SPECTRAL: AWESOME SPECTRAL INDICES DEPLOYED VIA THE GOOGLE EARTH ENGINE JAVASCRIPT API

David Montero[1,*], César Aybar[2], Miguel D. Mahecha[1,3], and Sebastian Wieneke[1]

[1]Remote Sensing Centre for Earth Systems Research (RSC4Earth), University of Leipzig, Talstraße 35, 04317 Leipzig, Germany
(david.montero, miguel.mahecha, sebastian.wieneke)@uni-leipzig.de
[2]Department of Geoinformatics (Z_GIS), University of Salzburg, 5020 Salzburg, Austria, csaybar@gmail.com
[3]Helmholtz Centre for Environmental Research, Leipzig, 04318 Leipzig, Germany

Commission IV, WG IV/4

KEY WORDS: Remote Sensing, Spectral Indices, Google Earth Engine, JavaScript, Earth Systems.

ABSTRACT:

Spectral Indices derived from Remote Sensing (RS) data are widely used for characterizing Earth System dynamics. The increasing amount of spectral indices led to the creation of spectral indices catalogues, such as the Awesome Spectral Indices (ASI) ecosystem. Google Earth Engine (GEE) is a cloud-based geospatial processing service with an Application Programming Interface (API) that is accessible through JavaScript (Code Editor) and Python. Tools for computing indices, including raster operations, normalized differences, and expression evaluation methods have been developed in the API. However, users still have to hard-code spectral indices for the JavaScript library since there are no implementations that link catalogues of spectral indices to the Code Editor. Here we present *spectral*, a module that links the Awesome Spectral Indices (ASI) catalogue to GEE for querying and computing spectral indices inside the Code Editor. The module allows accessing and computing spectral indices from the catalogue for multiple remote sensing products in GEE. All indices can be queried by using a key-value model and computed by using a single method. The module demonstrates that spectral indices can be easily computed inside the Code Editor. Image and Image Collection objects can be used for the calculation of all spectral indices in the catalogue if the specific dataset counts with the required bands. We anticipate that *spectral* will be used by most GEE users for Earth System research. Analyses conducted by the community will be sped up by avoiding hard-coding and RS investigations will be boosted.

1. INTRODUCTION

Monitoring the Earth's surface is a constant labor in Remote Sensing. Earth Systems research using Earth Observation (EO) data products are widely done at different scales using satellite imagery from multiple platforms, e.g., (Baumann, 2010; Guo et al., 2016; Crowley and Cardille, 2020; Ustin and Middleton, 2021). These platforms provide data as multispectral bands at different wavelengths of the electromagnetic spectrum, e.g., visible, Near-Infrared (NIR), and Shortwave Infrared (SWIR). The availability of different sensors led to the creation of a variety of spectral indices for a better monitoring of Earth in different application domains: vegetation, water, urban, snow, etc. Popular spectral indices have been created for investigating vegetation, e.g., Normalized Difference Vegetation Index (NDVI, Rouse et al., 1974); water bodies, e.g., Normalized Difference Water Index (NDWI, McFeeters, 1996); and other application domains like urban monitoring, e.g., Normalized Difference Built-Up Index (NDBI, Zha et al., 2010). The number of spectral indices is constantly increasing due to (1) modifications and enhancement of existent indices for specific tasks, e.g., improving corrections of undesired effects (Huete, 1988; Badgley et al., 2017; Camps-Valls et al., 2021), and (2) the launch of new satellite missions with newer bands in different regions of the spectrum, e.g., Red Edge (Dash and Curran, 2004; Frampton et al., 2013; Wang et al., 2018; Jiang et al., 2021). The increasing amount of spectral indices led to the creation of spectral indices catalogues, such as Awesome Spectral Indices (ASI), an open source ecosystem that consists of a curated spectral indices catalogue and a Python library for querying and computing spectral indices.

