

# ERDC Satellite Shoreline Mapper (SSM) User Manual Version 1.0

<b>User Manual</b>	<b>1</b>
I. Installation Instructions	1
II. To Create a Shapefile:	6
III. Input Parameters	9
IV. Running the Tool	10
V. Output Files	14
VI. Rerunning a Site (optional)	16
VII. Appendix	16
1. Case Study I:	16
2. Troubleshooting Errors	16
VII. Contact	19

## I. Installation Instructions

-- Place the Satellite\_Shoreline\_Mapper folder on your Desktop

-- CoastSat downloads imagery through Google Earth Engine (GEE) which requires users to register at <https://signup.earthengine.google.com/>. Requesting access to google earth engine can take up to a day so we suggest you do that first. See troubleshooting section for ways to tackle GEE related errors.

- You need to use a gmail account to sign up for Google Earth Engine

-- Install ArcPro

-- Install Anaconda

- The version of Anaconda should not matter. If you already have it on your computer you should be able to use the one you already have
- ACE-IT users: if you don't already have Anaconda download the Anaconda 2019.10 python 3.7 version from the app portal (see troubleshooting for help)
- Other users: <https://www.anaconda.com/download>

-- Open Anaconda prompt and cd into the Satellite\_Shoreline\_Mapper directory (should be on your Desktop). Your prompt should say (base) <path to Satellite\_Shoreline\_Mapper directory>

```
(base) C:\Users\RDCHLSMB>cd .\Desktop\Satellite_Shoreline_Mapper
```

-- Create the CoastSat Environment by typing:

```
conda env create -n coastsat_gui --file environment.yml
```

-- Test the coastsat\_gui environment

```
conda activate coastsat_gui
```

Your terminal will change from starting with (base) to starting with (coastsat\_gui) as shown in the second line below:

```
(base) C:\Users\RDCHLSMB\Desktop\Satellite_Shoreline_Mapper>conda activate coastsat_gui
(coastsat_gui) C:\Users\RDCHLSMB\Desktop\Satellite_Shoreline_Mapper>_
```

-- Once you have access to GEE, open anaconda prompt and type:

```
conda activate coastsat_gui
python
import ee
ee.Authenticate(auth_mode = 'notebook')
```

1. A web browser will open prompting you to log into your google account.
2. Then it will take you to a page that says Notebook Authenticator.
3. Create a cloud project with a name with only lower case letters, numbers and dashes. This name must be unique to proceed but does not impact the rest of the process.
4. Do **not** select the check box that says Use read-only scopes.
5. Then click generate token and you will see a button that allows you to copy the token.
6. Copy and paste that into your Anaconda terminal.
7. Type “`ee.Initialize()`” to ensure Earth Engine is authenticated

-- Setup runCC.bat

1. To edit runCC.bat, open Notepad. Click File>Open and then navigate to the directory or location where you have Satellite\_Shoreline\_Mapper stored. Open runCC.bat. Alternatively, you can right click on runCC.bat and select “Open with > Notepad”

*\*Note: Make sure in the bottom right you have it set to all files and not just .txt files\**

2. Open the Anaconda prompt and find the path to your Anaconda by typing `where anaconda`. It should be similar to ‘`C:\Users\username\Anaconda3`’, however this depends on your Anaconda set up.
  - a. In the example below three paths show up. In this case the path is the first part of each one:  
C:\Anaconda3

```
(coastsat_gui) C:\Users\RDCHLSMB\Desktop\Satellite_Shoreline_Mapper>where conda
C:\Users\RDCHLSMB\Anaconda3\Scripts\conda.exe
C:\Users\RDCHLSMB\Anaconda3\Library\bin\conda.bat
C:\Users\RDCHLSMB\Anaconda3\condabin\conda.bat
```

3. Change the 4 hardcoded paths to your computer (example image below):
  - a. In the orange box change the directory to **[path to your Anaconda]\Scripts\activate.bat**
  - b. In the yellow box change to **[path to your Anaconda]**
  - c. In the blue box change to **[path to your Anaconda]\envs\coastsat\_gui\python.exe**
  - d. In the green box change to the path to your Satellite\_Shoreline\_Mapper folder. Find this path by typing `cd` into your Anaconda Prompt while in the Satellite\_Shoreline\_Mapper folder. This path should be **[path to Satellite\_Shoreline\_Mapper ]\cc\cc\_LZMST.py**

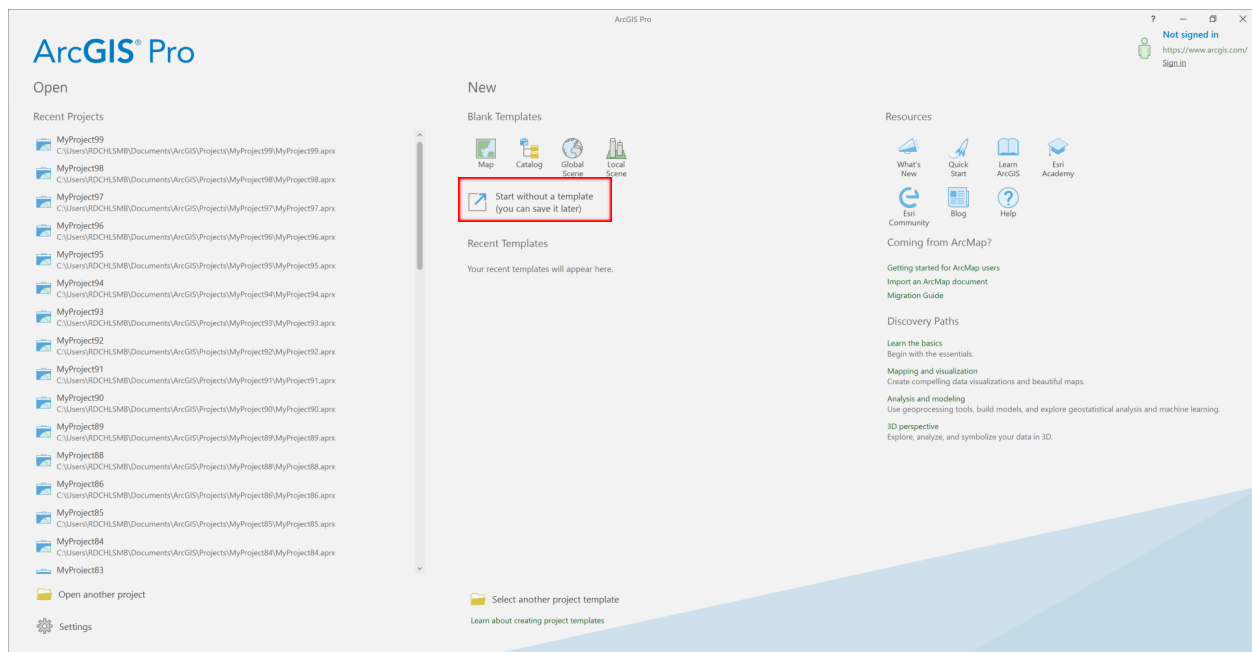
Make sure that the quotation marks are still there from before you started editing paths.

