Casper Bakker
casper.bakker@rws.nl
c.f.bakker1@students.uu.nl

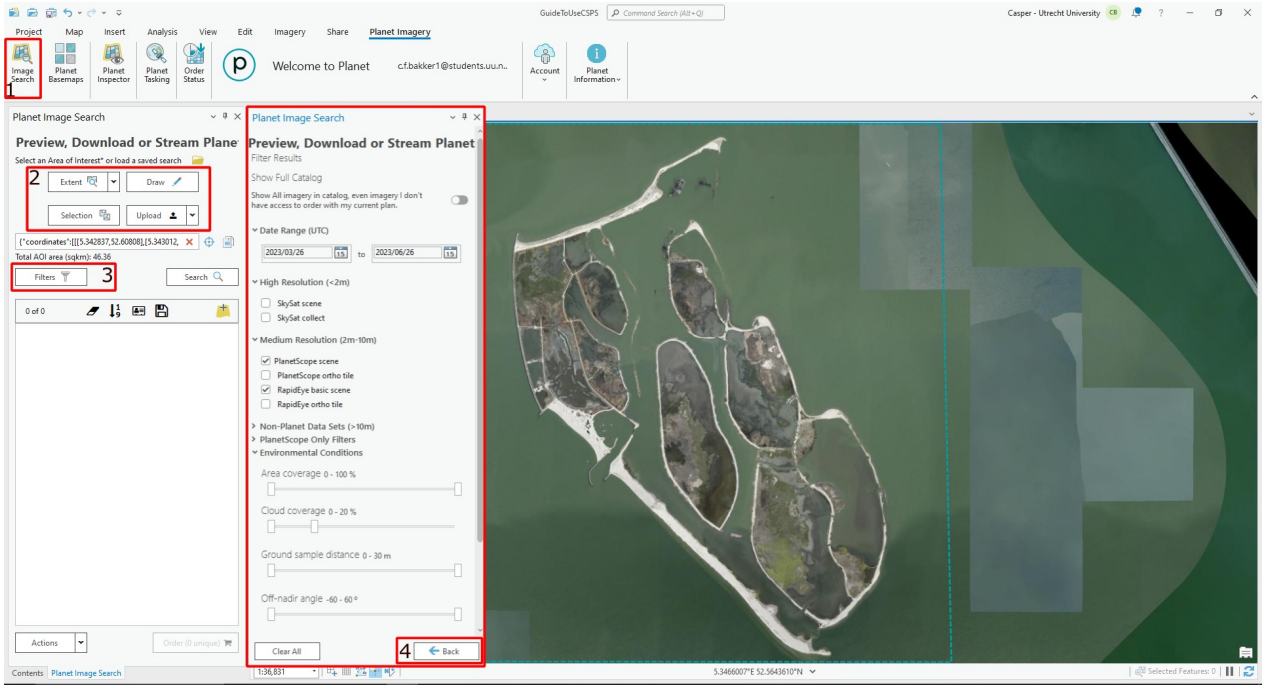
Guide
For the application of CoastSat.PlanetScope

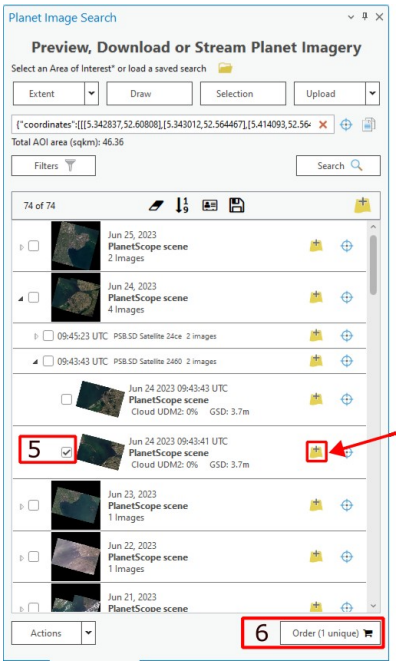
26/06/2023

Step 1:

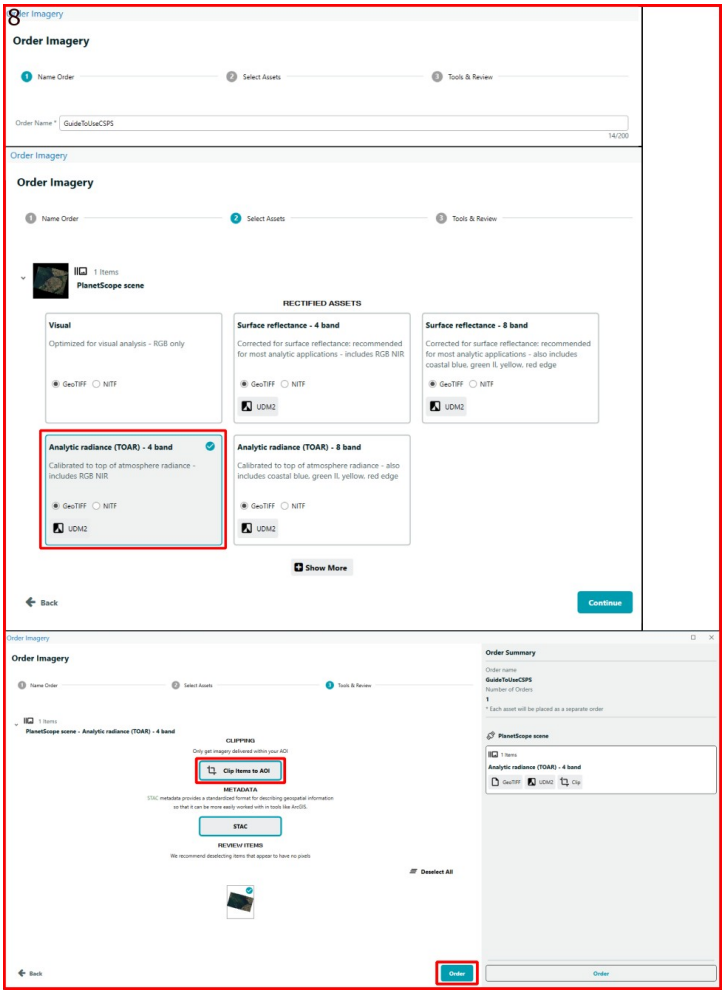
Access PlanetScope imagery by requesting an account. Students can request a free account via this link. Once you have an account, you can request images via QGIS, ArcGIS Pro or Google Earth Engine. More information can be found at this site. Below you will find a step-by-step guide for using ArcGIS Pro, for which the plugin for ArcGIS Pro must be installed.

1. Select the image search window.
2. Select the area of interest, which can be done by drawing an area or uploading a .json file.
3. Select the filters.
4. Choose a medium-resolution PlanetScope scene and a RapidEye basic scene (At the moment, high-resolution images are not available in the Netherlands). Make sure the cloud cover is less than 20%, although the location of the clouds may vary individually. The most important thing is that the beach is fully visible.
5. Confirm the settings.
6. Choose the satellite image, add the images to the map, and inspect them visually.
7. After selecting all the images, the order can be placed.
8. Confirm the order, specifying the image format as 'Analytic radiance (TOAR) - 4band GeoTIFF,' and choose 'clip to AOI.'
9. The order can be downloaded as a zip file, containing the .tiff files.

