

2.3_WQ_Oil_Spill_Detection

April 21, 2023



UNIVERSITY OF TWENTE.

1 Oil spill detection in water bodies

Author: Suhyb Salama, ITC

1.1 Purpose

This exercise introduces new remote sensing technology to detect water pollution resulting from oil spills. In this exercise you will use [Sentinel-1](#) observation in coastal waters. Sentinel-1 has a synthetic aperture radar (SAR) that transmits microwave signals in the *C band, centered at frequency of 5.405 GHz*, and receives back its echo or the backscattered signal from the Earth's surface. Because it is an active sensing with in the microwave region, it can observe through nights, clouds and rain, detecting changes in the water surface after events such as oil spill.

1.2 Prerequisites

- You should have an account in Google Earth Engine. If not done yet please [sign up now](#).
- You should have attended the lecture on oil spill mapping;

1.3 What you will learn

1. Access and Sentinel-1 observations;
2. Detect oil spill in water;
3. Visualize the results as maps.

1.4 What will you achieve

At the end of this exercise you will be able to use Google Earth Engine to access Sentinel-1 images, assess the surface roughness of the water and detect oil spill.

1.5 Introduction

Oil spill will dampen the roughness of the water surface, decreasing thereby the received backscattering signal.

1.5.1 Start

Load the required libraries and authenticate your Google account

```
[1]: # import Google earth engine module
import ee
import geemap
import geemap.colormaps as cm
Map =geemap.Map()
# import pandas as pd
# import matplotlib.pyplot as plt
```

```
<IPython.core.display.HTML object>
```

Enter verification code:

```
4/1AWtgzh7-r_0HwYhUxDX6hd-76c5xGkwQjqTGTFGmmY0eSGx7QmC78wfyn68
```

Successfully saved authorization token.

If you are not authenticated (logged in Google Earth) then you should acquire the authentication code, otherwise you can skip this cell

```
[2]: # Trigger the authentication flow.
ee.Authenticate()
```

```
# Initialize the library.
ee.Initialize()
```

```
<IPython.core.display.HTML object>
```

Enter verification code:

```
4/1AfgeXvu84Ndx7UwBkcZXLV9PH0czfuk7NmOKHBvkcoM0-JufJvRIWQbwqqA
```

Successfully saved authorization token.

1.6 Data access

Request Google Earth Engine to access [Sentinel 1 detected ground range](#). The retrieved images are calibrated and ortho-corrected products. To have a better understanding of the pre-processing required to handle Sentinel-1 data you can consult the [user guide of S1-SAR](#).

```
[2]: s1_coll = ee.ImageCollection("COPERNICUS/S1_GRD")
```

1.6.1 Inquire the band names of S1 image collection

- Select one image (the first);
- Inquire the bands names of this image and print.

```
[3]: first_image = s1_coll.first()
Bname = first_image.get('system:band_names')
print('system:band_names', Bname.getInfo())
```

```
system:band_names ['HH', 'HV', 'angle']
```

1.7 Filter the S image collection to:

- area of interest *AOI*
- time period *s_date* and *e_date*
- cloud percentage *cld_per* and probability *cld_prb*

```
[4]: aoi = ee.Geometry.Polygon([
    [
        9.121852596502947,
        43.20142752427495
    ],
    [
        9.121852596502947,
        42.55051822581004
    ],
    [
        10.657433730456233,
        42.55051822581004
    ],
    [
        10.657433730456233,
        43.20142752427495
    ],
    [
        9.121852596502947,
        43.20142752427495
    ]
])

s_date = '2018-10-08'
e_date = '2018-10-09'
```

```
[5]: # filter
s1_loc = s1_coll\
    .filterBounds(aoi)\\
    .filterDate(s_date, e_date)
```

```
[6]: count = s1_loc.size()
valCount=count.getInfo()
print('Number of imges is : ', str(valCount)+'\n')
```

Number of imges is : 2

- Display the mean/median in **GEEMAP** using the VV band

```
[7]: s1 = s1_loc.first().select('VV')
val= cm.palettes.Spectral
palette=val.default
Vis = {
    'min':-100,
    'max':-10,
    'palette': palette}
```

```
[8]: Map.addLayer(s1, Vis, 'S1',1)
#show the map
Map
```

```
Map(center=[20, 0], controls=(WidgetControl(options=['position',  
→'transparent_bg']), widget=HBox(children=(Togg...
```

1.8 Mask the oil spill based on a threshold value

```
[9]: threshold =s1.lt(-25)
mask=threshold.updateMask(threshold)
Map.addLayer(mask, {"palette":['Orange']}, 'Oil',1)
#show the map
Map
```

```
Map(bottom=754.0, center=[20, 0], controls=(WidgetControl(options=['position',  
→'transparent_bg']), widget=HBox(...
```

1.9 End of exercise

```
[ ]:
```