# CODE DOCUMENTATION

## Remote Sensing of Active Water Storage in Reservoirs and Lakes

**Authors**

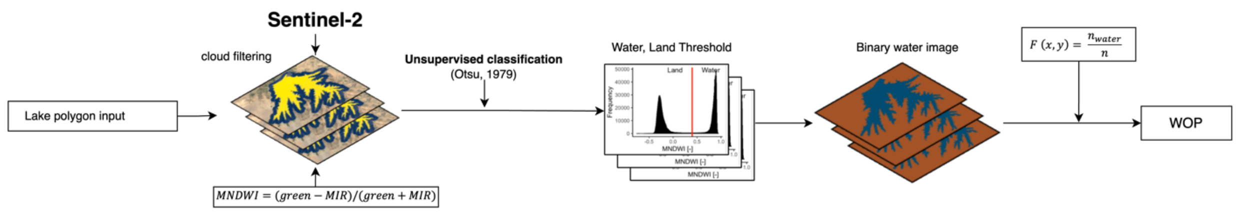
Adrian Kreiner
Silvan Ragettli, PhD

hydrosolutions GmbH
Venusstrasse 29
CH-8050 Zürich, Switzerland
Telefon: +41 43 535 05 80
email: ragettli@hydrosolutions.ch

# Table of Contents

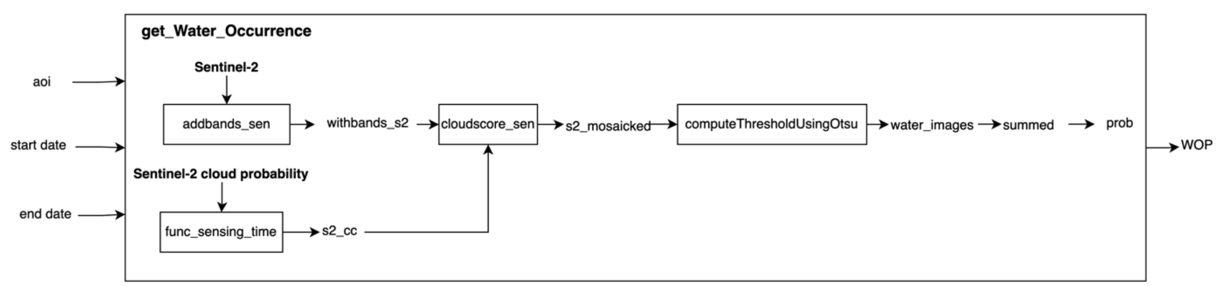
# 1 STEP 1: Surface Water Occurrence Probability

## 1.1 Description




The script export_SurfFreq uses Sentinel-2 images with a spatial resolution of 10m to obtain the water occurrence probability (WOP) over a specified period for specified areas. The areas are defined in a shapefile with the lakes and reservoirs as polygons. For each lake or reservoir, one water occurrence probability is calculated which is stored as an asset in the folder SurfFreqMaps and exported to Google Drive in the folder SurfFrequencyMaps. The water occurrence probability is the probability over a specified time period where the algorithm has specified this cell as water. Every cell has a value between 0 (was never a "water" cell) and 1 (was always a "water" cell). To obtain water occurrence images the script loops with each lake and reservoir over the function **get_Water_Occurrence** (see section 1.2).

## 1.2 get_Water_Occurrence




**Input:**

- Area of interest: defined as a polygon
- Start date
- End date

**Output:**

- One image of the water occurrence probability in the defined area and time period

**Description:**

The function **get_Water_Occurrence** obtains a set of Sentinel-2 satellite images (Sentinel-2 MSI: MultiSpectral Instrument, Level-1C) filtered by the start, end date, the area of interest, and the percentage of cloud cover. With the function **addbands_sen** (see section 1.2.1) the bands MNDWI and NDVI are calculated and added to every image as well as the bands B2, B3, B4, B8, B11, QA60 are renamed to B, G, R, NIR, SWIR1, QA60 and the Property SENSING_TIME is added.