In recent years, the explosion of web-optimized access formats has facilitated image access by enabling clients to request and conveniently process only specific regions of a large image by submitting Hypertext Transfer Protocol (HTTP) range requests. The SpatioTemporal Asset Catalog (STAC) specification (Radiant Earth Foundation, 2021) is dependent on this technology. STAC provides a simple design for reading metadata utilizing JavaScript Object Notation (JSON) and Cloud Optimized GeoTIFF (COG) for reading the internal pixel structure. Different EO platforms rely on STAC, such as Amazon Web Services (AWS) and Planetary Computer. Using the *spyndex* Python library, STAC user clients can compute spectral indices from the ASI ecosystem. However, due to the enclosed form of its API, the Google Earth Engine (GEE, Gorelick et al., 2017) web-based Integrated Development Environment (IDE), named Code Editor, is out of reach. GEE is one of the most popular cloud-based platforms for processing raster and vector data, and the GEE JavaScript API is widely used for rapid scripting and the creation of interactive web apps. The GEE API counts with several methods for computing spectral indices. These include mathematical raster operations, normalized differences, and expression evaluation methods in the case of more sophisticated indices. Third-party modules such as *geetools* (Principe, 2019) and *GEET* (Lacerda, 2022) have made use of these methods and implemented functions for computing spectral indices inside the Code Editor. Nevertheless, no implementation links comprehensive spectral indices catalogues, such as the ASI ecosystem, to the Code Editor for computing spectral indices in the

* Corresponding author

GEE JavaScript API.

The aim of this paper is to present the potential of *spectral*, a JavaScript module for accessing and computing spectral indices from the ASI ecosystem inside the GEE Code Editor. The module uses the ASI catalogue to create a dictionary on the client-side of the API that users can easily access. Additionally, the module implements methods to automatically compute spectral indices using datasets from the GEE catalogue as well as user-owned datasets inside the Code Editor. This introduces a whole new approach for computing more than 200 spectral indices (as of version 0.1.0 of the ASI ecosystem) using the GEE JavaScript API that is currently not available with other implementations.

This document is structured as follows: Section 2 introduces the *spectral* module and shows how it can be used inside the Code Editor; in Section 3 we demonstrate how the module can be used to create a GEE we app; in Section 4 we discuss the potential of the module as well as future developments, and in Section 5 we give the main conclusions.

2. JAVASCRIPT MODULE: *SPECTRAL*

spectral is a module for the GEE JavaScript API that works as an extension of the ASI ecosystem, linking the spectral indices catalogue to the Code Editor. This module mimics the `spyndex` Python library and allows users to query and compute spectral indices inside the GEE Code Editor (Figure 1). On the client-side of the API, the catalogue's spectral indices are automatically saved as dictionary objects. This enables users to access and compute numerous indices simultaneously. In addition, workflows involving computations of spectral indices for a single image can be embedded into a mapping pipeline for image collections.

2.1 Querying Spectral Indices and Parameters

As the spectral indices from the catalogue are saved as a dictionary object on the client-side of the API, they can be accessed easily by using keys or dot notation. The spectral indices as well as their attributes can be accessed through this object, named `indices`. Thus, if the module is required and stored in a variable named *spectral*, an index can be called from the spectral indices object (e.g. the NDVI can be queried as `spectral.indices.NDVI`) (Figure 1b). This also applies to the index attributes, which can be queried from each of the spectral indices directly. Since the "reference" attribute of all indices from the ASI catalogue is a URL string, it is parsed as such in the Code Editor and displayed as a link in the console when printed, or inside widgets on the map when working with User Interface (UI) design objects.

In the case of the bands, they are stored in a dictionary object named `bands`. This follows the same structure as the objects in the Python library of the ecosystem. Bands can be queried from this object following the standard created in the ecosystem (e.g. the NIR band can be queried as `spectral.bands.N`). Attributes of the band, such as the name and specific platform attributes (e.g. center wavelength, bandwidth, etc.) can also be queried from each specific band. This is useful for spectral indices that require the wavelength of a specific band for its computation.

