable of multi-spectral imaging. Flying in a sun-synchronous polar orbit they reach a high revisit time of 5 days at the equator. In total they provide 13 spectral bands with spatial resolutions between 10 and 60m and a swath width of 290km. Data is available in two processing levels (Level-1C, Level-2A) with the first containing the Top-of-Atmosphere (TOA) reflectances while the latter contains Bottom-Of-Atmosphere (BOA) reflectance values (Sentinel Online, n.d.b). For the application the BOA level was chosen.

Although the data is freely available and can be downloaded at any time (for example through the Open Hub at scihub.copernicus.eu), it is laborious and mostly done for single scenes. Because the app aims at global coverage, data retrieval without requirements on the user side and simple integration of data in the app, the data is taken from the Earth Engine Data Catalog (developers.google, n.d.). This data archive is an integral part of the Earth Engine platform (see 2.2) and has both datasets used for the app available to be easily integrated. Dataset availability is given from 03.10.2014 onwards for Sentinel-1 and 28.03.2017 respectively for Sentinel-2. The most recent images are usually from the previous day.

2.2 The Platform

The Google Earth Engine (GEE) is a cloud-based computation platform that provides geospatial processing services on a planetary scale (Gorelick et al., 2017). The core feature is the multi-petabyte data catalog of analysis ready data coupled with high-performance computing resources. This allows both remote sensing scientists as well as a general broader audience to utilize technical capacities on a level that was previously out of reach. The main components of the Earth Engine are the public data catalog, the cloud computing infrastructure, geospatial APIs, and an app server.

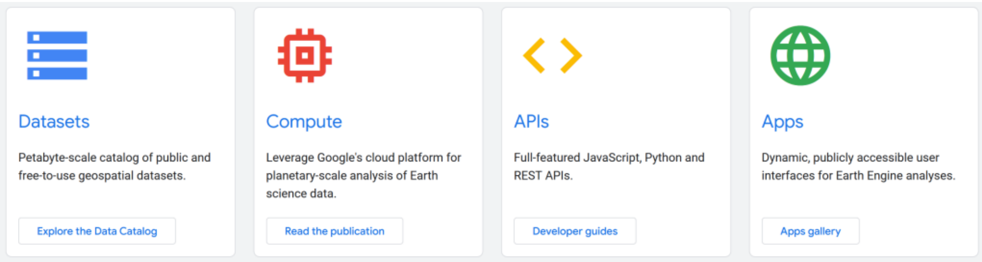


Figure 1: Overview on the Earth Engine components, developers.google.com/earth-engine/

Accessing the data catalog and the computation services is possible through one of the internet-accessible APIs or an interactive web-based development environment (IDE) that allows quick testing and visualizations of results. The most important aspects of the GEE platform for the app development are outlined in the following.

2.2.1 The Data Catalog

The GEE data archive is publicly available and contains multiple petabytes of geospatial datasets. This is to a large part remote sensing imagery acquired by various sensors including the entire Landsat and Sentinel-1/2 archives. It additionally includes datasets on topics such as land cover or climate forecasts but also vector data including country borders or socio-economic datasets. Continuous updating adds roughly 6000 scenes per day to the archive, with a usual latency of 1 day (Gorelick et al., 2017). Images with a relation such as acquired by the same sensor are grouped into an image collection. This is also the case for both datasets used in the app and has the benefit of allowing filtering of attributes such as dates, cloud cover, polarizations and many more.

2.2.2 Earth Engine Apps

The concept of GEE Apps is to provide dynamic and shareable interfaces of geospatial analyses in the Earth Engine to users. It allows both experts and non-experts to interact with the data catalog and analytical functionalities simply by interacting with interface elements like buttons, sliders, or text fields. The apps are easily shared out via an application-specific URL when the app is published. Accessing an app does not require a GEE account, making it accessible for a broad audience (Google Earth Engine Guides, n.d.).