The function retrieves a collection of cloud probability images for the same time and area of interest using the "COPERNICUS/S2_CLOUD_PROBABILITY" Image Collection. These images have one band containing the probability that the pixel is cloudy with values from 0 to 100. With the function **func_sensing_time** the property SENSING_TIME in the yyy-mm-dd format is added.

These cloud images are then mosaicked with the Sentinel-2 images to produce a single image for each time step and rename the band for the cloud probability to 'probability'.

→ **s2_mosaicked**

With the function **cloudscore_sen** (see section 1.2.2) the properties "cloud_cover" and "nodata_cover" are computed and the band "cloud" is added to the image. The images are then filtered by these properties with the two thresholds cc_tresh (5%) and nodata_thresh (10%).

For each image in s2_mosaiked, the following steps are performed:

- the band "cloud" and "MNDWI" is selected.
- MNDWI values greater than the maxValue of 0.3 are set to 0.3 because the function should not detect edges between free water and vegetation overgrown water.
- Only values with lower values than 100% within the "cloud" band are considered.
- A threshold th is computed for the MNDWI images using the function **computeThresholdUsingOtsu** (see section 1.2.3).
- The MNDWI image is masked with the computed threshold and only values greater than th are kept, resulting in a binary water mask image

The resulting image collection of binary water mask images is named **water_images**.

In the final step, the sum of all water_images is divided by the total number of images to obtain the water occurrence probability.

### 1.2.1 addbands_sen

Input:

- Sentinel-2 image

Output:

- Sentinel-2 image with:
  - Bands: B, G, R, NIR, SWIR1, QA60, MNDWI, NDVI
  - Property: SENSING_TIME

The function addbands_sen takes in a single Sentinel-2 satellite image as input and returns a modified version of the image with added bands. It performs the following steps

1. Calculates the modification of the normalized difference water index (MNDWI) for land–water classification (Xu, 2006) using the bands B3 and B11 and naming the band "MNDWI".
2. Calculates the "Normalized Difference Vegetation Index" (NDVI) using the bands B8 and B4 and renaming the resulting image band to "NDVI".
3. Selects the bands ["B2", "B3", "B4", "B8", "B11", "QA60"] and renames them to ["B", "G", "R", "NIR", "SWIR1", "QA60"].
4. Adds the "MNDWI" and "NDVI" bands to the image and sets a new property "SENSING_TIME" equal to the date of the image sensing time, formatted as "YYYY-MM-dd".

### 1.2.2 cloudscore_sen

The cloudscore_sen function takes the mosaicked Sentinel-2 image with the Sentinel-2 Cloud Probability as input and calculates the cloud cover of the image. The function calculates the cloud cover by creating masks that define which pixels in the image should be considered as clouds and then aggregates the masked image using the mean reducer to get the cloud cover percentage.

Input:

- Mosaicked and renamed Sentinel-2 image (s2_mosaicked)

Output:

- Image with added:
  - Bands:
    - cloud
    - probability
  - Properties:
    - nodata_cover: the percentage of pixels in the image that has no data.
    - cloud_cover: the percentage of pixels in the image that is considered clouds.
    - CLOUD_COVER: the percentage of pixels in the image that is considered as cloudy according to the "CLOUDY_PIXEL_PERCENTAGE" property of the input image.
    - SATELLITE: the type of satellite used to acquire the image, which is always "SENTINEL-2".

The function works as follows:

The function creates three “cloud” masks using the QA60 band and blue band (B) of the Sentinel-2 image and the probability band of the Sentinel-2 Cloud Probability combined within the function get_Water_Occurrence. The QA60 band contains information about the presence of opaque clouds (Bit 10, value 1024 means it has opaque clouds) or cirrus clouds (Bit 11, value 2048 means it has cirrus clouds) with a resolution of 60m.

