Rindcalc - A Spectral Index Raster Calculator

Release 2.0.1

Sep 17, 2020

Contents

CHAPTER 1

About

Rindcalc is an open source python library built on NumPy and GDAL with the goal of providing seamless raster index calculations and composites of satellite imagery for remote sensing. It looks to fill the gap left by proprietary softwares and open source initiatives alike when it comes to the need to create and process spectral index raster files.

CHAPTER 2

Satellites & Imagery

- Landsat-8
- National Agricultural Imagery Program - NAIP
- Sentinel-2

CHAPTER 3

Example of use:

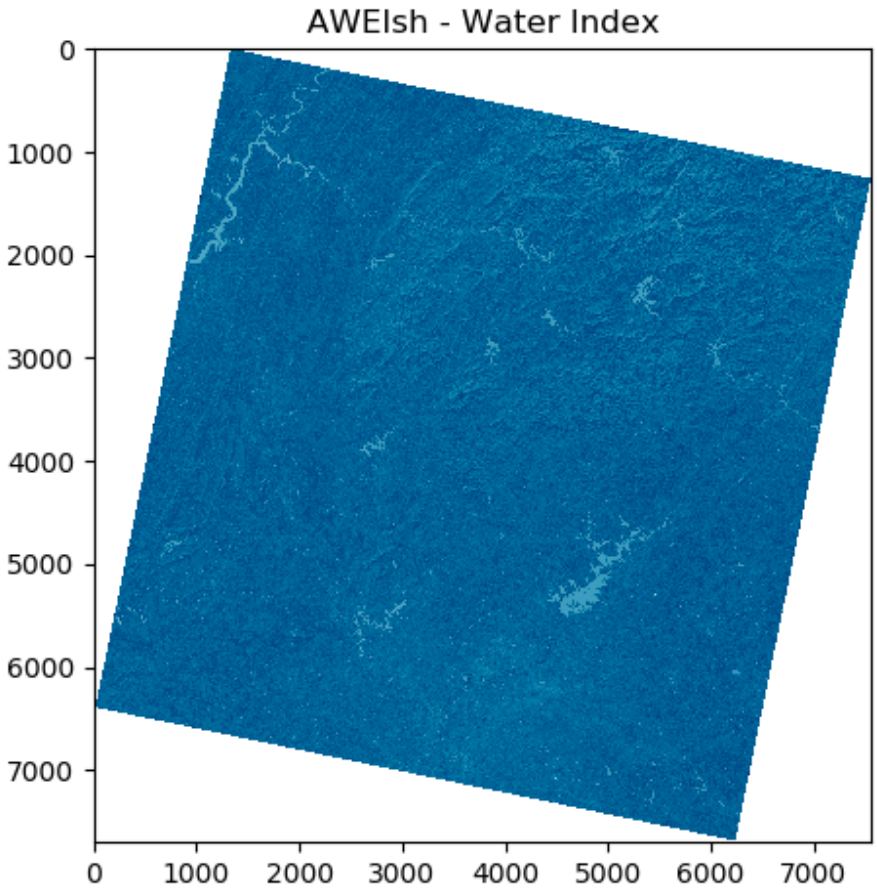
3.1 Calculating the ARVI of a NAIP tile and saving as a raster.

```
import rindcalc as rc

# set inputs and outputs
input_naip = '/naip_folder/m_3008101_ne_17_1_20151017.tif'
output_arvi = '/naip_outputs/ARVI_3008101_ne_17.tif'

data = rc.NAIP(path)
data.ARVI(output_ndvi)
```

Output ARVI raster:

3.2 Using in conjunction with matplotlib

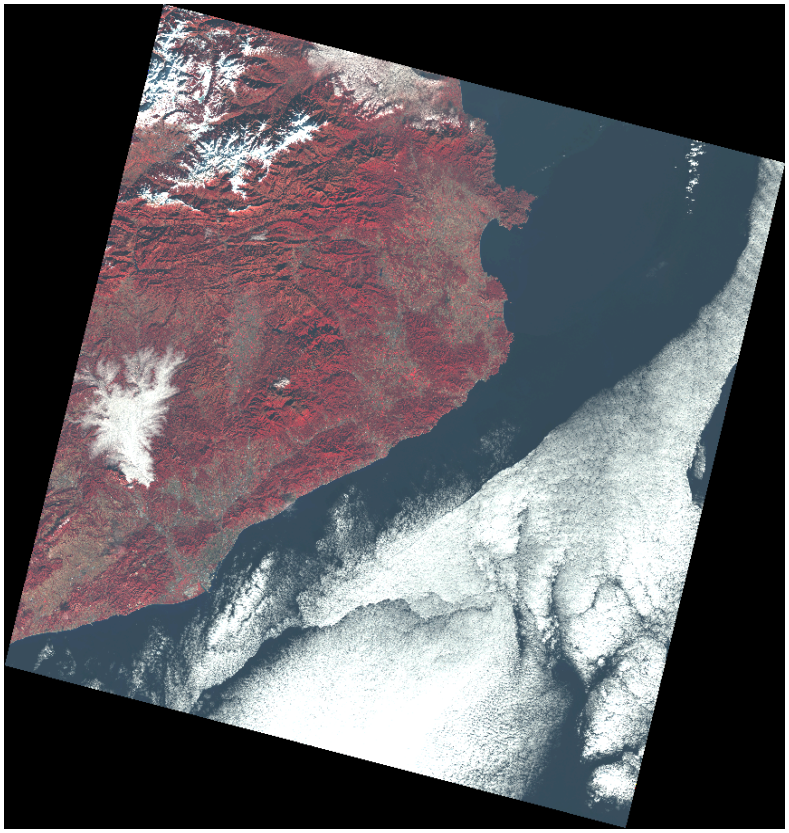
```
from rindcalc import Landsat
import matplotlib.pyplot as plt

ls = '/landsat_8/2019_11_28'
index = Landsat(ls).AWEIsh('/landsat_8/2019_11_28')
plt.imshow(index, 'ocean')
plt.title('AWEIsh - Water Index')
plt.show()
```



3.3 Creating a false color composite of a Landsat-8 Scene.

```
from rindcalc import Landsat

ls = Landsat('/landsat_8/LC08_L1TP_197031_20131212_20170428_01_T1')
ls.composite(['band_5', 'band_4', 'band_3'], '/landsat_8_outputs/FalseColor_Barcelona.
↪tif')
```

Output false color composite:



CHAPTER 4

Install

4.1 With pip from PyPI repository

PyPI repository

Dependencies

- GDAL (v 3.0.0 or greater)
- NumPy (v 1.0.0 or greater)

```
pip install rindcalc
```

4.2 With Conda from Anaconda Cloud

Conda Cloud

```
conda install -c rindcalc rindcalc
```

4.3 Latest development version

For latest version clone the Rindcalc GitHub Repository and add the module to path with sys.path.append.

4.3.1 Landsat-8

Rindcalc uses the standard naming convention of landsat bands, it only needs the folder in which Landsat-8 bands are contained as the input. This method allows for easy, quick, and consistent index calculations from Landsat-8 imagery.

The Landsat 8 satellite orbits the the Earth in a sun-synchronous, near-polar orbit, at an altitude of 705 km (438 mi), inclined at 98.2 degrees, and circles the Earth every 99 minutes. The satellite has a 16-day repeat cycle with an equatorial crossing time: 10:00 a.m. +/- 15 minutes.

Landsat 8 aquires about 740 scenes a day on the Worldwide Reference System-2 (WRS-2) path/row system, with a swath overlap (or sidelap) varying from 7 percent at the Equator to a maximum of approximately 85 percent at extreme latitudes. The scene size is 185 km x 180 km (114 mi x 112 mi) (USGS).