Save the changes with ctrl+s or File>save

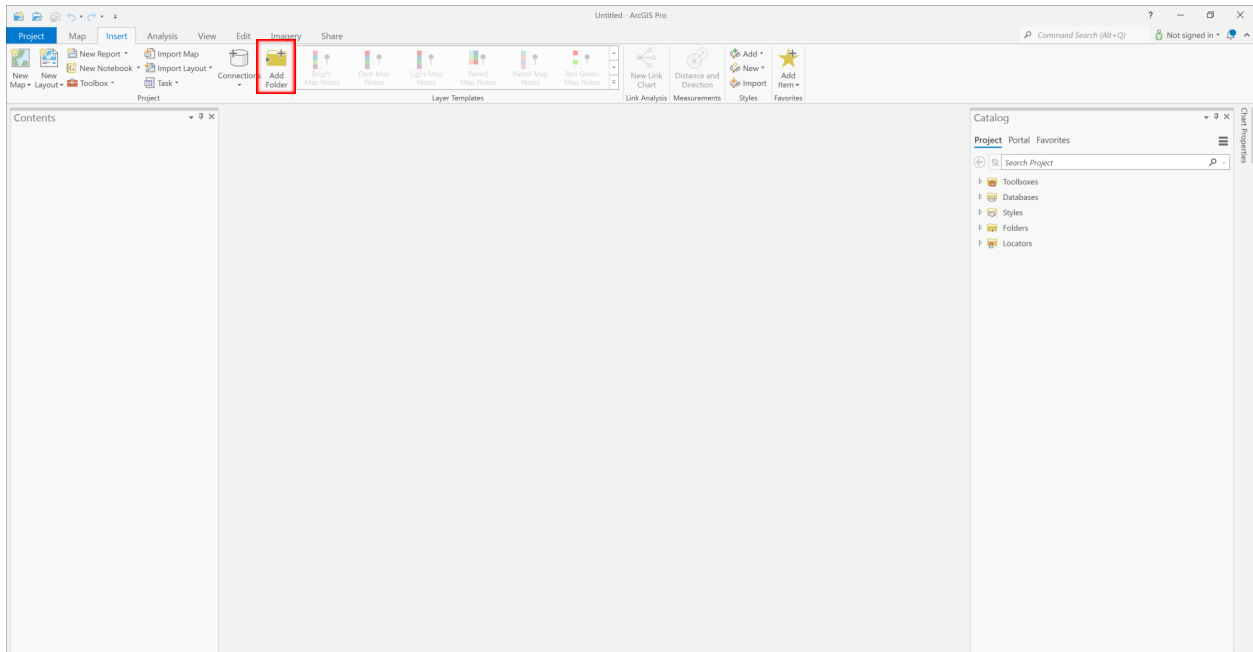
```
1 set PYTHONHOME=
2 set PYTHONPATH=
3 [cmd /C "C:\Users\RDCHLSMB\Anaconda3\Scripts\activate.bat C:\Users\RDCHLSMB\Anaconda3 && call activate coastsat_gui
4 pause
Continued on same line...
&& C:\Users\RDCHLSMB\Anaconda3\envs\csv2_01052023_2\python.exe C:\Users\RDCHLSMB\Desktop\coastsatv2\cc\cc_LZMST.py"
```

Now, you're ready to open the tool in ArcGIS!