The following masks named “cloud” are created:

1. Mask
   - 100 if: QA60 not equal to 1024, probability > 50

    - 0 if: B > 0
2. Mask 2:
    - 100 if: B > 0, probability > 50
    - 0 if: QA60 not equal to 1024
3. Mask 3:
    - 100 if: B > 0, probability > 50
    - 0 if: QA60 smaller than 1024

Mask2 is added as a band to the image named “cloud”. The mean of “mask” and “mask2” is added as properties with the names “nodata_cover”, respectively “cloud_cover”. It adds also the properties CLOUD_COVER, obtained from the input image property CLOUD_PIXEL_PERCENTAGE sets the SATELLITE property to SENTINEL-2

### 1.2.3 computeThresholdUsingOtsu

The function computeThresholdUsingOtsu returns a threshold value using the Otsu Method for an input image to separate between water and land pixel.

Input:

- image
- scale
- bounds: The region over which to perform the computation, given as a geometry object.
- cannyThreshold: The threshold value to be used in the Canny edge detection algorithm.
- cannySigma: The standard deviation to be used in the Canny edge detection algorithm.
- minValue: The minimum value to be used in the computation.
- debug: A Boolean indicating whether debugging information should be printed.

Output:

- threshold value.

**The function:**

First, the Canny edge filter (Canny 1986) is used to reduce the sampling region to only areas near water–land edges, because The Otsu method requires a bimodal histogram distribution and if land dominates the image over water, the histograms are unbalanced.

The Canny method first uses a Gaussian filter to smooth the image to remove the noise and then finds edges by looking for local maxima of the image intensity gradients.

Second, the **otsu** function is applied using the Otsu method with the identified areas. The Otsu method is a variation of the unsupervised k-means algorithm and separates pixels into two classes and returns a single value threshold. In this case, the function is used to distinguish between water and non-water pixels.

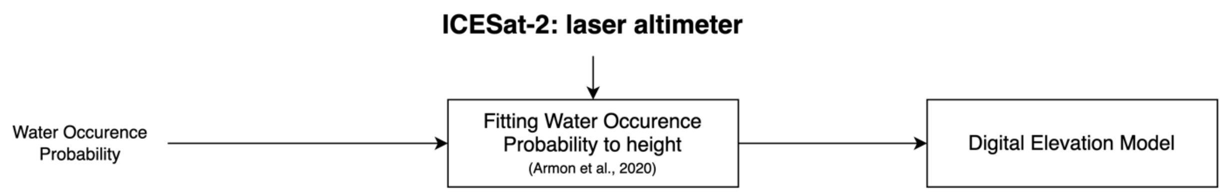
# 2 STEP 2: Download ICESat-2 Data

The R scripts, Icesat_DEM_atl08_20m, and Icesat_DEM_atl13, are utilized to download the ICESat-2 data products, ATL08 and ATL13, respectively. These downloads are facilitated by the Python scripts, download_icesat2_atl08_20m, and download_icesat2_atl13_per_groundtrack_par_adj.

The ICESat-2 ATL08 data product provides information about the height of the Earth's surface. The ICESat-2 ATL13 data product provides information about the height of the Earth's surface in areas covered by vegetation.

Both data products are stored after downloading and used in combination with the computed water occurrence probability to generate a digital elevation model of the lakes and reservoirs.

# 3 STEP 3: DEM Generation




corrections.csv file: the range of the data may be dominated by outliers, therefore use this file to specify outlier definitions/thresholds

Corrections: general description of outlier definitions (string, optional)

wo_Date: define dates of IceSat data which should be omitted

MinElev: minimum elevation of bathymetry

MaxElev: maximum elevation of bathymetry

# 4 STEP 4: Upload DEMs to GEE

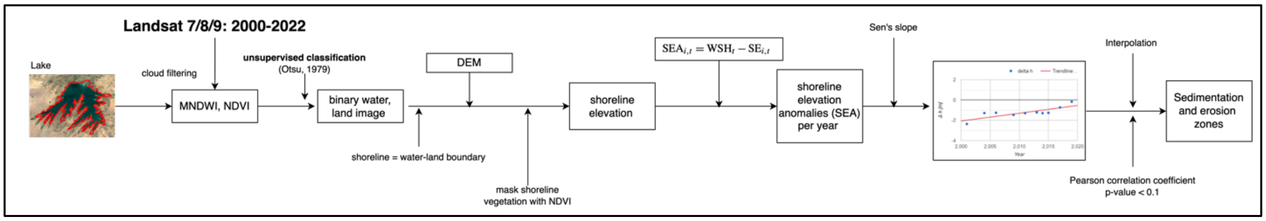
The generated bathymetries must be uploaded to the Google Earth Engine as an asset.