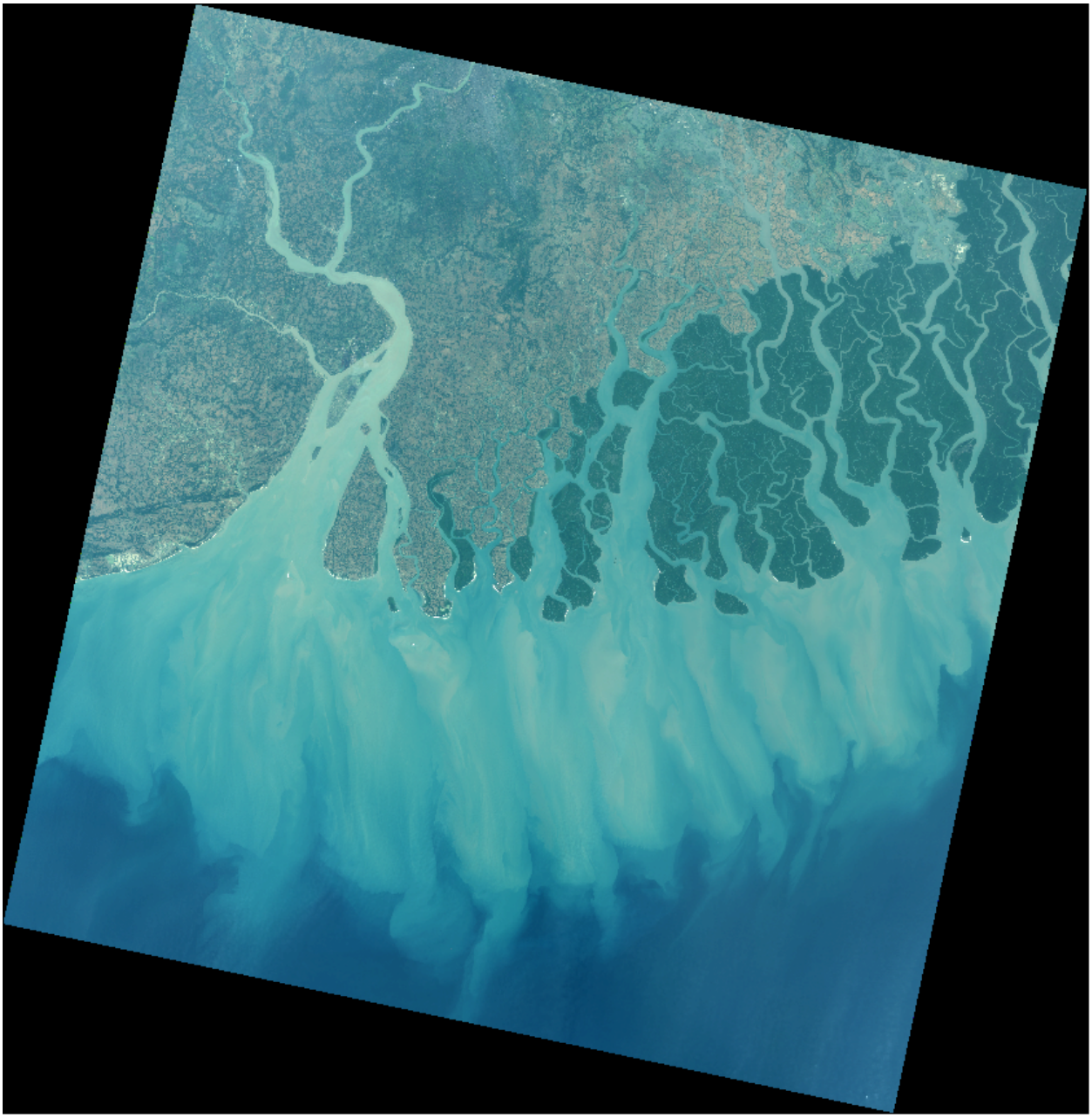


Fig. 1: Landsat-8 scene composite created with rindcalc RGB composite function.

class rindcalc.**Landsat**(*path*)

__init__(*path*)

Class to read and write Landsat-8 data from.

Parameters **path** (*str*) – Path to folder where Landsat-8 bands are contained.

path

Dictionary of the path for each Landsat-8 band.

Type dict

bands

Dictionary of arrays for the bands chosen to load.

Type dict, array

band_options

List of all options for band input names.

Type list

load_bands(*which_bands=None*)

Opens and reads bands into Float 32 arrays. If no list is passed into *which* bands then all bands are opened and added to the dictionary *self.bands*.

Parameters **which_bands** (*list, optional*) – A list of band names to open as arrays. e.g. which_bands=['band_1', 'band_2', 'band_3']

Returns **self.bands** – Updated self.bands dictionary

Return type dict

composite(*which_bands*, *out_composite*)

Creates a three band composite out of the specified bands.

Parameters

- **which_bands** (*list*) – A list of bands to save as a three band composite. Must be in order of how the bands are to saved within the output TIFF. e.g. which_bands=['band_1', 'band_2', 'band_3']
- **out_composite** (*str*) – The output filename to save the composite,

NDVI(*out_raster=None*, *mask_clouds=False*)

Calculates NDVI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

AWEIsh(*out_raster=None*, *mask_clouds=False*)

" Calculates AWEIsh index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

AWEInsh(*out_raster=None*, *mask_clouds=False*)
Calculates AWEInsh index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

NDMI(*out_raster=None*, *mask_clouds=False*)
Calculates NDMI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

MNDWI(*out_raster=None*, *mask_clouds=False*)
Calculates MNDWI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

GNDVI(*out_raster=None*, *mask_clouds=False*)
Calculates GNDVI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

SAVI(*out_raster=None*, *soil_brightness=0.5*, *mask_clouds=False*)
Calculates SAVI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **soil_brightness** (*float*) – Soil brightness factor to compute SAVI with. Defaults to 0.5
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

ARVI(*out_raster=None*, *mask_clouds=False*)
Calculates ARVI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