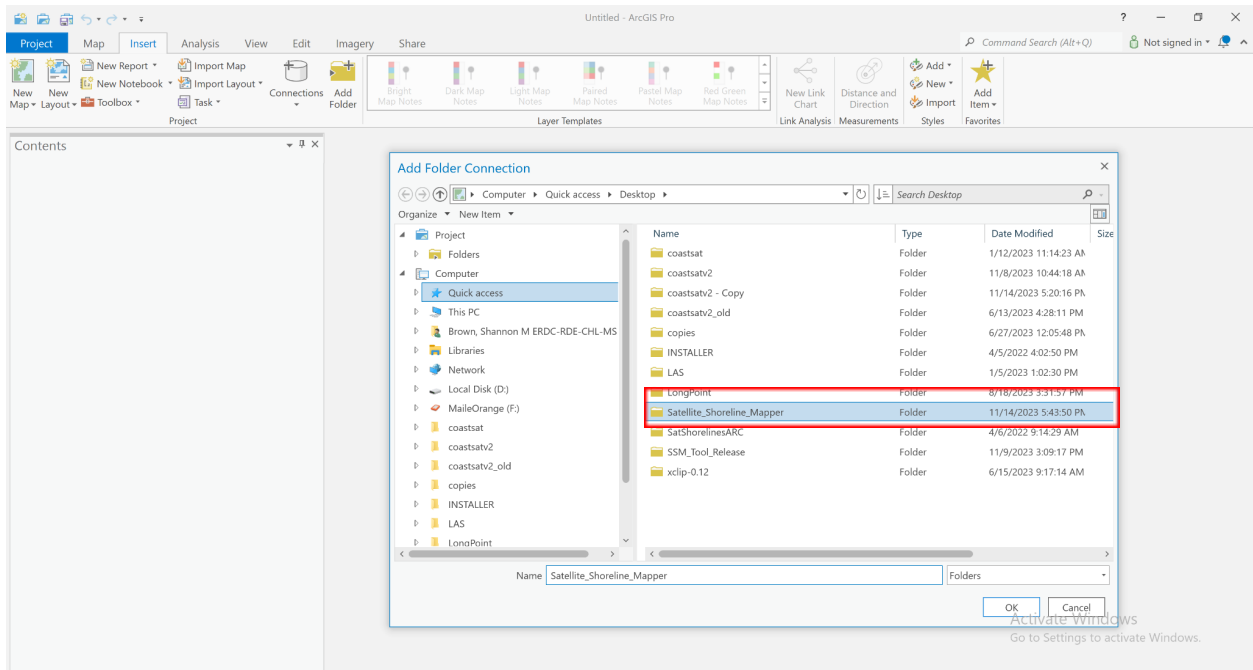
-- Open ArcMap, without a template



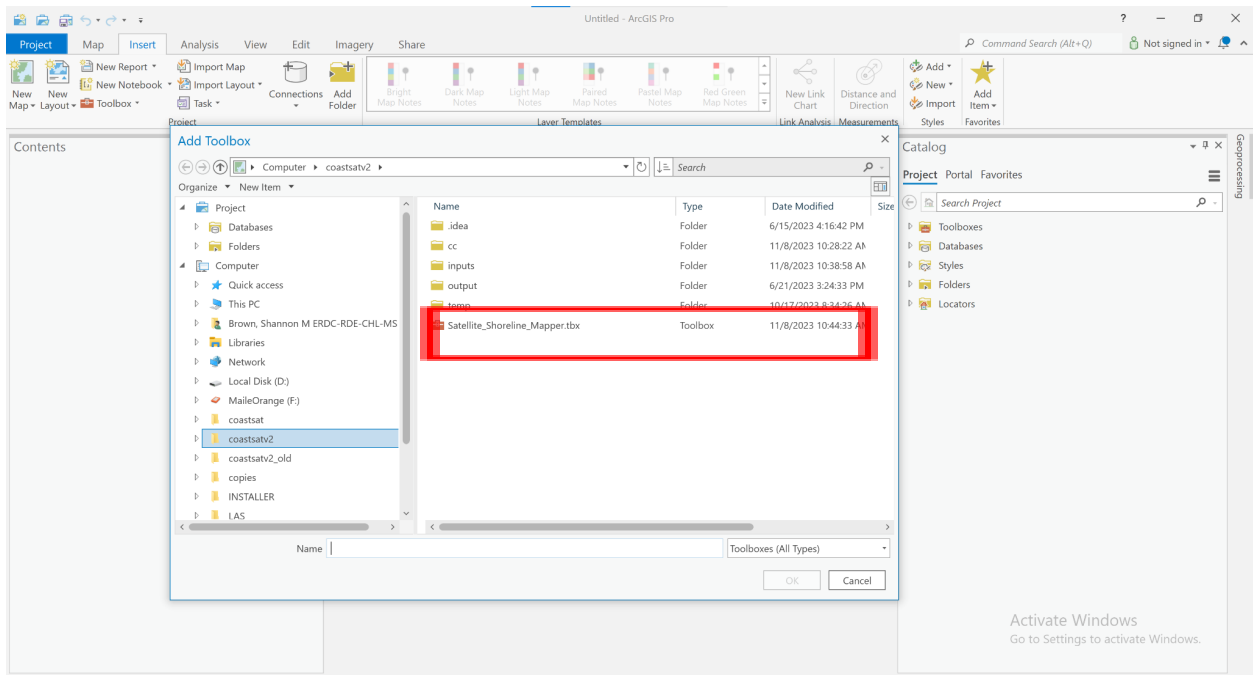
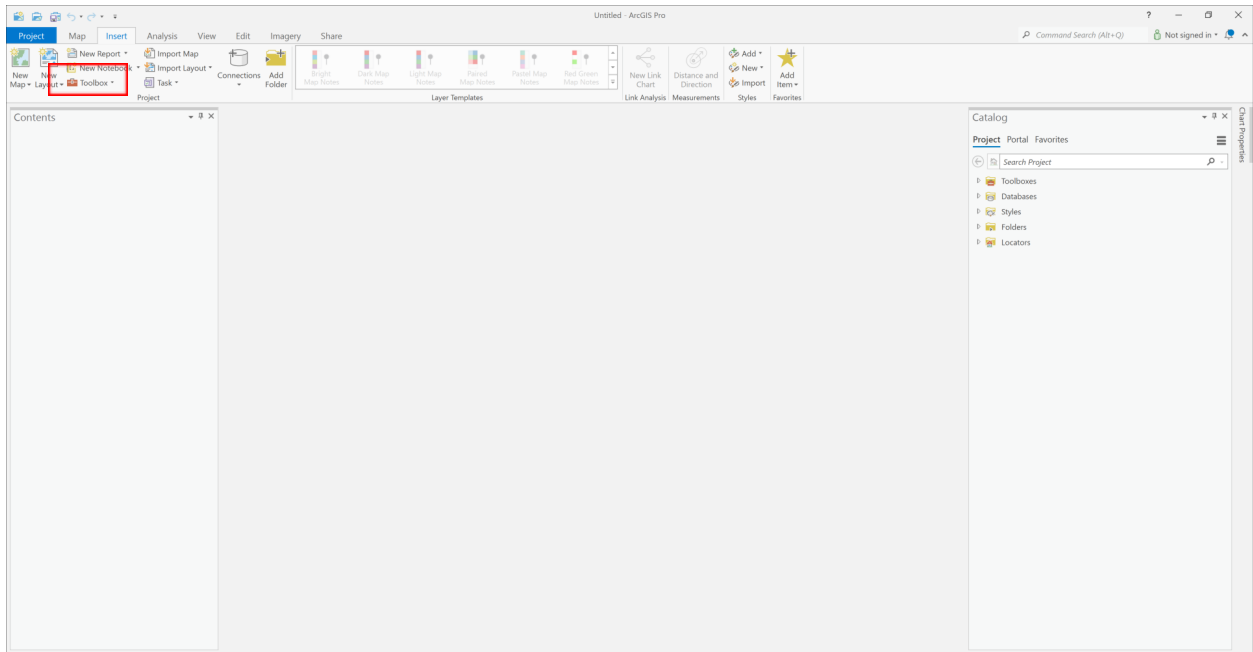
-- Insert Tab, Add folder



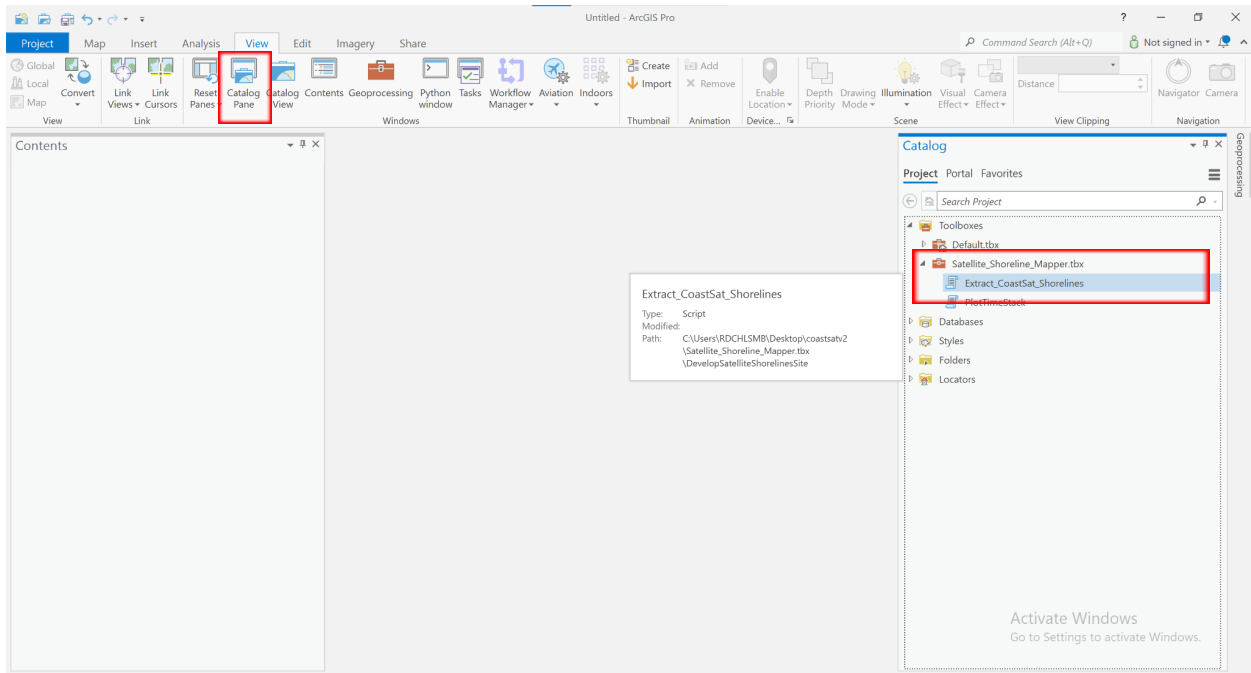
-- Navigate to Satellite\_Shoreline\_Mapper which should be on the Desktop, press "ok"



-- Click on the 'Insert' tab, 'Toolbox', 'Add Toolbox', click "Satellite\_Shoreline\_Mapper.tbx"



-- View catalog pane, "Satellite\_Shoreline\_Mapper.tbx" should be in the toolboxes section