conda create -n gee_up python=3.9

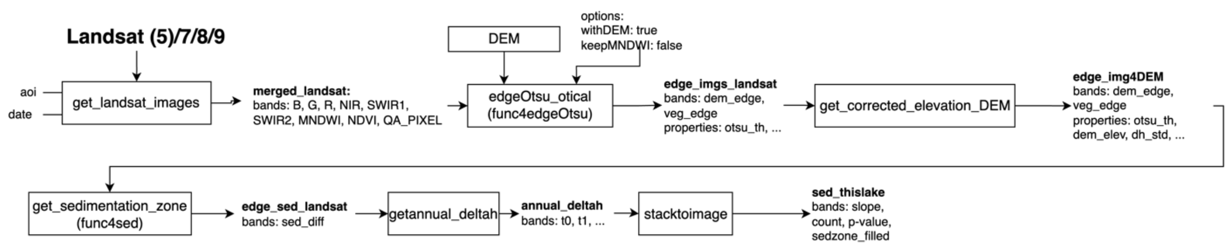
pip install geeup
conda install -c conda-forge earthengine-api=0.1.320
**earthengine authenticate --auth_mode=notebook**

example:
geeup upload --source "/Users/adrian/Library/CloudStorage/OneDrive-ETHZurich/ETH/Master/Masterarbeit/04_Bathymetry/STEP_3/Bathymetries/DEMs/SR_atl08and13_hOrtho_b" --dest "projects/ee-adriankreinerak/assets/DEMs_v2" --metadata "/Users/adrian/Library/CloudStorage/OneDrive-ETHZurich/ETH/Master/Masterarbeit/04_Bathymetry/STEP_3/Bathymetries/DEMs/SR_atl08and13_hOrtho_b/metadata.csv" --user "adriankreiner.ak@gmail.com"

# 5 STEP 5: Sediment Zones:




## 5.1 Description




The script uses a for loop to loop over all defined reservoir and lake IDs.

With the function **get_landsat_images** (see chapter 5.2) an Image Collection named **merged_landsat** of gap filled, cloud and no data filtered Landsat (5)/7/8/9 images within the specified period and area of interest is obtained. The bands cloud, MNDWI, and NDVI are added as well as the properties SENSING_TIME, 'cloud_cover', 'nodata_cover'.

The Image Collection merged_landsat is used as input for the function **func4edgeOtsu**. The function func4edgeOtsu maps over all images in merged_landsat. This function is defined by the function **edgeOtsu_optical** (see chapter 5.3) and the function is called withDEM=true and keepMNDWI= false (default) and creates the Image Collection **edge_imgs_landsat** with the bands

- 'dem_edge', containing the elevation data of the water-land edge and
- 'veg_edge', a binary image if the water-land edge is covered with vegetation

and added the property 'otsu_th' a threshold derived with the unsupervised machine learning algorithm Otsu to distinguish between water and land pixel.

The Image Collection **edge_imgs_landsat** is used as input for the function **get_corrected_elevation_DEM** (see chapter 5.4) and results in the Image Collection **edge_img4DEM**.

The function get_corrected_elevation_DEM returns the **water surface height** calculated by taking the median of the water-land edge (shoreline) excluding pixel with vegetation and adding this as a property named **dem_elev**. It also calculated the standard deviation of the shoreline elevation of each image, stored as a property in **dh_std**. The function is called with the option to exclude outliers where the standard deviation of the shoreline elevation pixel is not with a 5% confidence interval.

The Image collection edge_img4DEM is then passed to the function func4sed (equivalent to **get_sedimentation_zone** (see chapter 5.5) with the output **edge_sed_landsat**. The function computes the band **'sed_diff'** and adds the property **'t'**. The sed_diff band is the difference between every non-vegetation-covered shoreline and the median of these pixels, it is called shoreline elevation anomalies (SEA). The property 't' is the time difference between the input start_year and the time the image was captured.