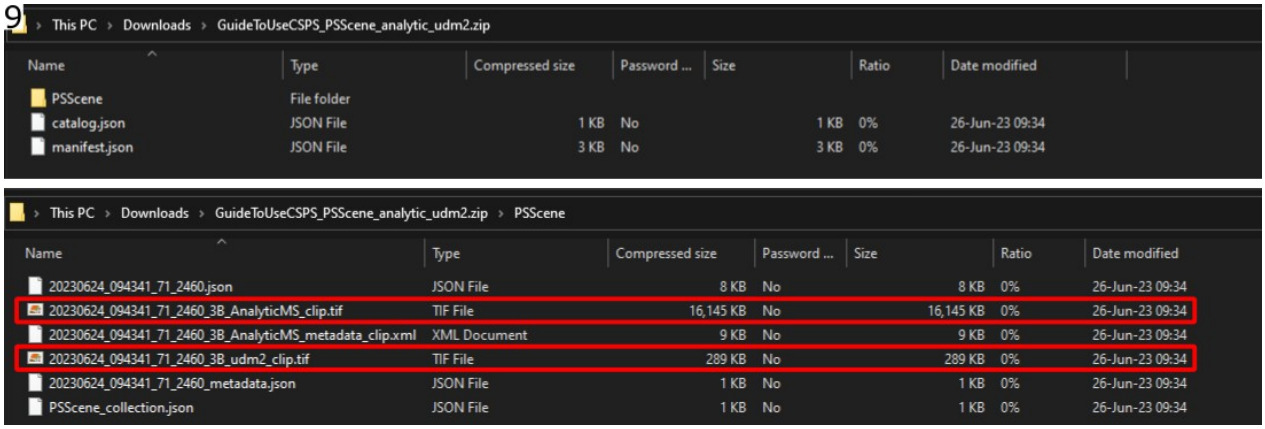




Bekijk de satellietbeelden van tevoren



9



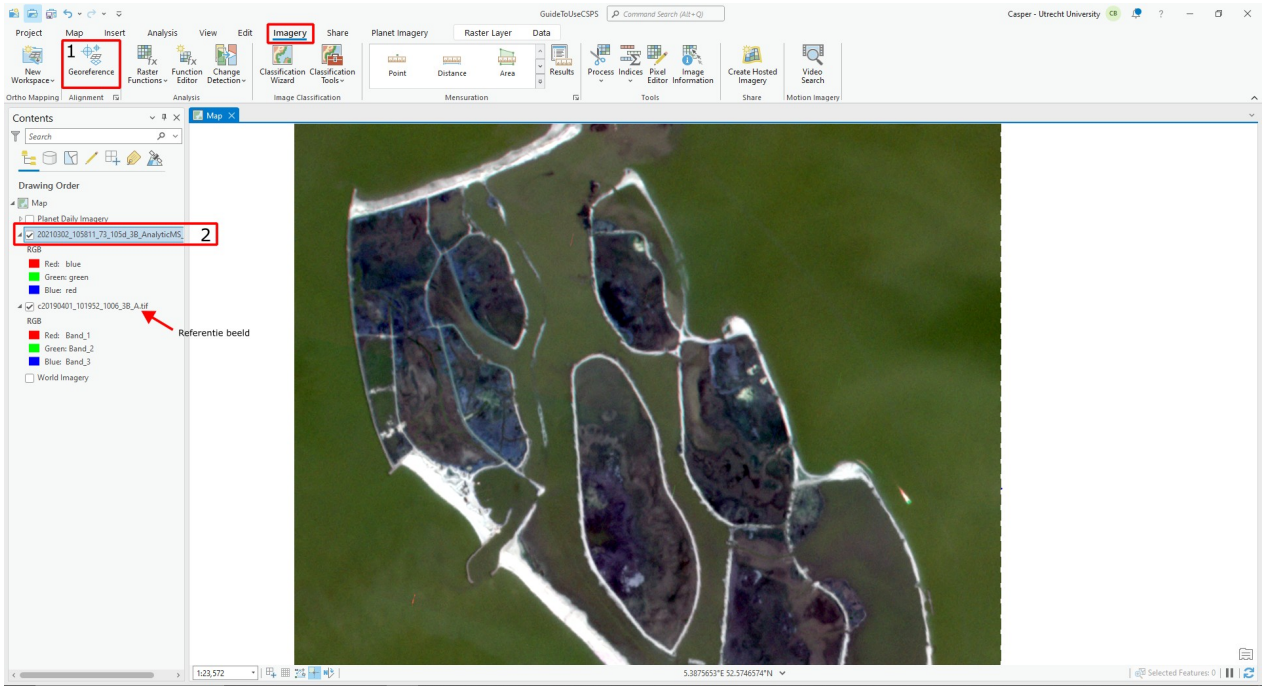
Step 2:

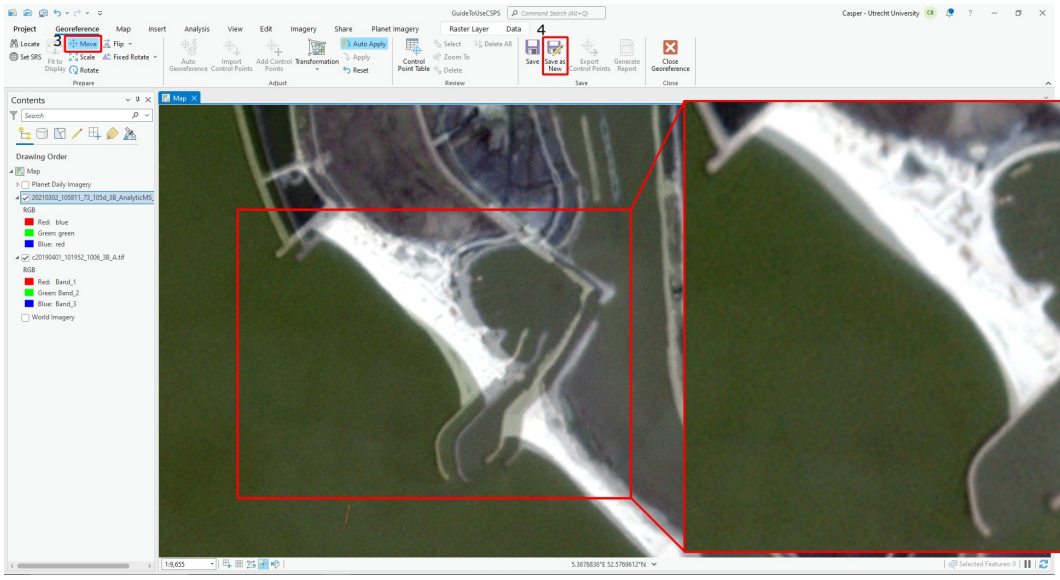
Select the reference image, which in the case of [**My Study**] is the PlanetScope image '20190401_101952_1006'.

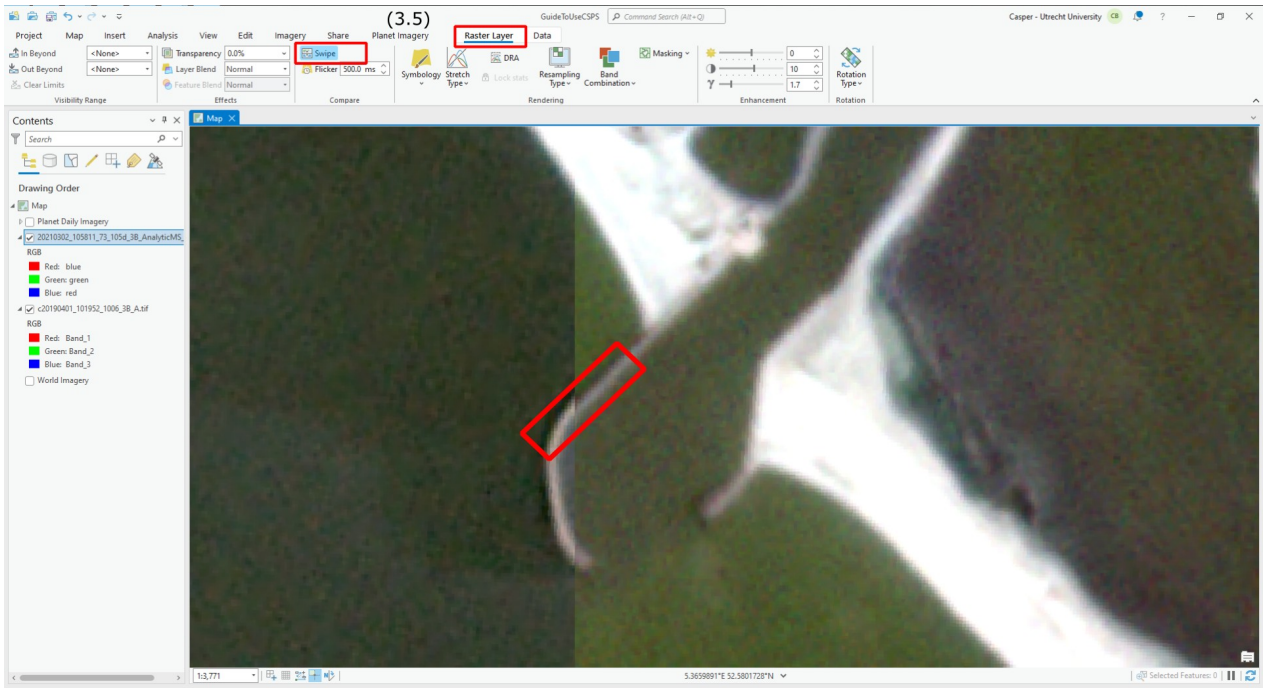
1. Within the '*Imagery*' tab, search for the '*Georeference*' tool.
2. Select the image that needs to be corrected.
3. Choose the '*move*' option and drag the image so that it aligns exactly with the reference image.