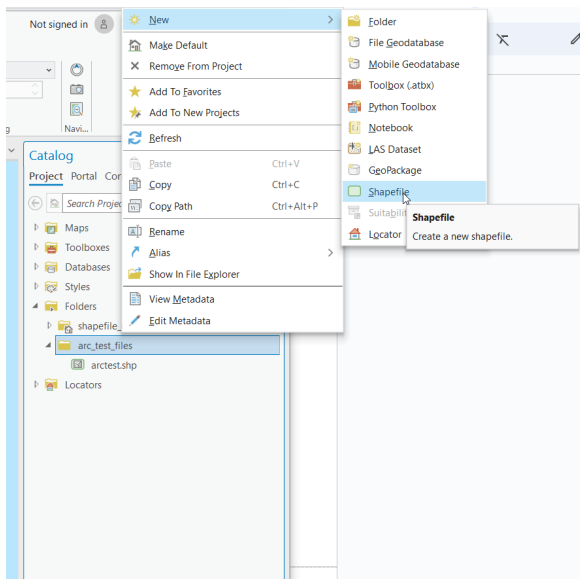


Extract\_CoastSat\_Shorelines is the toolbox. Right click on Extract\_CoastSat\_Shorelines and click open to access the tool

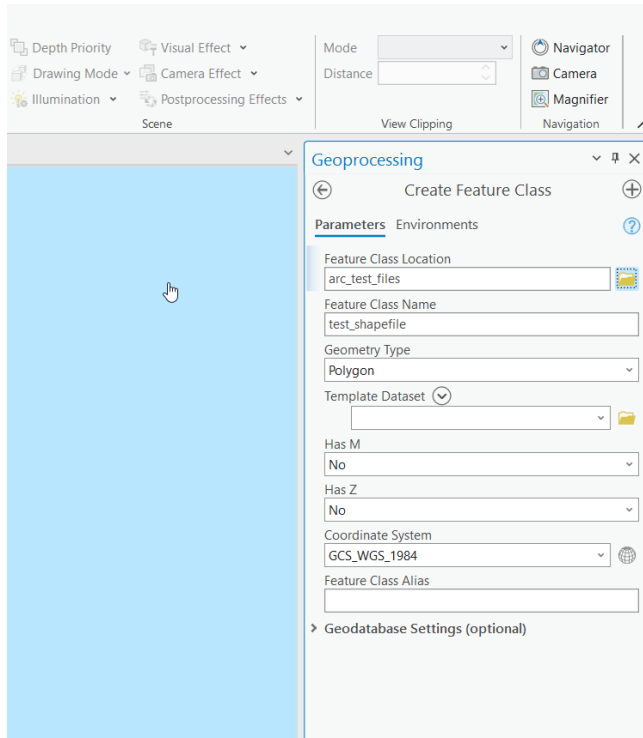
## II. To Create a Shapefile:

-- Select the "View" tab and select "Catalog Pane"

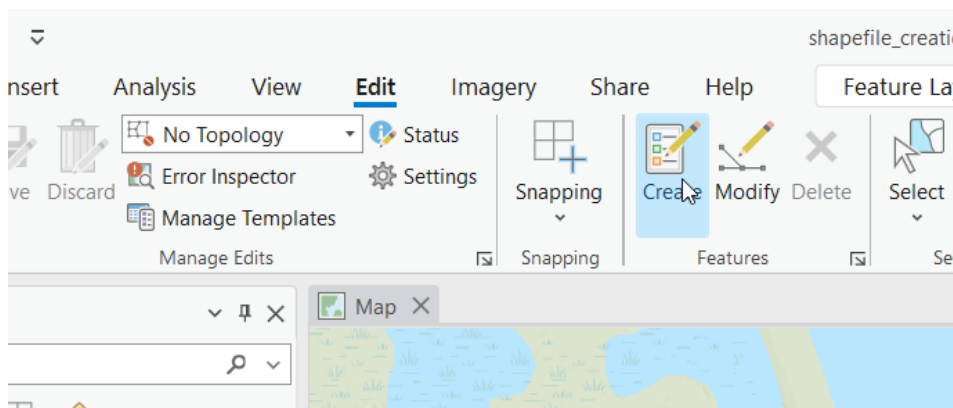
-- Navigate to the folder you wish to add the shapefile to, right click and select "New" and "Shapefile."



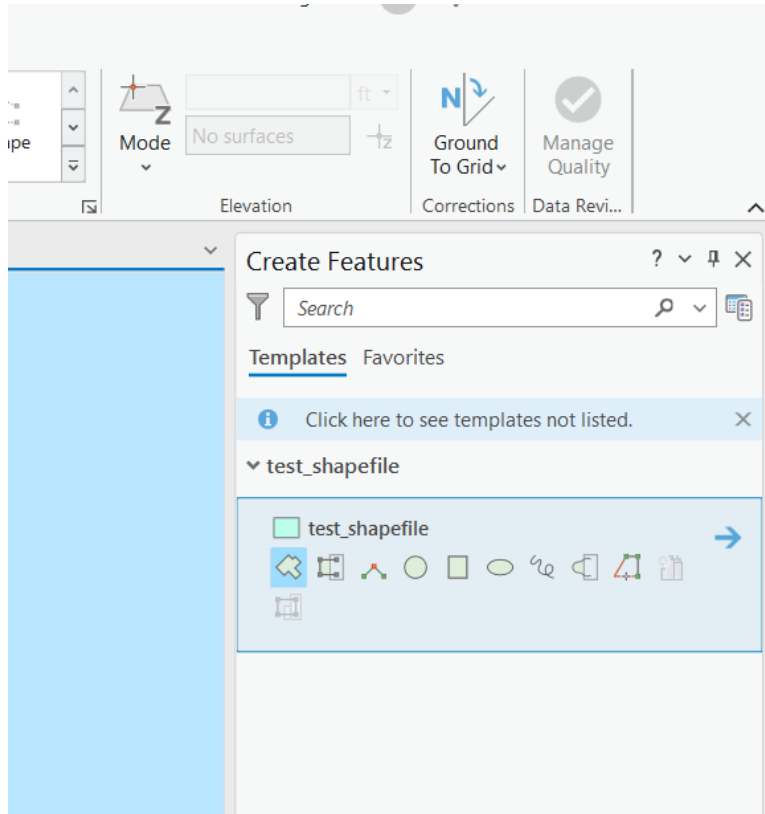
--Once you have entered your info, hit "Run"



--Click the "Edit" tab and select "Create"



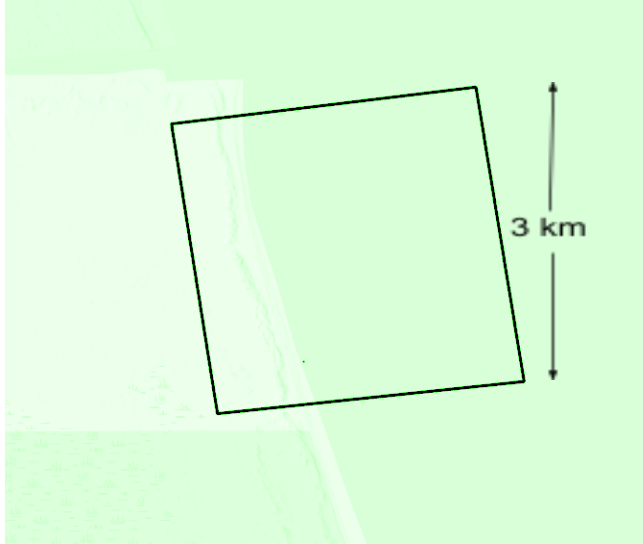
--Under the "Create Features" tab, select your shapefile and the first polygon icon.