The Image Collection **edge_sed_landsat** is used as input for the function **getannual_deltah** (see 5.6) resulting in the variable annual_deltah. This is used as input for the function **stacktoimage** (see 5.7). The shoreline elevation anomalies are converted into yearly values and interpolated using the linear regression of Sen's slope (Sen, 1968). Only significant values are considered using **Pearson's correlation coefficient** (p-value). In the final step, spatial interpolation is performed to fill areas without values, resulting in the creation of the **sedzone_filled** band.

## 5.2 get_landsat_images

**Input:**

- area of interest
- start/end date
- options for filtering

**Output:**

- Image Collection of gap filled, cloud and no data filtered Landsat 5/7/8/9 images with added/renamed:
    - Bands:
        - 'cloud',
        - ['B', 'G', 'R', 'NIR','SWIR1','SWIR2', 'cloud','QA_PIXEL']
    - Properties:
        - SENSING_TIME, 'cloud_cover', 'nodata_cover'

**The function works as follows:**

The following data product is loaded for the area of interest and period:

- USGS Landsat 5 Level 2, Collection 2, Tier 1
- USGS Landsat 7 Level 2, Collection 2, Tier 1
- USGS Landsat 8 Level 2, Collection 2, Tier 1
- USGS Landsat 9 Level 2, Collection 2, Tier 1

With the functions **cloudscore_L8_T1L1** (L8, L9) and **cloudscore_L7_T1L2** (L5, L7) (functions like cloudscore_sen chapter 1.2.2) clouds are removed from the different Landsat image collections and the two properties 'cloud_cover', 'nodata_cover' are added. The images are then filtered by these properties according to the defined thresholds cc_tresh for 'cloud_cover' (default: 30) and nodata_tresh (nodata_tresh_l7) for 'nodata_cover' (default: 30, for Landsat 7 default: 50).

With the functions **addbands_l8** (L8, L9) and **addbands_l7** (L5, L7) the **MNDWI**, (L8/L9: using the bands SR_B3 (green) and SR_B6 (short wave infrared), L5/L7: using the bands SR_B2 (green) and SR_B5 (shortwave infrared)) and **NDVI** (L8/L9: using the bands SR_B5 (near infrared) and SR_B4 (red), L5/L7: using the bands SR_B4 (near infrared) and SR_B3 (red)) are calculated. The functions rename the following bands (function similar to addbands_sen, see detailed description: 1.2.1)

- L8/L9:
    - ['SR_B2', 'SR_B3', 'SR_B4', 'SR_B5','SR_B6','SR_B7', 'cloud','QA_PIXEL']
- L5/L7:
    - ['SR_B1', 'SR_B2', 'SR_B3', 'SR_B4','SR_B5','SR_B7', 'cloud','QA_PIXEL']
- Renamed to:
    - ['B', 'G', 'R', 'NIR','SWIR1','SWIR2', 'cloud','QA_PIXEL']

With the function, **slc_gapfilled** gaps are filled in the MNDWI band for the L7, L8 and L9 images. This function takes an image as input and fills any gaps by using a weighted mean, computed from neighboring pixels. The weighting is done using a circle-shaped kernel with a radius of 30m and repeated 8 times. The final picture is combined (blended) with the original picture and the MNDWI band is replaced.

## 5.3 edgeOtsu_optical (func4edgeOtsu)

**Input:**

- **Image**: an optical image: here **merged_landsat**
- **DEM**: a digital elevation model
- **Options**: a canny threshold, canny sigma, minimum and maximum value, scale, radius, vegetation threshold, keepMNDWI, withDEM. Radius: buffer
- **Aoi**_img: an image of the area of interest

**Output:**

- Image with added:
    - Bands: (If withDEM = true)
        - dem_edge: elevation of the water-land edge
        - veg_edge: binary image if edge has vegetation
    - Property:
        - 'otsu_th': threshold to distinguish between water and land
        - If keepMNDWI added properties:
            - area_water: area covered by water
            - inside_aoi: area covered by water inside input aoi

**The function works as follows:**

The function **computeThresholdUsingOtsu** (see chapter 1.2.3) is called with the MNDWI band to obtain a threshold (th) value to distinguish between land and water pixels.