In the same line, additional parameters that also follow the standard of the ecosystem are stored in a dictionary object named `constants`. From this object, users can query the attributes of the parameters, including their default value (i.e. default value specified in the literature). Users can query parameters as `spectral.constants.L`, where L is the canopy background adjustment factor in this example. Parameters for kernel indices are also stored in this object, this includes the scale-length parameter in the Radial Basis Function (RBF) Kernel (K_{RBF}) and the polynomial degree in the Polynomial Kernel (K_{poly}). Although all parameters included here have default values, users can choose to use them or to optimize them for their specific needs.

2.2 Computing Spectral Indices

2.2.1 Methods: Expressions in the GEE API can be evaluated by using the `ee.Image.expression` method, which uses a dictionary of bands representing the operands in the expression to evaluate. Since all spectral indices in the catalogue have a curated expression representing the formula of each specific index stored in the "formula" attribute, this expression can be used for their evaluation using the GEE API.

By taking advantage of this, a single spectral index can be computed by passing the required expression and the dictionary of required parameters. In *spectral*, this is leveraged by the `computeIndex` method, which receives the image, the acronym of the index, and the dictionary of parameters (Figure 1a). This method becomes very flexible as it doesn't strictly require the use of bands from the image in the first argument, meaning that additional images representing parameters or images from other datasets can also be used if the use case requires it.

In addition, the `computeIndex` method allows the computation of multiple spectral indices by passing an array of indices' acronyms instead of a single index acronym. In this scenario, the user must pass the required arguments for all indices to compute in the dictionary of required parameters. This becomes handy when various indices are needed for a specific analysis, allowing a high-level scripting and a significant reduction in code length.

Furthermore, generalized kernel indices can also be computed using this method. Nevertheless, it is required to pass the specific kernels for the required parameters in the computation of these spectral indices. For this, the module is equipped with the `computeKernel` method, which allows the computation of kernels $K(a, b)$, where K is a kernel function, and a and b are bands or parameters. There are three kernel functions implemented in *spectral*: (1) the Linear Kernel, which is given by:

$$K_{\text{linear}}(a, b) = ab \tag{1}$$

(2) the Polynomial Kernel:

$$K_{\text{poly}}(a, b) = (ab + c)^p \tag{2}$$

where c = trade-off parameter
p = polynomial degree

and (3) the RBF Kernel:

$$K_{\text{RBF}}(a, b) = \exp\left(-\frac{(a-b)^2}{2\sigma^2}\right) \tag{3}$$

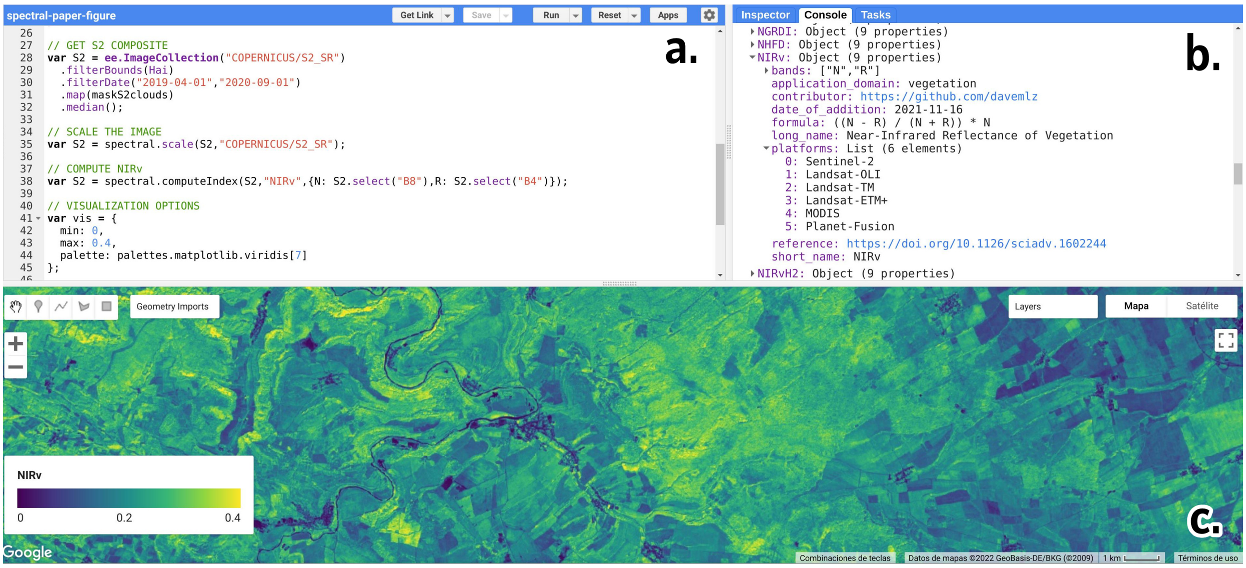

Figure 1. Functioning of the module inside the GEE Code Editor. a) Script tab that shows the functions used for computing spectral indices, b) console tab showing the set of spectral indices of the catalogue after being queried (the NIRv is displayed with its attributes as an example), c) map visualization of the NIRv computed using the module for Sentinel-2 over the Hainich National Park in Central Germany.

where σ = length-scale parameter

Note that the K_{RBF} is suggested for the computation of the kNDVI with $\sigma = 0.5(\rho_{\text{NIR}} + \rho_{\text{red}})$ (Camps-Valls et al., 2021). However, users are free to optimize kernel parameters.

2.2.2 Compatibility with Earth Engine objects: The expression evaluation method in the GEE API is available for `ee.Image` as well as `ee.Number` objects in the server-side. Currently, *spectral* only supports the calculation of spectral indices using the `ee.Image` objects inside the GEE Code Editor. These objects can either represent constant as well as variable values (i.e. constant parameters and actual remote sensing images). Additionally, objects that can be parsed as `ee.Image` objects can also be used (e.g. numerical values on the client-side, or `ee.Number` objects on the server-side).

Since `ee.ImageCollection` objects do not count with the expression evaluation method, they can't be used for spectral indices computation directly. Nevertheless, users can create a function that computes spectral indices for single `ee.Image` objects and then map it across an image collection with the required parameters for the spectral indices computation. Anonymous functions can also be directly passed to the mapping function to avoid bounding it to an identifier.

2.3 Scaling and offsetting raster products

The multi-petabyte GEE catalogue counts with numerous remote sensing collections that can be used for spectral indices computation. Various collections are scaled and offset such that a 16-bit integer datatype is achieved for storage. However, multiple spectral indices require that the reflectances (or digital levels in the case of RGB imagery) are strictly bounded to $[0, 1]$.

Generally, users have to manually scale and offset images or image collections in order to compute spectral indices. Nevertheless, *spectral* provides users with two methods to automatically scale and offset any image from the GEE catalogue. The methods `scale` and `offset` of the module require the image as first argument, and the collection identifier as the second argument. The scale and offset values are automatically extracted from the GEE STAC and saved as dictionary objects for each collection and band.

3. CASE STUDY

Here we present how the module can be used to compute multiple spectral indices inside the GEE Code Editor by creating a simple GEE App. The aim of this case study is to show the potential of *spectral* for Earth Systems research by using MODIS as a test case for computing different spectral indices in four application domains. Users can access the GEE App publicly at https://dmlmont.users.earthengine.app/view/spectral.

