**SEA SEARCH ROUTINES**

1. Sea\_Search\_Routine.py:

Main routine, using the Sea\_Search\_Detection function, invokes the gpt command line, executing a XML file created using defined input parameters.

Modules, toolbox, classes, methods:

* Sea\_Search\_Detection (from Sea\_Search\_Process)

Input Parameters:

* filepath: It is the sentinel original zip file path.
* AOI: Python dictionary structure with the geographic bounding box for a specific area within the original sentinel image.
* AdaptiveThresholding\_Parameters: Python dictionary structure with the window configuration for adaptive threshold detector parameters:
  + targetWindowSizeInMeter: Normally should be about the size of the smallest dimension of a vessel to be detected. (i.e. 40 m)
  + guardWindowSizeInMeter: Normally should be about the size of the the maximum length of a vessel to be detected. (i.e. 400 m)
  + backgroundWindowSizeInMeter: Defines the area within which background statistics are calculated, should be large enough to estimate accurately the local statistics. (i.e. 3xguardWindowSizeInMeter = 1200 m)
  + pfa = A positive number for parameter x. Probability of False Alarm (**PFA** = 10 ^(-x)). (i.e. if x=6, then PFA = 10^(-6) which is 0.000001)
* Object\_Discrimination\_Parameters: Python dictionary structure with the Object Discrimination operator parameters:
  + minTargetSizeInMeter: Target with dimension smaller than this threshold is eliminated. (i.e. 40 m)
  + maxTargetSizeInMeter: Target with dimension larger than this threshold is eliminated. (i.e. 400 m).

Sea Search Detection method:

Return a string (outputfile) with the name of the file created after to the sea search detection process.

* outputfile = Sea\_Search\_Detection(filepath, AOI, AdaptiveThresholding\_Parameters, Object\_Discrimination\_Parameters).

1. Sea\_Search\_Process.py

Includes all the necessary methods to unzip a sentinel file, to create a xml file with all the vessel detection operators, and invoking the xml file using gpt command line.

Modules, toolbox, classes, methods:

* os
* subprocess
* zipfile (from ZipFile): The only additional module installed.

Methods:

* write\_XML\_parameters, write\_XML\_parameters: Writes in the XML file the sources and the generic head for each operator.
* Write\_XML\_FILE: Writes the XML file. Returns a sring with xml file path There are two main dictionaries:
  + Sources\_GrapXML: Contains the initial sources for each operator. The sources are useful to identify the links between the operators and the way that the process is following.
    - (i.e Read ()-> Ellipsoind-Correction\_GG(‘Read’) -> Subset(‘Ellipsoid-Correction-GG’) ->Land-Sea-Mask(‘Subset’)->Calibration(‘Land-Sea-Mask’)->Speckle-Filter(‘Calibration’)->AdaptiveThresholding(‘Speckle-Filter’)->Object-Discrimination(‘AdaptiveThresholding’)->Write(‘Object-Discrimination’)
  + Parameters\_GrapXML: Contains all the input parameters for each operator.
* do\_unzip\_sentinel: Unzips sentinel original file.
* AOI\_TO\_Geom: Converts the boundaries box to a string with snap geometrical format.
* Sea\_Search\_Detection: Using the above functions, invoke the xml file using the gpt command line.

1. Output

 The end file is a raster (Tiff format) with 3 bands, the first two bands are the calibrated sigma value (VH and VV polarization) and the last band is the ship detection result (all the pixel with value equal to 1 are considered as an object detected in the sea).