2.3 Methods

The overall methodological approach of the app is to guide the user through the process of information visualization and extraction from radar imagery. Therefore, an interface is developed that implements a step-by-step workflow and guides the user through app functionalities. The schematic flow is shown in Figure 2.

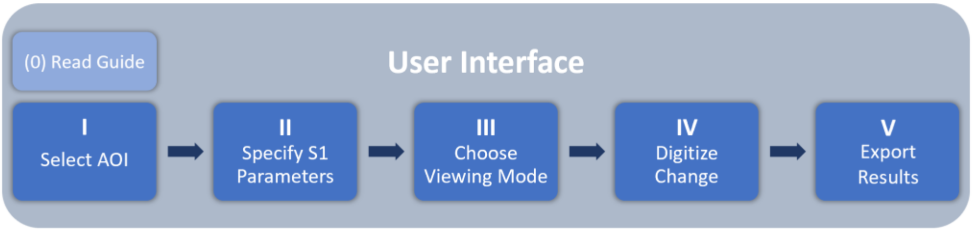


Figure 2: Schema of app workflow

Concerning geospatial methods, the focus is set on visualization techniques to highlight bi-temporal changes. This means on the image level the creation of false color composites for which the RGB band combination uses the image from timestep 1 for the red and green bands and timestep 2 as blue band. This leads to a colored image in which prominent changes between the timesteps appear blue, making it easily interpretable for a non-expert view. Additionally, two viewing modes are implemented representing different approaches for visual comparison of images. The first one is a split panel view which shows two images side by side, separated by a slider. Users can shift the slider from side to side, revealing more of one image while simultaneously overlaying the second one. This allows for fast and precise comparison of images. The second mode implements a linked maps approach in which four synchronized maps are included in the viewing panel. In addition to the radar imagery from both timesteps they include the false color composite and optical reference image from the second timestep. Although this approach reduces the map size, it provides the benefit of presenting an extended information space including optical validation data (if available). In combination it is hoped that users can quickly identify areas where change has occurred, both on detailed level as well as in the overall scene, and to visually validate their findings on optical data.

When areas of change are identified, the user can enter the digitization mode which utilizes the linked maps mode as predefined background. As the app was developed around the use case of dwelling detection, polygons are the only geometry option for digitization. Export functionalities include the option of naming the output file and the generation of a download link that allows saving the digitized data in GeoJSON format.

On the coding side, a modular approach was chosen by creating a separate script for processing functions such as the composite creation, and one for the user interface functions, including the definition of widgets and the respective styles. The main script where all comes together contains only very specific functionality that could not be transferred to other modules.

3 Results

The project results are summarized within the finished app. The interface presented to the user is displayed in Figure 3. It shows the two major components, the tool and information panel to the left and the map(s) panel to the right in which the interaction happens or the respective results are shown.

Figure 3: App user interface after initializing it.

With the focus on non-expert app users in mind, the first thing presented to the user in an introductory text with the option to open a small step-by-step guide that lines out how to use the app and how the respective functionalities work. Subsequently, the numbering indicates that now the area of interest (AOI) needs to be specified. This happens by simply drawing a polygon in the map panel. The map itself will focus and center around the drawn AOI. Next the user needs to set all relevant parameters surrounding radar data. Most important, however, are the two date periods to be investigated. Other parameters are the polarization, the instrument mode, and the orbit pass. With a single click a collection of images is created in the background, the radar mosaics for both time steps, a false color composite and the optical reference data. The map panel switches per default into a Switch Panel mode showing the time step 1 radar image in the left panel and the second time step image in the right one. Now a first inspection of differences between the scenes can be undertaken with the option of displaying the additional image composites. For a different view on the data, it is possible to switch into a Linked Maps view which shows all 4 images simultaneously. This allows for direct comparison of specific areas within the AOI. Subsequently, the digitization button starts the digitization session by jumping into the Linked Maps view and allowing manual drawing of polygons in the false color composite map. This way it is possible to focus on highlighted areas to delineate while simultaneously performing visual checks on the radar and optical imagery to increase delineation accuracy. After the drawing session is finished, the data collected by the user can be downloaded via a link in GeoJSON format. This allows easy integration of the dataset into common GIS software packages. Each key step is visually summarized in Figure 4.