3.1 App description

3.1.1 Data: We used the MCD43A4 product from the GEE catalogue. This product has three bands in the visible (blue, green, and red), one band in the NIR, and three bands in the Shortwave Infrared spectrum (SWIR). However, from the three SWIR bands just bands 6 and 7 were used. All bands are corrected to Nadir BRDF Adjusted Reflectance (NBAR) at a spatial resolution of 500 m and a daily temporal resolution. The standard cloud mask of the product was used and a median composite for winter 2020 was created. From this composite 8 spectral indices were computed for 4 application domains. Vegetation Indices (VI): NDVI and NIR Reflectance of Vegetation (NIRv, Badgley et al., 2017), Water Indices (WI): NDWI and Modified NDWI (MNDWI, Xu, 2006), Urban Indices: NDBI and Enhanced Modified Bare Soil Index (EMBI, Zhao and Zhu, 2022), and Snow Indices: Normalized Difference Snow Index (NDSI, Riggs et al., 1994) and Snow Water Index (SWI, Dixit et al., 2019). The complete list of indices can be found in the Appendix section.

3.1.2 UI: The App is divided into 8 panels, a pair for a specific application domain: vegetation, water, urban, and snow.

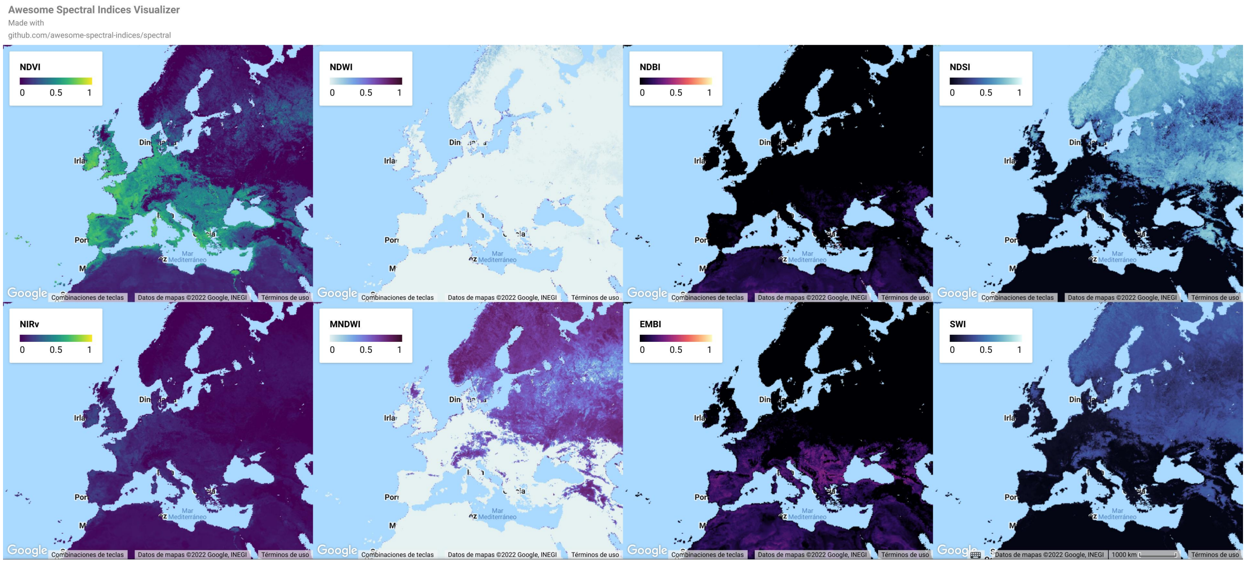

Figure 2. Screenshot of the Awesome Spectral Indices Visualizer App. Column 1 shows vegetation indices: NDVI and NIRv. Column 2 shows water indices: NDWI and MNDWI. Column 3 shows urban and bareness indices: NDBI and EMBI. Column 4 shows snow indices: NDSI and SWI.

For each pair of panels, a map of a spectral index is displayed with a specific color palette according to the application domain. The spectral indices that are shown in each pair of panels correspond to each application domain. The acronym of the index as well as the colorbar indicating their values is also displayed.