**If withDEM = true** (default: false)
The image dem_edge which contains the elevation of the water-surface edge with buffer at the edges of

+/-10m (default, or defined in radius as input variable) using the threshold th (based on MNDWI) is calculated using the function getEdge10m

The binary image veg_edge is created to identify if there is vegetation at the water-land edges using the NDVI and the threshold for vegetation th_veg (default: 0.5, or defined as input variable).

The image dem_edge and veg_edge are added as bands to the image. The threshold th is added as a property named 'otsu_th'

**If withDEM = false**
Only the property th is added as 'otsu_th'

**If keepMNDWI = true** (default: false)
The total area covered with water (using the obtained threshold by the Otsu method) is added as property as 'area_water'. Also, the area covered by water inside the input variable area of interest is added as 'inside_aoi.

## 5.4 get_corrected_elevation_DEM

**Input:**

- Image Collection
- Optional: Sed_thislake, an optional image containing the slope of the sedimentation zone, can be used to update DEM edges.
- Options:
    - Filter_outliers: has to be specified
    - Pvalues (default: false)
    - Additional_exports (default: false)
- Aoi_img

**Output:**

- Adds properties:
    - dem_elev: elevation of the shoreline
    - dh_std: the standard deviation is an indicator of noise

Filters dem_edge (elevation of the shoreline with a buffer of 10m) and excludes values with vegetation at the water-land edges (veg_edges, NDVI > 0.5) and calculates the median over the water surface height. Also, the standard deviation is calculated as an indicator of noise.

**If sed_thislake = true:**
The DEM edge data will be updated based on the slope data within the sediment zones (updates the DEM edge by masking out pixels with a slope less than 0.01).

**If filter_outliers =true**
The data is filtered based on the standard deviation of the shoreline elevation pixel. A threshold is found by taking the mean of the standard deviation and multiplying it by 1.96, indicating a 5% confidence interval has been selected.

**If additional export = true:**
Also, calculate the water surface area

## 5.5 get_sedimentation_zone (func4sed)

Extracts the shoreline anomalies of individual contours

**Input:**

- image
- start date
- options

**Output**:

- Uses bands dem_edge and veg_edge to replace the band:
    - sed_diff
- Adds the property t: time difference between the input time and the time the image was captured

**Description:**

The function is within a for loop which iterates over all years and therefore the input start_date is increasing by one year every iteration.

The function creates the variable deltat which is the difference between the input start_year and the time the image was captured and adds it as a property.

From every shoreline pixel without vegetation the median elevation is subtracted leading to the sediment difference (sed_diff): this is called **shoreline elevation anomalies (SEA).**

## 5.6 getannual_deltah

The **getannual_deltah** function calculates the annual mean delta height (sediment difference) using edge_sed_landsat height data for the given date range. The output is a stacked image with each band representing each year's annual mean delta height within the specified date range.

**Input:**

- daily_deltah: **Image Collection**
- end_date

**Output:**

- image annual_deltah: stacked image: one image with oneband per year

## 5.7 stacktoimage

Input:

- annual_deltah: image containing bands named "t0", "t1", ..., "t22", which represent shoreline anomalies.

Output:

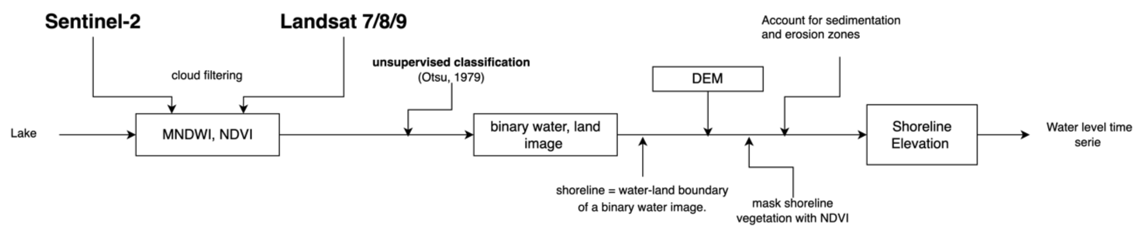
- sed_thislake: bands: slope, count, p-value, sedzone_filled

Calculates the **Pearson's correlation coefficient** between the 't' and 'sed_diff' bands for the images in annual_deltah and stores the result in correlation( pvalue_threshold = 0.1).