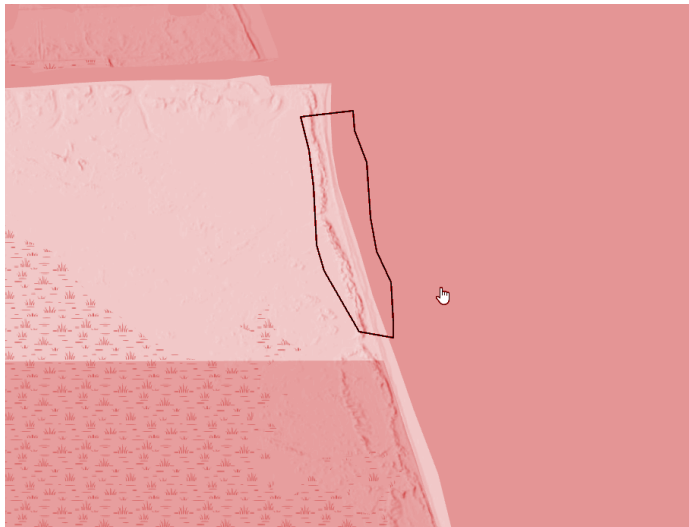
-- Click four corners of your area of interest. Be sure to include a sufficient amount of land and water (see example below). Roughly 50% should be land and 50% water. Double click to complete the polygon.

--Good shapefile example:





Bad shapefile example:



### III. Input Parameters

Is this a repeated run? [boolean]

If you have run this site before, you can check this box to skip the downloading portion of the code which can be resource intensive. In a re-run, you only need to change the sitename parameter.

Shore Polygon Filename? [.shp]

Include the path to a shapefile in which you are interested. Do not include any quotations and use only one '\'. The shapefile must contain only one polygon with exactly 4 points.

Instructions to create a shapefile in section II. **To create a shapefile**

Start Date [yyyy-mm-dd]

Include the first day you would like to download imagery from in a yyyy-mm-dd format. Imagery archives began in 1984.

End Date [yyyy-mm-dd]

Include the last day you would like to download imagery from in a yyyy-mm-dd format.

Contour? [m in ref to msl]

In reference to msl, what elevation contour would you like to track?

Estimated Slope? [int, unitless]

It is encouraged that you include one estimated slope for your study site. This is used to tidally correct the shorelines. If you do not have an estimated slope, keep -999 and it will use CoastSat.Slope. When using CoastSat.slope we recommend a longer dataset (20+ years) so that the tool has ample amounts of imagery. CoastSat.slope is most accurate when the ratio of the tidal range/beach slope is less than 10.

Tidal Gage Number? [int]

A list of gages can be found in the output folder inside Satellite\_Shoreline Mapper. Please type in the line number of the tide gage number you need minus 1. If you do not know the gage number and keep the default value of -1, it will find one for you.

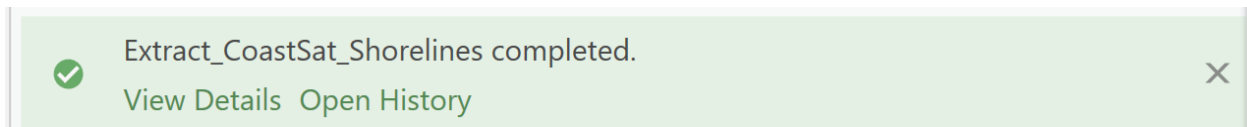
Transect spacing [m]

What distance spacing would you like your cross-shore transects to be generated?

## IV. Running the Tool

After you fill the parameters described in section III, hit run!

In the bottom of the geoprocessing panel, the screen will display a progress bar reading "Extract\_CoastSat\_Shorelines Running.... Shortly after, it will change to a green box with the writing "Extract\_CoastSat\_Shorelines Complete".



This means the tool was successful in setting up the python environment, but it has not run the Coastsat yet. Coincidentally, a command prompt will open (below) and start running CoastSat.

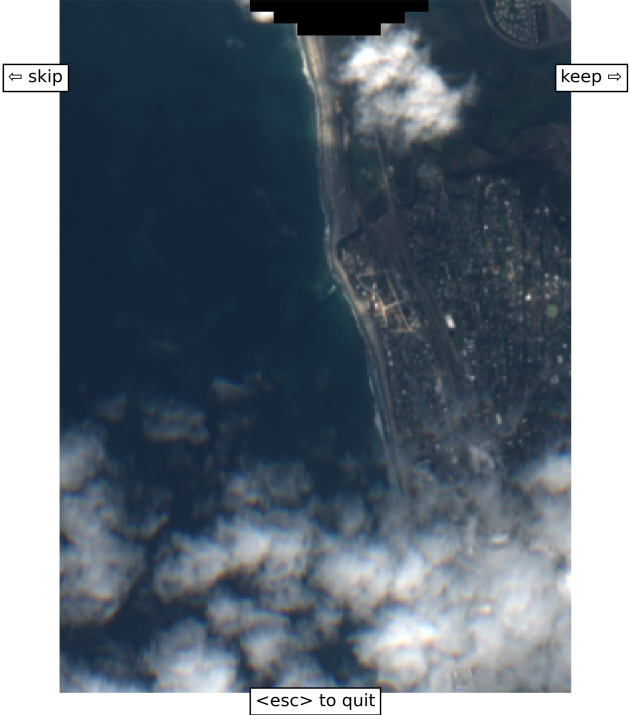
```
C:\WINDOWS\system32\cmd.exe
C:\Users\RDCHLSMB\Desktop\coastsatv2>cmd /C "C:\Users\RDCHLSMB\Anaconda3\Scripts\activate.bat C:\Users\RDCHLSMB\Anaconda3
&& call activate csv2_01052023_2 && C:\Users\RDCHLSMB\Anaconda3\envs\csv2_01052023_2\python.exe C:\Users\RDCHLSMB\Desкто
p\coastsatv2\cc\cc_LZMST.py"
Number of images available between 2021-10-01 and 2022-01-01:
- In Landsat Tier 1 & Sentinel-2 Level-1C:
  L5: 0 images
  L8: 6 images
  S2: 32 images
  Total to download: 38 images
- In Landsat Tier 2 (not suitable for time-series analysis):
  L5: 0 images
  L8: 13 images
  Total Tier 2: 13 images
Number of images available between 2021-10-01 and 2022-01-01:
- In Landsat Tier 1 & Sentinel-2 Level-1C:
  L5: 0 images
  L8: 6 images
  S2: 32 images
  Total to download: 38 images
- In Landsat Tier 2 (not suitable for time-series analysis):
  L5: 0 images
  L8: 13 images
  Total Tier 2: 13 images

Downloading images:
L5: 0 images
L8: 6 images
100%
```

It will download all available imagery, then ask the user to choose a good satellite image. The left arrow on your keyboard will signify it is a bad image and the right arrow will make that satellite image the reference shoreline image.