VARI(*out_raster=None*, *mask_clouds=False*)
Calculates VARI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

NDBI(*out_raster=None*, *mask_clouds=False*)
Calculates NDBI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

NDBaI(*out_raster=None*, *mask_clouds=False*)
Calculates NDBaI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

NBLI(*out_raster=None*, *mask_clouds=False*)
Calculates NBLI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

EBBI (*out_raster=None*, *mask_clouds=False*)
Calculates EBBI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

UI (*out_raster=None*, *mask_clouds=False*)
Calculates UI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

NBRI (*out_raster=None*, *mask_clouds=False*)
Calculates NBRI index

Parameters

- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.
- **mask_clouds** (*bool, optional*) – Whether or not to apply cloud masking to output index with the QA band.

Returns **equation** – Output array of the generated index.

Return type array

4.3.2 NAIP

The National Agriculture Imagery Program (NAIP) acquires aerial imagery during the agricultural growing seasons in the continental U.S. A primary goal of the NAIP program is to make digital ortho photography available to governmental agencies and the public within a year of acquisition.

The default spectral resolution is natural color (Red, Green and Blue, or RGB) but beginning in 2007, some states have been delivered with four bands of data: RGB and Near Infrared (USDA).

class rindcalc.**NAIP** (*path*)

__init__ (*path*)
Class to read and write NAIP data from.

Parameters **path** (*str*) – Path to folder where Sentinel-2 bands are contained.

path
Dictionary of the path for each Landsat-8 band.
Type dict

bands
Dictionary of arrays for the bands chosen to load.
Type dict, array

band_options
List of all options for band input names.
Type list

load_bands(*which_bands=None*)
Opens and reads bands into Float 32 arrays. If no list is passed into *which* bands then all bands are opened and added to the dictionary *self.bands*.

Parameters **which_bands** (*list, optional*) – A list of band names to open as arrays. e.g. which_bands=['band_1', 'band_2', 'band_3']

Returns **self.bands** – Updated self.bands dictionary

Return type dict

composite(*which_bands*, *out_composite*)
Creates a three band composite out of the specified bands.

Parameters

- **which_bands** (*list*) – A list of bands to save as a three band composite. Must be in order of how the bands are to saved within the output TIFF. e.g. which_bands=['band_1', 'band_2', 'band_3']
- **out_composite** (*str*) – The output filename to save the composite,

NDVI(*out_raster=None*)
Calculates NDVI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

ARVI(*out_raster=None*)
Calculates ARVI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

VARI(*out_raster=None*)
Calculates VARI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

SAVI(*soil_brightness=0.5*, *out_raster=None*)
Calculates SAVI index

Parameters

- **soil_brightness** (*float*) – Soil brightness factor to compute SAVI with. Defaults to 0.5
- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

RedRatio(*out_raster=None*)

Calculates ARVI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

4.3.3 Sentinel-2

"The Copernicus Sentinel-2 mission comprises a constellation of two polar-orbiting satellites placed in the same sun-synchronous orbit, phased at 180° to each other. It aims at monitoring variability in land surface conditions, and its wide swath width (290 km) and high revisit time (10 days at the equator with one satellite, and 5 days with 2 satellites under cloud-free conditions which results in 2-3 days at mid-latitudes) will support monitoring of Earth's surface changes. The coverage limits are from between latitudes 56° south and 84° north." - (European Space Agency)

class rindcalc.**Sentinel**(*path*)

__init__(*path*)

Class to read and write Sentinel-2 data from.

Parameters **path** (*str*) – Path to folder where Sentinel-2 bands are contained.

path

Dictionary of the path for each Sentinel-2 band.

Type dict

bands

Dictionary of arrays for the bands chosen to load.

Type dict, array

band_options

List of all options for band input names.

Type list

load_bands(*which_bands=None*)

Opens and reads bands into Float 32 arrays. If no list is passed into *which* bands then all bands are opened and added to the dictionary *self.bands*.

Parameters **which_bands** (*list, optional*) – A list of band names to open as arrays. e.g. which_bands=['band_1', 'band_2', 'band_3']

Returns **self.bands** – Updated self.bands dictionary

Return type dict

composite(*which_bands*, *out_composite*)

Creates a three band composite out of the specified bands.