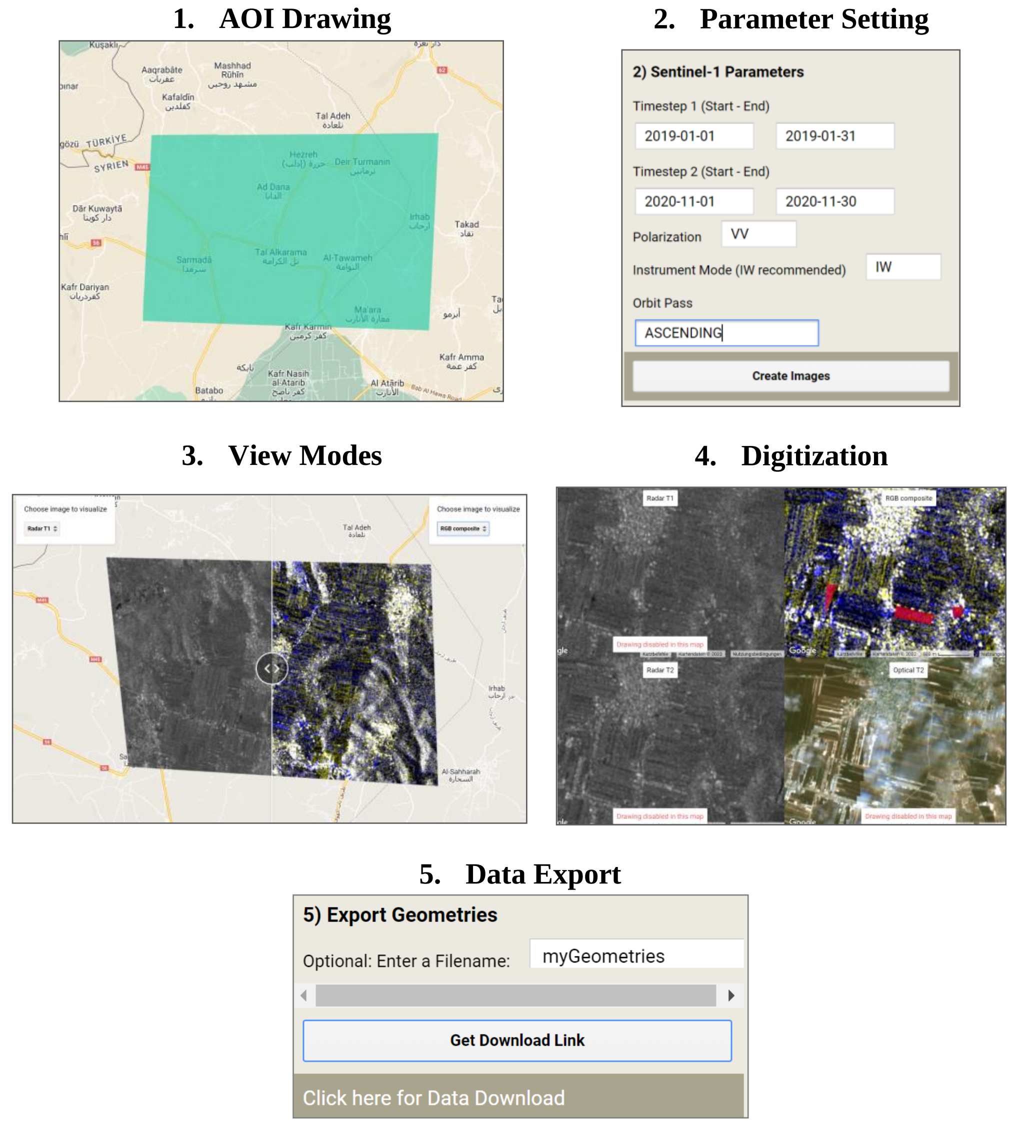


Figure 4: App process flow overview.