3.2 Results

Figure 2 shows a screenshot of the GEE App with its 8 panels. As mentioned previously, each pair of panels shows two spectral indices for each application domain. The visualization was centered at Europe, but users can navigate over the whole world. The screenshot shows that north-east of Europe is covered by snow (represented by the snow indices), while the south-west shows snow-free low vegetation values (represented by VIs). Since small water bodies are not visible at the presented scale the NDWI shows values below zero. MNDWI and NDSI show the same patterns as their formulas are equivalent, although they are used for different application domains. In the case of the urban indices, NDBI values are below zero at the visualized scale, however, bare soil effects are visible. These bare soil areas are enhanced by EMBI since it aims to discriminate them from urban areas. These indices are valuable for the previously listed application domains, and their display on a public GEE app makes it simple to show the results of Earth science investigations.

4. DISCUSSIONS

We here present the benefits of linking the ASI ecosystem with the GEE JavaScript API. We discuss how the GEE catalogue can be exploited by using the *spectral* module and how it can serve Earth Systems research. Furthermore, we elaborate how spectral indices can be included in Machine Learning (ML) pipelines using the GEE API as well as future developments.

4.0.1 Exploiting the GEE Catalogue: Besides the infrastructure behind GEE, the GEE catalogue represents a key feature in terms of remote sensing resources. Multiple remote sensing collections are included in the GEE catalogue and the potential of exploiting them to compute spectral indices increase with the variety of data. We expect that users will compute spectral indices for the most used satellite imagery platforms. This includes the Landsat Collection 2 (e.g. Landsat 4-5 TM, Landsat 7 ETM+, Landsat 8-9 OLI) as well as Sentinel-2 imagery. However, coarse resolution products such as MODIS (e.g. MCD43A4, MOD09GQ, MOD09GA) and VIIRS (e.g. VNP09GA) are also expected to be exploited. Furthermore, as GEE allows users upload their own datasets, remote sensing products that are not usually included in the GEE catalogue can also be used inside the GEE Code Editor. This includes satellite imagery from private companies (e.g. Planet, RapidEye) and drone imagery.

4.0.2 Machine Learning: ML methods are being widely used for several application domains in Earth Systems research. The ASI ecosystem as well as the GEE catalogue can be further exploited by computing spectral indices and using them as inputs in ML pipelines. This is possible inside the GEE Code Editor since the GEE API counts with several ML tools for this kind of approaches. Ensemble methods (e.g. Random Forest, Gradient Boosting), kernel methods (e.g. Support Vector Machines), and probabilistic methods (e.g. Naive Bayes) are available in the GEE API and they can be used with the outputs from the *spectral* module.

4.0.3 Apps: One pivotal advantage of the GEE JavaScript API is that it can be used for creating public web apps. As it was shown in the case study, *spectral* can be used for creating GEE Apps quickly. The interactivity of the UI widgets provided by the GEE API can be combined with the multiple spectral indices available in the ASI catalogue, and the several attributes can also be exploited. This allows researchers and private users to explore spectral indices using interactive visualizations displayed as GEE Apps.

4.0.4 Future developments: Further improvements of the *spectral* module are planned. For example, users can just select the spectral indices to compute without passing the dictionary of required bands for a specific platform. As passing a dictionary enhances the flexibility of the module for the big amount of collections in the GEE catalogue, an automatic implementation for computing spectral indices for the most used satellite platforms would reduce drastically coding times. This means that built-in methods can automatically select the bands from collections such as Landsat, Sentinel, or MODIS according to the ASI standard to compute spectral indices.