Example of a bad reference image:

Press <right arrow> if image is clear enough to digitize the shoreline.  
If the image is cloudy press <left arrow> to get another image



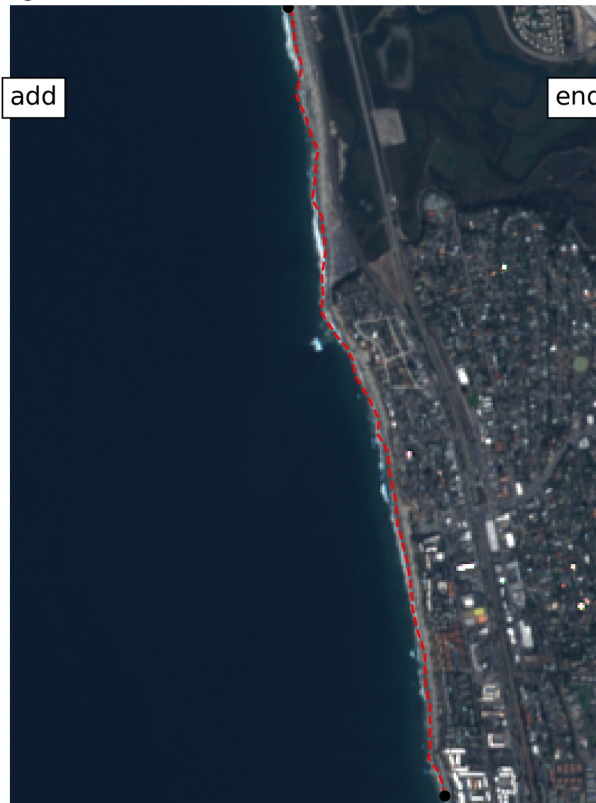
Example of a good reference image:

Press <right arrow> if image is clear enough to digitize the shoreline.  
If the image is cloudy press <left arrow> to get another image

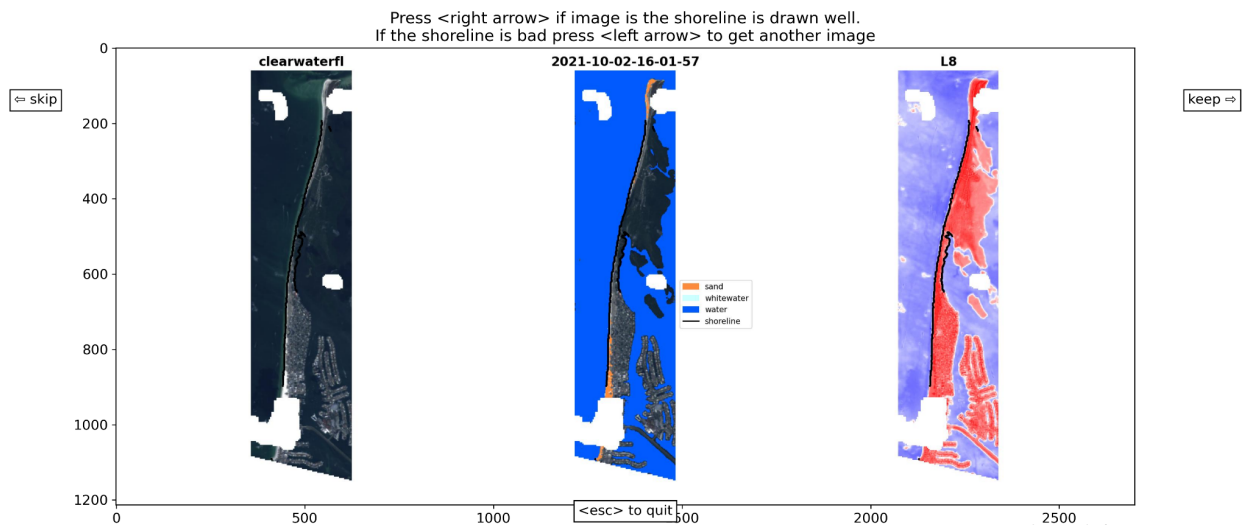


Hit the “keep” button on the screen or press the right arrow on your keyboard. It will then ask you to draw a shoreline on the image. It does not matter what end (N, S, E, W) you start on, but click points following the shoreline in the image. See examples below of good reference shorelines.

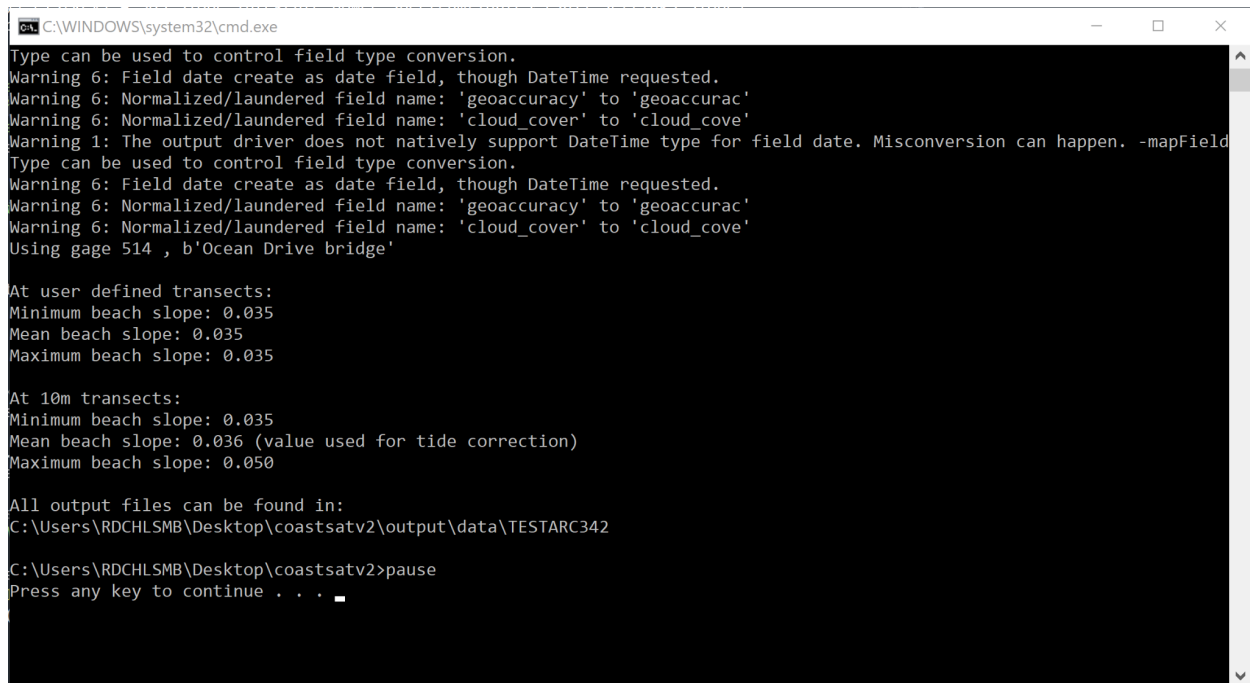
click on <add> to digitize another shoreline or on <end> to finish and save the shoreline(s)



It will then extract shorelines from all downloaded imagery. This portion may take about 30 minutes depending on how long your dataset is. After that portion is done, a figure will pop up and ask you to use the right arrow to indicate it is a 'good' or the left arrow key to indicate a 'bad' shoreline similar to the reference shoreline choice. After 5 'good' shorelines, the model will start to label shorelines 'good' or 'bad' by comparing the current shoreline to the 'good' shorelines. If the current shoreline is within 75% similarity to the 'good' shorelines, it will keep it.