Parameters

- **which_bands** (*list*) – A list of bands to save as a three band composite. Must be in order of how the bands are to saved within the output TIFF. e.g. which_bands=['band_1', 'band_2', 'band_3']
- **out_composite** (*str*) – The output filename to save the composite,

AWEIsh(*out_raster=None*)
Calculates AWEIsh index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

NDVI(*out_raster=None*)
Calculates NDVI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

SIPI(*out_raster=None*)
Calculates SIPI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

ARVI(*out_raster=None*)
Calculates ARVI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

NDI45(*out_raster=None*)
Calculates NDI45 index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

MTCI(*out_raster=None*)
Calculates MTCI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

MCARI(*out_raster=None*)
Calculates MCARI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

GNDVI(*out_raster=None*)
Calculates GNDVI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

PSSR(*out_raster=None*)
Calculates PSSR index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

S2REP(*out_raster=None*)
Calculates S2REP index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

IRECI(*out_raster=None*)
Calculates IRECI index

Parameters **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

SAVI(*soil_moisture=0.5*, *out_raster=None*)
Calculates SAVI index

Parameters

- **soil_moisture** (*flt*) – Soil moisture value, default = 0.5
- **out_raster** (*str, optional*) – Output filepath for calculated TIFF.

Returns **equation** – Output array of the generated index.

Return type array

4.3.4 Band Utilities

rindcalc.utils.resample.**resample**(*band*, *cell_size*, *out=None*)
Utility function to resample a raster and output it as either an array or TIFF

Parameters

- **band** (*str*) – Path to raster file to resample
- **cell_size** (*int*) – New size of the cells
- **out** (*str, optional*) – Filename to save the output TIFF

Returns resampled array

rindcalc.utils.gen_stats.**gen_stats**(*raster_path*)
Prints minimum, maximum, mean, median, and standard deviation values for a raster.

Parameters **raster_path** (*str, required*) – input raster with which to generate statistical summary of.

Returns

- *minimum*
- *maximum*
- *mean*
- *median*
- *standard deviation*

4.3.5 Index Formula List

All index formulas are grouped by specific use here.

Water Indices

Indices designed for water detection

Automated Water Extraction Index | AWEIsh

- For areas with increased shadow.
- **AWEIsh = ((Blue + 2.5 * Green - 1.5 * (NIR + SWIR1) - 0.25 * SWIR2)) /** (Blue + Green + NIR + SWIR1 + SWIR2))

Automated Water Extraction Index | AWEInsh

- For areas with minimal shadow.
- **AWEInsh = ((4 * (Green - SWIR1) - (0.25 * NIR + 2.75 *** SWIR1)) / (Green + SWIR1 + NIR))

Normalized Difference Moisture Index | NDMI

- NDMI = ((NIR - SWIR1) / (NIR + SWIR1))

Modified Normalized Difference Water Index | MNDWI

- MNDWI = ((Green - SWIR1) / (Green + SWIR1))

Vegetation Indices

Indices designed for vegetation detection

Normalized Difference Vegetation Index | NDVI

- NDVI = (NIR - Red) / (NIR + Red)

Green Normalized Difference Vegetation Index | GNDVI

- GNDVI = (NIR - Green) / (NIR + Green)

Atmospherically Resistant Vegetation Index | ARVI

- ARVI = (NIR - (2 * Red) + Blue) / (NIR + (2 * Red) + Blue)

Visual Atmospherically Resistant Index | VARI

- VARI = ((Green - Red) / (Green + Red - Blue))

Soil Adjusted Vegetation Index | SAVI

- **SAVI = ((NIR - Red) / (NIR + Red + L)) x (1 + L)**
  - *L = Soil Brightness Factor*

Structure Insensitive Pigment Index | SIPI

- SIPI = (NIR – Blue) / (NIR – Red)

Urban / Landscape Indices

Indices designed for urban and landscape detection

Normalized Difference Built-up Index | NDBI

- NDBI = (SWIR1 - NIR) / (SWIR1 + NIR)

Nomrmalized Difference Bareness Index | NDBaI

- NDBaI = ((SWIR1 - TIR) / (SWIR1 + TIR))

Normalized Bare Land Index | NBLI

- NBLI = ((Red - TIR) / (Red + TIR))

Enhanced Built-up and Barness Index | EBBI

- EBBI = ((SWIR1 - NIR) / (10 * (np.sqrt(SWIR1 + tir))))

Urban Index | UI

- UI = ((SWIR2 - NIR) / (SWIR2 + NIR))

Burn / Fire Indices

Indices designed for fire and burned area detection

Normalized Burn Ratio Index | NBRI

- NBRI = ((NIR - SWIR2) / (NIR + SWIR2))

4.3.6 Contact

Owen Smith

Project GitHub: rindcalc

Email: ocsmit7654@ung.edu

Authors Owen Smith, University of North Georgia IESA

Version 2.0.5

License GPL v3.0

Python Module Index

r

Index

Symbols

A

B

C

E

G

I

L

M

N

P

R

S

U

V