5. CONCLUSIONS

In order to exploit the spectral indices catalogue from the ASI ecosystem inside the GEE JavaScript API it is required to have a connection module that serves as a link between both ecosystems. Here we presented *spectral*, a JavaScript module for the GEE JavaScript API to query and compute spectral indices from the ASI ecosystem. The module uses the catalogue from the ASI ecosystem and stores it as a dictionary object on the client-side. Furthermore, the module implements user-friendly methods for automatically computing indices from the ASI catalogue using the GEE catalogue's products. We expect that the module will be used by the remote sensing community for computing spectral indices extensively using the GEE JavaScript API for Earth Systems research. ML pipelines will be boosted using inputs from the module and final results will be displayed as GEE Apps for outreach. Although the module is planned to be further developed, it already offers an overarching tool for computing spectral indices in GEE.

ACKNOWLEDGEMENTS

This project was financially supported by Niedersächsisches Vorab (ZN 3679), Ministry of Lower-Saxony for Science and Culture (MWK). We would like to thank the authors and creators of the multiple spectral indices cited in this paper and that are part of the ASI ecosystem. We also thank the providers of public remote sensing data such as NASA and USGS, as well as the GEE team.

REFERENCES

Badgley, G., Field, C. B., Berry, J. A., 2017. Canopy near-infrared reflectance and terrestrial photosynthesis. *Science Advances*, 3(3). doi.org/10.1126/sciadv.1602244.

Baumann, P., 2010. The OGC web coverage processing service (WCPS) standard. *GeoInformatica*, 14(4), 447–479. doi.org/10.1007/s10707-009-0087-2.

Camps-Valls, G., Campos-Taberner, M., Moreno-Martínez, Á., Walther, S., Duveiller, G., Cescatti, A., Mahecha, M. D., Muñoz-Marí, J., García-Haro, F. J., Guanter, L., Jung, M., Gamon, J. A., Reichstein, M., Running, S. W., 2021. A unified vegetation index for quantifying the terrestrial biosphere. *Science Advances*, 7(9), 7447–7473. doi.org/10.1126/sciadv.abc7447.

Crowley, M. A., Cardille, J. A., 2020. Remote Sensing's Recent and Future Contributions to Landscape Ecology. *Current Landscape Ecology Reports*, 5(3), 45–57. doi.org/10.1007/s40823-020-00054-9.

Dash, J., Curran, P. J., 2004. The MERIS terrestrial chlorophyll index. *International Journal of Remote Sensing*, 25(23), 5403–5413. doi.org/10.1080/0143116042000274015.

Dixit, A., Goswami, A., Jain, S., 2019. Development and Evaluation of a New "Snow Water Index (SWI)" for Accurate Snow Cover Delineation. *Remote Sensing*, 11(23), 2774. doi.org/10.3390/rs11232774.

Frampton, W. J., Dash, J., Watmough, G., Milton, E. J., 2013. Evaluating the capabilities of Sentinel-2 for quantitative estimation of biophysical variables in vegetation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 82, 83–92. doi.org/10.1016/J.ISPRSJPRS.2013.04.007.

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. doi.org/10.1016/J.RSE.2017.06.031.

Guo, H., Wang, L., Liang, D., 2016. Big Earth Data from space: a new engine for Earth science. *Science Bulletin*, 61(7), 505–513. doi.org/10.1007/s11434-016-1041-y.

Huete, A. R., 1988. A soil-adjusted vegetation index (SAVI). *Remote Sensing of Environment*, 25(3), 295–309. doi.org/10.1016/0034-4257(88)90106-X.

Jiang, W., Ni, Y., Pang, Z., Li, X., Ju, H., He, G., Lv, J., Yang, K., Fu, J., Qin, X., 2021. An Effective Water Body Extraction Method with New Water Index for Sentinel-2 Imagery. *Water*, 13(12), 1647. doi.org/10.3390/w13121647.

Lacerda, E., 2022. sacridini/GEET: Google Earth Engine Toolbox - Library to write small EE apps or big/complex apps with a lot less code.

McFeeters, S. K., 1996. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *International Journal of Remote Sensing*, 17(7), 1425–1432. doi.org/10.1080/01431169608948714.