Then, a set of cross-shore transects at 10 meter spacing will be generated as well as a set of transects at the spacing you requested. The shorelines will be tidally corrected using the water level recorded by the user-input or tool retrieved tidal station at the time of image acquisition and the user or coastsat generated average beach slope at every 10 meter transect. Once it is finished, it will tell the user where the output files are below.



```
C:\WINDOWS\system32\cmd.exe
Type can be used to control field type conversion.
Warning 6: Field date create as date field, though DateTime requested.
Warning 6: Normalized/launched field name: 'geoaccuracy' to 'geoaccurac'
Warning 6: Normalized/launched field name: 'cloud_cover' to 'cloud_cove'
Warning 1: The output driver does not natively support DateTime type for field date. Misconversion can happen. -mapField
Type can be used to control field type conversion.
Warning 6: Field date create as date field, though DateTime requested.
Warning 6: Normalized/launched field name: 'geoaccuracy' to 'geoaccurac'
Warning 6: Normalized/launched field name: 'cloud_cover' to 'cloud_cove'
Using gage 514 , b'Ocean Drive bridge'

At user defined transects:
Minimum beach slope: 0.035
Mean beach slope: 0.035
Maximum beach slope: 0.035

At 10m transects:
Minimum beach slope: 0.035
Mean beach slope: 0.036 (value used for tide correction)
Maximum beach slope: 0.050

All output files can be found in:
C:\Users\RDCHLSMB\Desktop\coastsatv2\output\data\TESTARC342

C:\Users\RDCHLSMB\Desktop\coastsatv2>pause
Press any key to continue . . .
```

## V. Output Files

All of the output files are saved in the site directory located within the Satellite\_Shoreline\_Mapper/output/data folder structure.

There should be four folders labeled **jpg\_files**, **L5**, **L8**, **S2**, and **Timeseries Plots**. The L5, L8, and S2 folders contain the raw satellite imagery downloaded through Google Earth Engine along with the associated metadata files. The post-processed, classified imagery is saved inside jpg\_files/detection. The bad folder holds the imagery the classification was bad and produced an unrepresentative shoreline. The Timeseries Plots will have a figure showing the user-indicated spacing transects colored by the linear regression rates along your whole site and a figure for each user-indicated spacing transect showing the cross-shore evolution through time. That figure will have every shoreline position in blue, a moving average shown in orange, and a linear regression rate shown in red.

**metadata** (pkl) - contains a dictionary for each satellite (L5, L8, S2). Inside the dictionary are 4 lists of certain attributes for each of the satellite images downloaded, the georeference accuracy of the image (acc\_georef), the date, the EPSG, and the filename.

**sitename\_output** (pkl, shp, geojson) - this contains the raw (not QAQCd nor tidally corrected) shorelines extracted using coastsat. In the pickle file, the coordinates are saved in the 'shorelines' element. Also saved in the pickle file is the percent cloud cover in the image, the date, the filename, the geoaccuracy,

the classification (sand, water, whitewater, other land) of each pixel in the image (imClassifs), the MNDWI threshold for the image, and the satellite that took the image.

**sitename\_outputTidalC** (geojson) - this contains the final (QAQCd and tidally corrected) shoreline points as a geojson. In addition to the coordinates of the shoreline, the file contains the date of each of the shorelines, the satellite name that captured the image, the geoaccuracy of that image, and the percent cloud cover. If this file contains many 'bad' shorelines, see section V.

**sitename\_outputTidalC\_lines** (shp) - this contains the final (QAQCd and tidally corrected) shorelines as a shapefile. In addition to the coordinates of the shoreline, the file contains the date of each of the shorelines, the satellite name that captured the image, the geoaccuracy of that image, and the percent cloud cover. If this file contains many 'bad' shorelines, see section V.

*\*An important note: because of the way the tool calculates the tidal correction, the tool removes points for quality control out of the shorelines generated, thus the lines shapefile appears to smooth over some of the shoreline. These tidally corrected shorelines might appear smoother than the corresponding non tidally corrected shoreline for that date. They are meant to be representative and show a general trend. A user can plot them as points if they wish using the geojson or in python using the points in the shapefile*

**sitename\_outputTrim** (shp) - this contains the QAQCd (not tidally corrected) shorelines extracted using CoastSat. In addition to the coordinates of the shoreline, the file contains the date of each of the shorelines, the satellite name that captured the image, the geoaccuracy of that image, and the percent cloud cover.

**sitename\_reference\_shoreline** (geojson, shp) - these are the coordinates of the reference shoreline the user drew while running the tool.

**cross\_distance** (pkl) - for every transect generated, there is an array of distances. This distance is the distance from the landward endpoint of the transect and where the shoreline intersects that cross-shore transect.

**cross\_distance\_tidally\_corrected** (pkl) - for every transect generated, there is an array of distances. This distance is the distance from the landward endpoint of the transect and where the tidally corrected shoreline intersects that cross-shore transect.

**transect\_time\_series** (csv) - cross\_distance.pkl repackaged as a csv file where columns are transects and rows are the times of each shoreline with its distance to the landward endpoint of the transect. Empty cells are distances that were removed in QAQC

**transect\_time\_series\_tidally\_corrected** (csv) - cross\_distance\_tidally\_corrected.pkl repackaged as a csv file where columns are transects and rows are the times of each shoreline with its distance to the landward endpoint of the transect. Also includes column of the tide at the time of each shoreline. Empty cells are distances that were removed in QAQC

**slope\_ests** (pkl,txt) - a record of the satellite derived beach slope estimate at each transect from transectSmall. The average of these beach slopes is used for the tide correction.

**TideStationSelection** (png) - figure illustrating the distance from the user's site and the nearest tide gauges. A low resolution shoreline is illustrated in black dots. The red dot is the center point of the study

site, the cyan dots are the surrounding tide stations (labeled with the number associated in the Satellite\_Shoreline\_Mapper/output/gages.txt file)

**transects** (pkl, shp, geojson) - the transects created with the tool at the 10 m spacing

**transectsFull** (pkl) - the transects created with the tool at 10 m spacing

**transectsroots** (shp) - all the points that make up the landward extent of each transect in transectFull

**transectsRotations** (pkl) - rotation of transects that are used to create transects perpendicular to shore, instances of Rotation (scipy.spatial.transform.Rotation)

**transectsSmall** (shp, geojson) - the transects created with the tool at the user defined spacing

## VI. Rerunning a Site (optional)

This last step is for reducing the amount of 'bad' or noisy shorelines in the output files. If the sitename\_outputTidalC\_lines.shp file created has too many 'bad' shorelines, the model is not able to weed out enough of the bad shorelines at this site. In this scenario, the user would go into the coastsatv2\output\data\sitename\jpg\_files\detection directory and manually move bad shoreline images into the '\bad' directory. Then, go to run the tool with the same parameters, but check the box next to 'is this a rerun?'. This will indicate to the tool that all the remaining shorelines in the detection directory are 'good' shorelines and it will perform the tidal correction on only those shorelines.

## VII. Appendix

### 1. Troubleshooting Errors

#### 1.1. Gcloud missing:

Follow the instruction here: <https://cloud.google.com/sdk/docs/install>

#### 1.2. GEE Bad Request:

---

```
conda activate coastsat_gui  
earthengine authenticate --auth_mode=notebook
```

---

A web-browser will pop up and ask you to sign into your gmail account. Then it will ask for access to your account and the cloud, allow for both. You will then receive a new authentication token, copy and paste this into the anaconda terminal when it asks for the new authentication token.

if you don't see 'enter verification code:', rerun the 'earthengine authenticate --auth\_mode=notebook'



command and go through the process again. For more information, go to the official website below:

<https://developers.google.com/earth-engine/guides/access>

### 1.3.Connection Error:

```
File "C:\Users\RDCHLSMB\Anaconda3\envs\csv2_01052023_2\lib\site-packages\requests\adapters.py", line 547, in send
    raise ConnectionError(err, request=request)
requests.exceptions.ConnectionError: ('Connection aborted.', TimeoutError(10060, 'A connection attempt failed because the
connected party did not properly respond after a period of time, or established connection failed because connected host
has failed to respond', None, 10060, None))
```

This is a common error during the image downloading process of CoastSat. Delete the site directory that was created by the tool (Satellite\_Shoreline\_Mapper\output\sitename) and rerun with the same parameters. The connection broke or timed out.