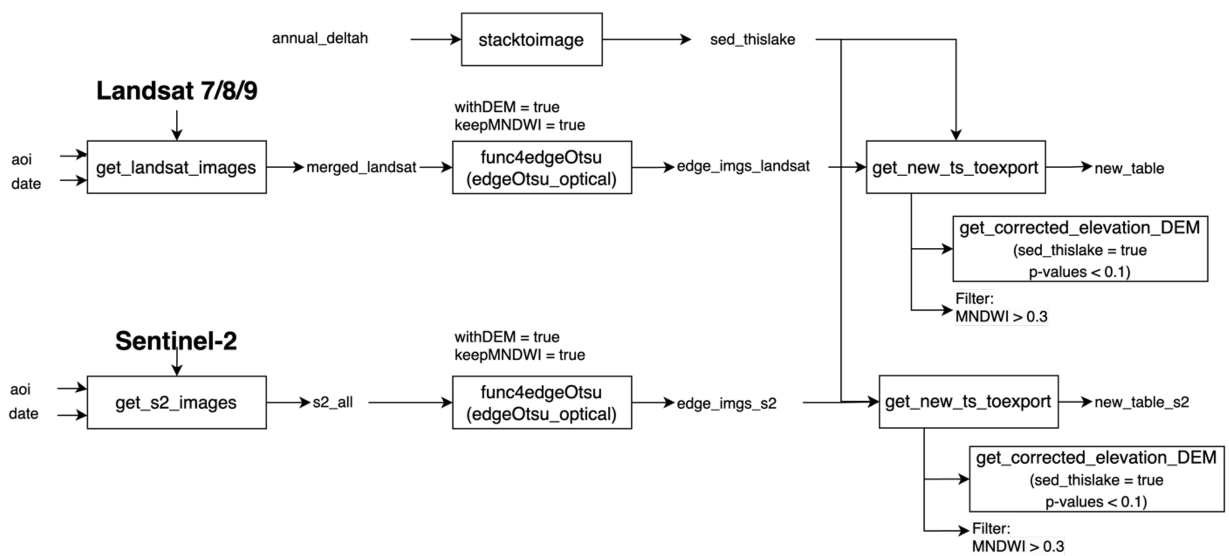
Fits a slope for the 't' and 'sed_diff' bands of the images in annual_deltah using the **Sens slope reducer**. This gives the bands **'slope'**. Adds a band called **'count'** representing the **number of bands in the input image**, and adds the **'p-value'** band from the correlation result. (p-value comes from Pearson's correlation coefficient). The function also checks if the data is unevenly distributed by splitting the annual SEAs into two time periods (from 2000-2010 and 2011-2021).

In summary, this function computes the correlation and Sens slope of shoreline anomalies stored in input image bands and returns an image with additional bands like 'slope', 'count', and 'p-value', along with the properties of the input image. In the final step, spatial interpolation is performed to fill areas without values, resulting in the creation of the **sedzone_filled** band.

# 6 STEP 6: Time series




## 6.1 Description




Using aoi, date and options for filtering Landsat (5)/7/8/9 with the function get_landsat_images (see 5.2) are obtained. With the function, get_s2_images Sentinel-2 images are obtained (the function works similarly to the get_landsat_images function). Using the function func4edgeOtsu (equal to the function esgeOtsu_optical with withDEM = true and keepMNDWI = true, see 5.3) getting the bands:

- 'dem_edge', containing the elevation data of the water-land edge and
- 'veg_edge', a binary image if the water-land edge is covered with vegetation

for the Sentinel-2 and Landsat image collection.