4 Discussion

The app fulfills its declared goal of lowering the entry barrier to information visualization and information extraction from radar data. However, by focusing on the backscatter values, the app only scratches the surface of the information contained in satellite radar imagery. This is a natural limit in the GEE environment as only the GRD backscatter data is available in the catalog, but it should not neglect the many other ways of extracting information from radar data. Most notable are interferometry and polarimetry from which the former uses the phase information in the signal to, for example, measure height displacements or generate digital elevation models (Gens & Van Genderen, 1996). The latter approach uses changes in polarizations caused by different scattering mechanisms of ground objects for a variety of applications like in agriculture (McNairn & Brisco, 2004). To fully exploit these methods requires firstly the more complex data products at a lower processing level (single look complex – SLC) and secondly specialized software that is able to handle the complexity of the data and provides the respective tools for the analysis. Although both is freely available in the form of public data such as Sentinel-1 and the SNAP software package (*SNAP - ESA Sentinel Application Platform*, n.d.), the workflow remains complex and requires a certain amount of understanding of radar/SAR-specific characteristics. Therefore, these methods are not likely to be used by non-expert users and theirs absence in the app functionalities is accepted.

Currently the app is focused on the use case of dwelling detection as it was built around the idea of detecting newly erected dwellings as consequence of movement of displaced people. This information can help humanitarian organizations to better estimate the number and location of people in need and plan relief efforts more efficiently. To support humanitarian operations even better the app would need to provide functionalities to extract information for other use cases as well. With respect to radar´s valuable capability to penetrate clouds, methods to generate information on flood cases would be a useful extension to the app as floods usually go hand in hand with heavy cloud cover. Because this prohibits fast image retrieval from optical sensors, radar can fill this data gap.
In terms of app functionality this could be realized by implementing methods for flood mask extraction. This could mean an automated approach like Otsu thresholding (Otsu, 1979) where separating the image back- and foreground is done by calculating the optimal threshold. A rather manual method would be a user-defined threshold that can be fine-tuned by providing immediate visual feedback in the map or complete manual digitization by the user. The most user-friendly approach could also be a combination in which the Otsu methods provides a suggested threshold that can then be adjusted by the user.

5 Conclusion and Outlook

This project developed a GEE app that lowers the entry barrier for non-experts to explore radar data as complementary EO data source and to extract information of it. By guiding the app user through a predefined workflow, a simple entry to learning how radar imagery can look like, what and how objects can be detected and how simple changes are identified is created. It thereby successfully demonstrated the benefits of including radar data in an operational context. Additionally, it serves as a streamlined application for observing an area of interest quickly and extracting manual delineated geospatial objects from. The intuitive use is supported by hiding all complexity from the user, meaning it requires no complex environment setup or inclusion of additional software packages. As also no user account of any sort is required for app usage, there are no barriers left for anyone to access it and explore radar imagery.

Looking into the future a variety of further developments come to mind for improving the app´s functionality and robustness. Most notably is the extension in functionality towards other application cases. The first app version was built around the case of dwelling/settlement detection, but radar data can support humanitarian organizations in many other of their operation fields. A prominent example would be the case of flood

detection. The unique radar imaging technique enables the detection of larger water surfaces as dark areas quite well and can ensure continued data acquisition through potential cloud cover. With extending on the functionality side comes the restructuring of the user interface that gains complexity. The respective interface demands to be far more adaptive to the user´s choices than it currently is and needs to be improved alongside extended functionality. A rather smaller but nevertheless useful extension, especially for non-expert users would be to include an example case for each functionality (dwellings, floods) that demonstrates how each steps work, from AOI drawing to data export.
Although the list could go on, it is concluded here with the suggestion of improving the robustness in terms of resistance against erroneous user inputs. While currently this issue results in no constructive feedback to the user, it simply does not work or the app needs reloading. This could be mitigated by including more precise feedback and implementing security check such as parameter validation during input.
It is hoped that contributions in the form of tools like this can promote the uptake of radar data in operation context by humanitarian organizations to better exploit the existing and freely available earth observation capacities.