### 1.4.Downloading Anaconda from the App Portal

\*\* This tool should work with most versions of Anaconda. If you already have one installed, you can use that version. \*\*

ACE-IT Users: Some versions of Anaconda on the USACE App Portal don't install correctly. As of April 2023 the working version was 2019.10. However, the App Portal updates often so that version may no longer work. If you are having issues downloading Anaconda (Anaconda prompt doesn't appear, anaconda navigator does not open correctly, etc.) try:

1. On the app portal uninstall Anaconda and install a different version
2. Reach out the ACE-IT for support

Other users: make sure you are downloading the the correct version for your computer off of <https://www.anaconda.com/download>

### 2.5 Warnings shown

```
Warning 1: The output driver does not natively support DateTime type for field date. Misconversion can happen. -mapField
Type can be used to control field type conversion.
Warning 6: Field date create as date field, though DateTime requested.
Warning 6: Normalized/laundered field name: 'geoaccuracy' to 'geoaccurac'
Warning 6: Normalized/laundered field name: 'cloud_cover' to 'cloud_cove'
```

These warnings will pop up on every run. They are expected and can be ignored. They do not affect the accuracy of the process

### 2.6 ShapeFileException (or other file exception errors)

```

Traceback (most recent call last):
  File "C:\Users\u4hfakrd\Desktop\coastsatv2\cc\cc_LZMST.py", line 53, in <module>
    sf = shapefile.Reader(polygon)
  File "C:\Anaconda3\envs\csv2_test\lib\site-packages\shapefile.py", line 1048, in __init__
    self.load(path)
  File "C:\Anaconda3\envs\csv2_test\lib\site-packages\shapefile.py", line 1193, in load
    raise ShapefileException("Unable to open %s.dbf or %s.shp." % (shapeName, shapeName))
shapefile.ShapefileException: Unable to open C:\Users\u4hfakrd\Desktop\coastsatv2\inputs\testarc.dbf or C:\Users\u4hfakrd\Desktop\coastsatv2\inputs\testarc.shp.

```

If you get an error (example shown above) that says “Unable to open...” you might have one of the files from a previous run open (either in arcGIS, Excel, or another application). This will result in the code not being able to access it. Close the indicated file and restart the run.

## 2.7 FileNotFoundError

```

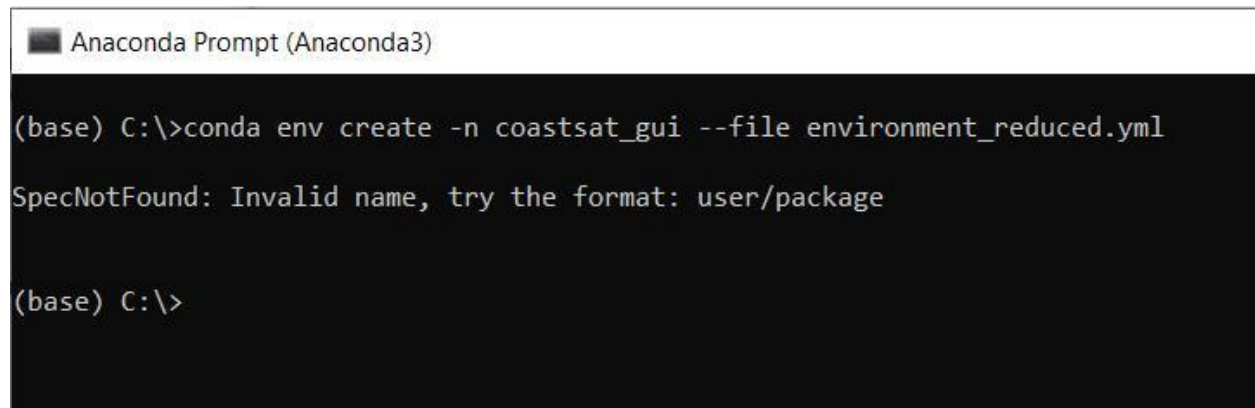
Traceback (most recent call last):
  File "C:\Users\u4hfakrd\Desktop\coastsatv2\cc\cc_LZMST.py", line 187, in <module>
    with open(filepath+"\\metadata0.pkl", 'rb') as f:
FileNotFoundError: [Errno 2] No such file or directory: 'C:\\Users\\u4hfakrd\\Desktop\\coastsatv2\\output\\data\\TESTARC2\\metadata0.pkl'

```

If you get an error (example shown above) that the file or directory does not exist when you are trying to run a rerun:

1. It is possible that the first run failed for some reason and the needed files to rerun do not exist. Run the tool again, unchecking the rerun box
2. Check your spelling of sitename. In the above example the sitename should have been TESTARC, but was written as TESTATC.

## 2.8 SpecNotFound



```

Anaconda Prompt (Anaconda3)

(base) C:\>conda env create -n coastsat_gui --file environment_reduced.yml

SpecNotFound: Invalid name, try the format: user/package

(base) C:\>

```

Make sure you are in the Satellite\_Shoreline\_Mapper directory

## 2.9 shapely.errors.GEOSEException: IllegalArgumentExpection

```

At 10m transects:
Minimum beach slope: 0.035
Mean beach slope: 0.120 (value used for tide correction)
Maximum beach slope: 0.200

Traceback (most recent call last):
  File "C:\Users\u4hfakrd\Desktop\coastsatv2\cc\cc_LZMST.py", line 536, in <module>
    gdf = SDS_tools.output_to_gdf(output, geomtype)
  File "C:\Users\u4hfakrd\Desktop\coastsatv2\cc\coastsat\SDS_tools.py", line 691, in output_to_gdf
    geom = geometry.LineString(output['shorelines'][i])
  File "C:\Anaconda3\envs\coastsat_gui\lib\site-packages\shapely\geometry\linestring.py", line 73, in __new__
    geom = shapely.linestrings(coordinates)
  File "C:\Anaconda3\envs\coastsat_gui\lib\site-packages\shapely\decorators.py", line 77, in wrapped
    return func(*args, **kwargs)
  File "C:\Anaconda3\envs\coastsat_gui\lib\site-packages\shapely\creation.py", line 120, in linestrings
    return lib.linestrings(coords, out=out, **kwargs)
shapely.errors.GEOSException: IllegalArgumentException: point array must contain 0 or >1 elements

C:\Users\u4hfakrd\Desktop\coastsatv2>pause
Press any key to continue . . .

```

This error indicates that there are too many “bad” shorelines that made it through the classifier. Please see section VI. Rerunning a site (optional) on page 16.

2.10 If images fail to download completely, a re-run will not work:

```

FileNotFoundError: [Errno 2] No such file or directory:
'C:\\Users\\rfuser\\Desktop\\Satellite_Shoreline_Mapper\\output\\data\\Duck_15yr\\metadataO.pkl'

```

## VII. Contact

If you have questions/problems that are not covered in this manual please reach out to our development team:

**Ian Conery**

[Ian.W.Conery@usace.army.mil](mailto:Ian.W.Conery@usace.army.mil)

**Shannon Brown**

[Shannon.M.Brown@usace.army.mil](mailto:Shannon.M.Brown@usace.army.mil)

**Katherine DeVore**

[Katherine.R.DeVore@usace.army.mil](mailto:Katherine.R.DeVore@usace.army.mil)