sentipy

Feb 22, 2021

Contents

CHAPTER 1

Getting started

1.1 Installation

sentipy is hosted on pypy and can therefore be pip installed:

```
$ pip install sentipy
```

1.2 Usage

The intended responsibility of the **sentipy** project is to process pre-loaded Sentinel imagery into useful derivative indices such as those included in the S2 Toolbox (Fcover, Fapar, Canopy water). This is likely to expand into other spectral indices as the project grows.

We therefore do not involve ourselves in the loading of bands / pixels from imagery; our contract assumes that you will be able to provide pixel values for each band (as required) in the form of numpy (masked) arrays, and we take care of the processing logic.

Our preferred method for loading Sentinel imagery is to use the rasterio package.

```
from sentipy import s2_toolbox
import rasterio

with rasterio.open('example.tif') as dataset:
    # read your bands in & stack them into a single array here
    input_arr = np.stack(
        [
            dataset.read(1),
            dataset.read(2),
            dataset.read(3),
            # ...
            # and you'll need the image meta-data here too: view zenith, sun zenith,␣
↪rel azimuth
```

(continues on next page)

(continued from previous page)

```
        ]
        , axis=0
    )

# Initialise whichever calculator you wish to use, eg the Fapar calculator:
calculator = s2_toolbox.Fapar()
# All you have to do is call the calculator's `.run()` method on the input array and␣
↪you'll get an output array of
# matching size (but not shape) with your calculated value(s)
fapar = calculator.run(input_arr)
```

And that's really all there is to it!

CHAPTER 2

Examples

2.1 Processing a Sentinel-2 raster image into actionable info.

CHAPTER 3

API Reference

3.1 sentipy.settings

Settings and configuration that apply across the whole project

3.1.1 Sentinel 2 bands

https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric

Band label	Band code	Band description	Central wavelength	Resolution (m)
Band 1	“B01”	Coastal aerosol	0.443	60
Band 2	“B02”	Blue	0.49	10
Band 3	“B03”	Green	0.56	10
Band 4	“B04”	Red	0.665	10
Band 5	“B05”	Vegetation Red Edge	0.705	20
Band 6	“B06”	Vegetation Red Edge	0.74	20
Band 7	“B07”	Vegetation Red Edge	0.783	20
Band 8	“B08”	NIR	0.842	10
Band 8A	“B8a”	Vegetation Red Edge	0.865	20
Band 9	“B09”	Water vapour	0.945	60
Band 10	“B10”	SWIR - Cirrus	1.375	60
Band 11	“B11”	SWIR	1.61	20
Band 12	“B12”	SWIR	2.19	20

> The *Band code* is used to refer to Sentinel-2 bands throughout the project.

`sentipy.settings.`**`DEFAULT_BAND_SEQUENCE = ('B03', 'B04', 'B05', 'B06', 'B07', 'B8a', 'B11',`**

Default sequence of Sentinel-2 bands & meta-data assumed throughout project

3.2 sentipy.s2_toolbox

Python implementation of some Sentinel-2 toolbox processing tools.

Home for the objects & logic that deal with the calculation of FAPAR

class sentipy.s2_toolbox.**CanopyWater**
Bases: sentipy.s2_toolbox.S2BiophysicalCalculator

Calculator for the Canopy Water biophysical parameter.

VALIDATION_RANGES = {'B03': {'max': 0.263, 'min': 0.0}, 'B04': {'max': 0.3, 'min'

class sentipy.s2_toolbox.**Fapar**
Bases: sentipy.s2_toolbox.S2BiophysicalCalculator

Calculator for the Fapar biophysical parameter; the fraction of photosynthetically-active radiation absorbed (by vegetation).

VALIDATION_RANGES = {'B03': {'max': 0.263, 'min': 0.0}, 'B04': {'max': 0.3, 'min'

class sentipy.s2_toolbox.**Fcover**
Bases: sentipy.s2_toolbox.S2BiophysicalCalculator

Calculator for the Fcover biophysical parameter; the fraction of ground covered by vegetation.

class sentipy.s2_toolbox.**S2BiophysicalCalculator**
Bases: abc.ABC

run(*input_arr: <sphinx.ext.autodoc.importer._MockObject object at 0x7fa369017950>*, *band_sequence: List[str] = ('B03', 'B04', 'B05', 'B06', 'B07', 'B8a', 'B11', 'B12', 'COS_VIEW_ZENITH', 'COS_SUN_ZENITH', 'COS_REL_AZIMUTH')*, *validate: bool = True*) → Union[float, <sphinx.ext.autodoc.importer._MockObject object at 0x7fa368eeacd0>]
Run the calculator on an input array

By default, the calculator expects only the following bands to be passed in the sequence below:

- B03
- B04
- B05
- B06
- B07
- B8a
- B11
- B12
- COS_VIEW_ZENITH
- COS_SUN_ZENITH
- COS_REL_AZIMUTH

If band values are to be passed in a different set or sequence, the band_sequence parameter must be passed with band names (matching those above) for each element in the input array. eg. ["extra_band_1", "COS_SUN_ZENITH", "B03", "B04", ..., "COS_REL_AZIMUTH", "extra_band_2"]

Parameters

- **input_arr** – Input values for the calculator to use

- **band_sequence** – Names of bands included in the input array (names must match those used above for the required bands)
- **validate** – Flag for whether or not to apply validation ranges to the inputs

Returns Scalar estimate of the biophysical property

3.3 sentipy.agri_indices

A simple python API for processing raster arrays into common indices used in the agricultural domain.

There are a wide range of agriculturally-relevant indices to be processed from (multi-spectral) optical satellite imagery in general.

This module provides implementations of a number of indices, largely inspired by the Index Database and the SentinelHub custom scripts repository. They are implemented principally with Sentinel-2 in mind, but can be applied to imagery from other constellations where equivalent bands are available (it is left to the user to decide where this is the case).

Available implementations:

- NDVI
- NDWI
- OSAVI

`sentipy.agri_indices.`**`ndmi`**(*img_arr*, *band_sequence: tuple = ('B08', 'B11')*)
Normalized Difference Moisture Index

NDMI = (NIR - SWIR)/(NIR + SWIR) = (B08 - B11)/(B08 + B11)

`sentipy.agri_indices.`**`ndvi`**(*img_arr*, *band_sequence: tuple = ('B04', 'B08')*)
Normalized Difference Vegetation Index

NDVI = (NIR - RED)/(NIR + RED) = (B08 - B04)/ (B08 + B04)

`sentipy.agri_indices.`**`ndwi`**(*img_arr*, *band_sequence: tuple = ('B03', 'B08')*)
Normalized Difference Water Index

NDWI = -(GREEN - NIR)/(GREEN + NIR) = -(B03 - B08)/(B03 + B08)

`sentipy.agri_indices.`**`osavi`**(*img_arr*, *band_sequence: tuple = ('B04', 'B08')*)
Optimized Soil-Adjusted Vegetation Index Index DB link

OSAVI = (1+0.16)*(NIR - RED)/(NIR + RED + 0.16) = (1.16)*(B08 - B04)/(B08 + B04 + 0.16)

CHAPTER 4

Indices and tables

- genindex
- modindex
- search