Acknowledgements

I would like to acknowledge the supervision and guidance by Mr. Herrman Klug over the course of the I3 project. Considerable effort was made in form of personal consulting and feedback for intermediate products and throughout the project development steps. Acknowledged are also the valuable contributions from the fellow students of the course whose feedback helped to consider new aspects and looking at the project from different perspectives.

References

Braun, A., & Hochschild, V. (2017). Potential and Limitations of Radar Remote Sensing for Humanitarian Operations. *GI_Forum*, *1*, 228–243. https://doi.org/10.1553/giscience2017_01_s228

developers.google. (n.d.). *Earth Engine Data Catalog*. https://developers.google.com/earth-engine/datasets/

ESA. (n.d.). *Europe´s Copernicus programme*. https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Europe_s_Copernicus_programme

Gens, R., & Van Genderen, J. L. (1996). Review Article SAR interferometry—Issues, techniques, applications. *International Journal of Remote Sensing*, *17*(10), 1803–1835. https://doi.org/10.1080/01431169608948741

Google Earth Engine Guides. (n.d.). *Earth Engine Apps*. https://developers.google.com/earth-engine/guides/apps

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, *202*, 18–27. https://doi.org/10.1016/j.rse.2017.06.031

Lang, S., Füreder, P., Riedler, B., Wendt, L., Braun, A., Tiede, D., Schoepfer, E., Zeil, P., Spröhnle, K., Kulessa, K., Rogenhofer, E., Bäuerl, M., Öze, A., Schwendemann, G., & Hochschild, V. (2020). Earth observation tools and services to increase the effectiveness of humanitarian assistance. *European Journal of Remote Sensing*, *53*(sup2), 67–85. https://doi.org/10.1080/22797254.2019.1684208

Lang, S., Schoepfer, E., Zeil, P., & Riedler, B. (2017). Earth Observation for Humanitarian Assistance. *GI_Forum*, *1*, 157–165. https://doi.org/10.1553/giscience2017_01_s157

McNairn, H., & Brisco, B. (2004). The application of C-band polarimetric SAR for agriculture: A review. *Canadian Journal of Remote Sensing*, *30*(3), 525–542.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Sentinel Online. (n.d.a). *Mission Summary*. https://sentinel.esa.int/web/sentinel/missions/sentinel-1/overview/mission-summary

Sentinel Online. (n.d.b). *Sentinel-2 Product Types*. https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/product-types

SNAP - ESA Sentinel Application Platform (v8.0). (n.d.). [Computer software]. http://step.esa.int

Voigt, S., Giulio-Tonolo, F., Lyons, J., Kučera, J., Jones, B., Schneiderhan, T., Platzeck, G., Kaku, K., Hazarika, M. K., Czaran, L., Li, S., Pedersen, W., James, G. K., Proy, C., Muthike, D. M., Bequignon, J., & Guha-Sapir, D. (2016). Global trends in satellite-based emergency mapping. *Science*, *353*(6296), 247–252. https://doi.org/10.1126/science.aad8728

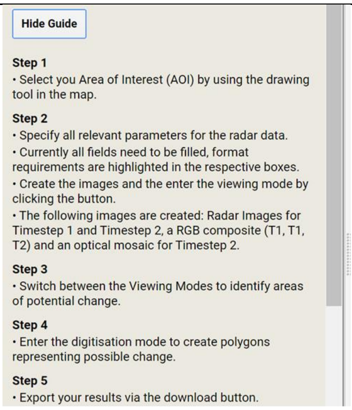
Appendix I

All documents surrounding this project, including code files and documentation in wiki form can be accessed in a GitLab repository here.