Nguyen, C. T., Chidthaisong, A., Diem, P. K., Huo, L. Z., 2021. A Modified Bare Soil Index to Identify Bare Land Features during Agricultural Fallow-Period in Southeast Asia Using Landsat 8. *Land*, 10(3), 231. doi.org/10.3390/land10030231.

Principe, R., 2019. fitoprincipe/geetools-code-editor: A set of tools to use in Google Earth Engine Code Editor (JavaScript).

Radiant Earth Foundation, 2021. radiantearth/stac-spec: SpatioTemporal Asset Catalog specification - making geospatial assets openly searchable and crawlable.

Riggs, G. A., Hall, D. K., Salomonson, V. V., 1994. Snow index for the Landsat Thematic Mapper and moderate resolution imaging spectroradiometer. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 4, 1942–1944. doi.org/10.1109/IGARSS.1994.399618.

Rouse, J. W., Haas, R. H., Schell, J. A., Deering, D. W., 1974. Monitoring vegetation systems in the Great Plains with ERTS. Technical report, NASA.

Ustin, S. L., Middleton, E. M., 2021. Current and near-term advances in Earth observation for ecological applications. *Ecological Processes*, 10(1), 1–57. doi.org/10.1186/s13717-020-00255-4.

Wang, X., Xie, S., Zhang, X., Chen, C., Guo, H., Du, J., Duan, Z., 2018. A robust Multi-Band Water Index (MBWI) for automated extraction of surface water from Landsat 8 OLI imagery. *International Journal of Applied Earth Observation and Geoinformation*, 68, 73–91.

Xu, H., 2006. Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery. *International Journal of Remote Sensing*, 27(14), 3025–3033. doi.org/10.1080/01431160600589179.

Zha, Y., Gao, J., Ni, S., 2010. Use of normalized difference built-up index in automatically mapping urban areas from TM imagery. *International Journal of Remote Sensing*, 24(3), 583–594. doi.org/10.1080/01431160304987.

Zhao, Y., Zhu, Z., 2022. ASI: An artificial surface Index for Landsat 8 imagery. *International Journal of Applied Earth Observation and Geoinformation*, 107, 102703. doi.org/10.1016/J.JAG.2022.102703.

APPENDIX

Table 1 shows the formulas and references of the indices presented in the case study. For all indices, ρ_i is the reflectance in band i. The band naming follows the standard of the ASI ecosystem.

Index	Formula	Reference
NDVI	$\frac{\rho_N - \rho_R}{\rho_N + \rho_R}$	(Rouse et al., 1974))
NIRv	$\text{NDVI} \times \rho_N$	(Badgley et al., 2017)
NDWI	$\frac{\rho_G - \rho_N}{\rho_G + \rho_N}$	(McFeeters, 1996)
MNDWI	$\frac{\rho_G - \rho_{S1}}{\rho_G + \rho_{S1}}$	(Xu, 2006)
NDBI	$\frac{\rho_{S1} - \rho_N}{\rho_{S1} + \rho_N}$	(Zha et al., 2010)
MBI	$\frac{\rho_{S1} - \rho_{S2} - \rho_N}{\rho_{S1} + \rho_{S2} + \rho_N} + 0.5$	(Nguyen et al., 2021)
EMBI	$\frac{\text{MBI} - \text{MNDWI} - 0.5}{\text{MBI} + \text{MNDWI} + 1.5}$	(Zhao and Zhu, 2022)
NDSI	$\frac{\rho_{S1} - \rho_N}{\rho_{S1} + \rho_N}$	(Riggs et al., 1994)
SWI	$\frac{\rho_G(\rho_N - \rho_{S1})}{(\rho_G + \rho_N)(\rho_N + \rho_{S1})}$	(Dixit et al., 2019)

Table 1. Spectral indices used in the Awesome Spectral Indices Visualizer GEE App.