Appendix II

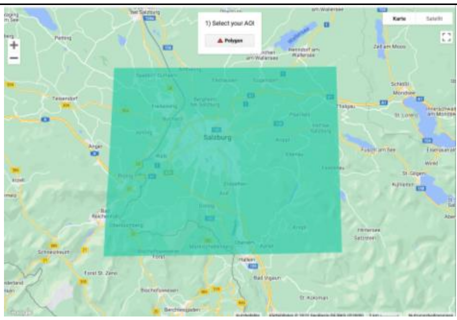
Step-By-Step Description

Step 0 (optional, but recommended): Read Guide



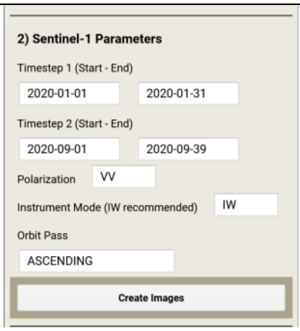
The guide provides short descriptions on what to do or what to expect in each step of the app. To conserve space, it is collapsed by default.

Step 1: Draw AOI



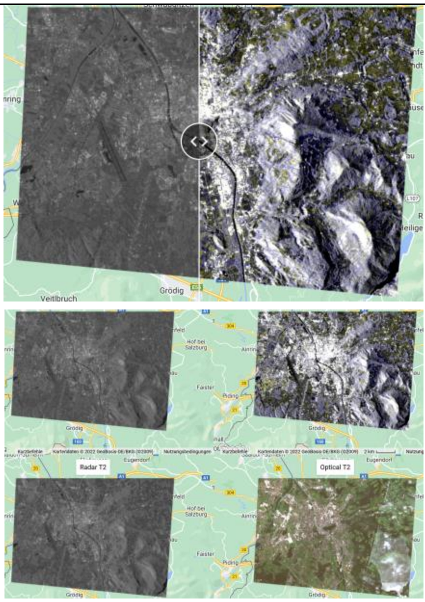
Drawing the AOI is done by digitizing a polygon. When the polygon is closed, the map will automatically center and zoom to the drawn shape.

Step 2: Specify Parameters



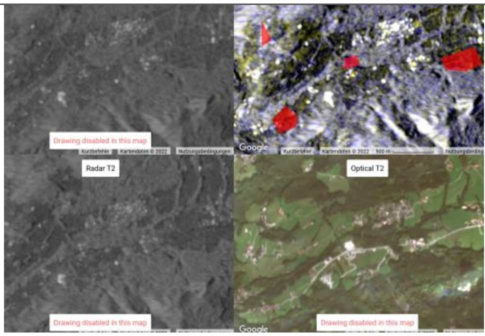
Parameter specification is crucial as it defines the data that is investigated. In terms of radar data this means differences between polarizations, instrument mode and orbit passes. The expected input format and available options are suggested in the textboxes. Adhering to these formats is mandatory.

Step 3: Investigate Change in Viewing Modes



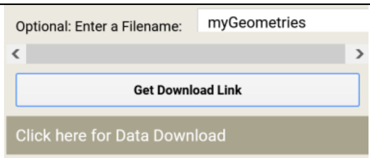
The two viewing modes implemented (Linked Maps & Split Panel) provide different approaches on how to interactively visualize data. Both can be switched back and forth.

Step 4: Digitize Change



The manual digitization of change is also polygon based as most use cases include area-based objects. The digitization mode needs to be specifically started and finished.

Step 5: Export Data



The export of the created features is possible via a download link. Individual file naming is possible, otherwise a default name is used. A GeoJSON file is created that can be used in common GIS systems.