(3.5) Use the '*swipe*' tool to verify the alignment of the images.

4. '*Save as new*' for the corrected image, in a folder containing all the corrected images.







Step 3:

These are the exact settings that were used and need to be adjusted in the main script of CoastSat.PlanetScope. Add the corrected satellite images, along with the original UDM (.tiff) files, to the downloads folder.

```
settings = {

            ### General Settings ###
            # Site name (for output folder and files)
            'site_name': 'MarkerWadden_Zuidstrand',
            # Maximum image cloud cover percentage threshold
            'cloud_threshold': 10, # Default 10
            # Minimum image AOI cover percentage threshold
            'extent_thresh': 25, # Default 80
            # Desired output shoreline epsg
            'output_epsg': '28992',


            ### Reference files (in "...CoastSat.PlanetScope/user_inputs/") ###
```

```
            # Area of interest file
            'aoi_kml': 'Zuidstrand.kml',
            # Local folder planet imagery downloads location (provide full folder path)
            'downloads_folder': '___Init directory___/CoastSat_PlanetScope/
                                'user_inputs/Satellite_Imagery/',

            ### Processing settings ###
            # Machine learning classifier filename
            'classifier': 'ZS_All_tresh50_900000_NARRA_9639.pkl', # Newly trained classifier
                                                                  # based on 20 Images of Zuidstrand
                                                                  # (See: Bakker(2023))
            # Image co-registration choice ['Coreg Off', 'Local Coreg', 'Global Coreg']
            'im_coreg': 'Local Coreg',


            ### Advanced settings ###
            # Buffer size around masked cloud pixels [in metres]
            'cloud_buffer': 9, # (3 pixels)
            # Max distance from reference shoreline for valid shoreline [in metres]
            'max_dist_ref': 75,
            # Minimum area (m^2) for an object to be labelled as a beach
            'min_beach_area': 20*2500,
            # Minimum length for shoreline [in metres]
            'min_length_sl': 2500,
            # GDAL location setting
            'GDAL_location': '___/anaconda3/envs/___/bin/', # Need the full path to directory

            }
```

Step 4:

To find the shoreline, the code of CoastSat.PlanetScope is used. Since CoastSat.PlanetScope doesn't exactly meet the requirements for this application, a new script has been written that can be added to the original code. This new code is freely available through this link. Below is the 'main' script that is executed in the terminal and ensures that everything is done automatically.

```
from Classes.AdditionalFunctions import Translate_udmMask
from Classes.AdditionalFunctions import Store_Coordinates
from CoastSat_PS import settings as Input_Settings

def main():
    """
        Function to run the main script and skip the
        steps that are unneccesary for this method.
        (This script runs till line 122, check if this
         untill step 3)
    """

    with open('CoastSat_PS.py', 'r') as file:
        script_lines = file.readlines()
    script_to_execute = ''.join(script_lines[:122]) # read script till line 122
    exec(script_to_execute)

if __name__ == '__main__':

    print('========================== Step 1 ============================')
    print('Translating the UDM to the same extent as the georeferenced imagery:')
    Translate_udmMask(Input_Settings['downloads_folder'])

    print('========================== Step 2 ============================')
    print('Running the main script of CoastSat.PlanetScope')
```

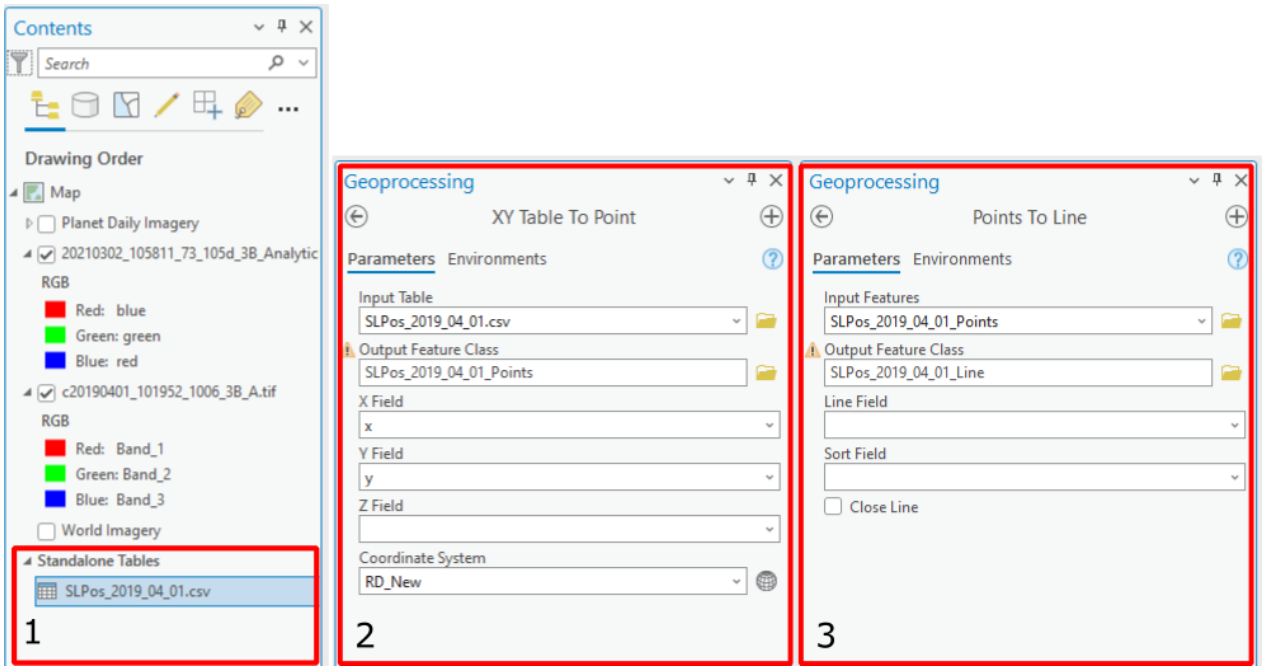
```
main()

print('========================== Step 3 ==========================')
print('Storing the output shorelines as .csv-files: ')
pkl_file = 'shoreline outputs/Local Coreg/NDWI/Peak Fraction/' + str(Input_Settings['site_name'])
data = os.path.join('outputs', str(Input_Settings['site_name']), pkl_file)
Storage_location = os.path.join('Shoreline_Positions/', str(Input_Settings['site_name']))
if os.path.exists(Storage_location):
    Store_Coordinates(data, Storage_location)
else:
    os.mkdir(Storage_location)
    Store_Coordinates(data, Storage_location)
```

Step 5:

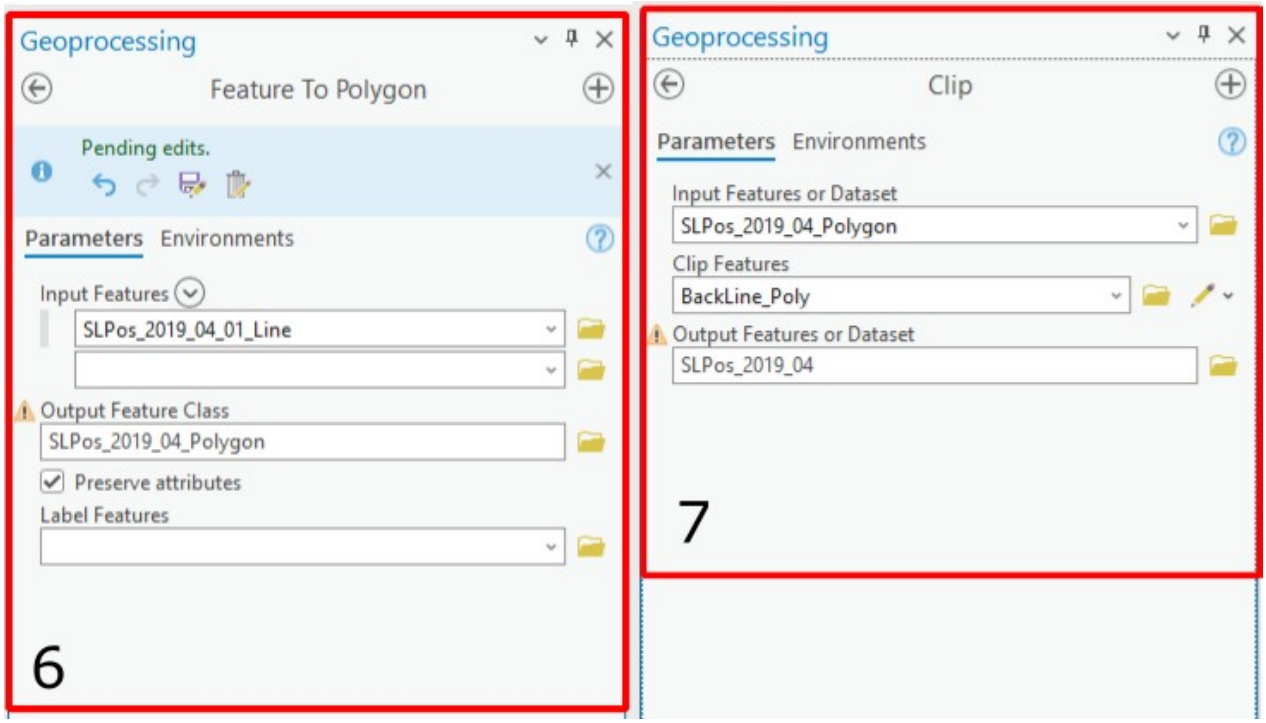
The final step is processing the shoreline into an area in ArcGIS Pro. This is done by converting the lines into closed polygons. Here are the steps:

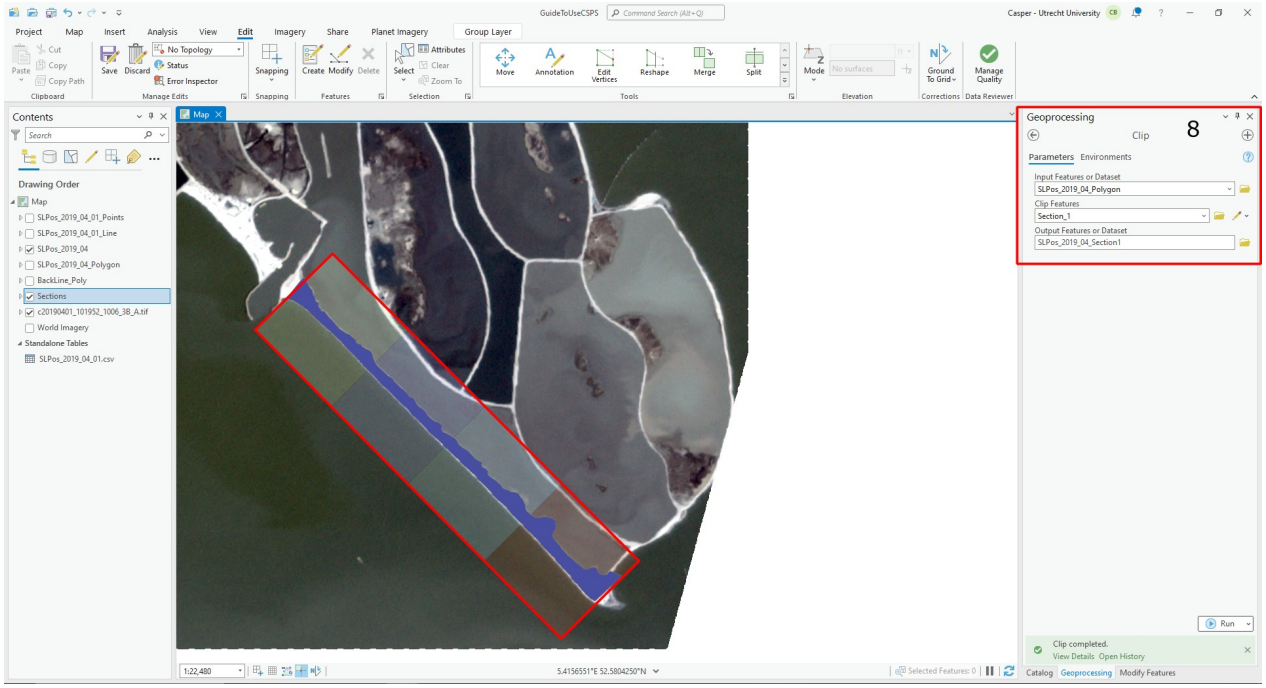
1. Drag the .csv files containing the shoreline positions into the GIS project, so that the files are added as standalone tables.
2. In the ”Geo-processing”section of the Analysis Toolbox, search for ”XY Table To Point”to add points representing the shoreline.
3. In the ”Geo-processing”section of the Analysis Toolbox, search for ”Points To Line”to connect the points and create a line.
4. In the ”Edit Tools”section, search for ”Copy parallel.”Select the line and enter the correction value (ensure the correction value is positive or negative, and the method is left or right). Proceed with the new line and remove the uncorrected line.
5. Edit the line by clicking on ”Modify Featureänd ”Continue Feature”to create a closed loop.
6. In the ”Geo-processing”section of the Analysis Toolbox, search for ”Feature To Polygon”to convert the line into a polygon.
7. Clip the polygon with another polygon representing the backshore of the beach, ensuring that the line aligns with the backshore. This will create a new polygon that accurately follows the shape of the beach.
8. Extra step: Clip the beach polygon with additional sections along the coast to study alongshore variations.
9. Open the Attribute Table (CTRL-T) and the value in the ”Shape-Area”field represents the area of the beach (in square meters).




Map
Modify Features
Copy Parallel
Change the selection.
SLPos_2019_04_01_Line (1)
5365.69090858746
Template
Use Template
SLPos_2019_04_01_Line
Keep source attribute values
Features
Copy to separate features
Copy to a single feature
Values 4
Distance 2.3 m
Repetitions 1
Method
Side
Both
Left
Right
Corners
Mitered
Rounded
Beveled
Remove self-intersecting loops
Align the direction of connected lines
Show Preview
Copy
1:1,145
5.3832367°E 52.5725626°N
Selected Features: 1
Catalog
Geoprocessing
Modify Features


Map
Modify Features
Continue Feature
Continue Feature completed.
Change the selection.
SLPos_2019_04_01_Line (1)
5365.69090858746
Continue Feature 5
1:22,480
5.3646301°E 52.5872424°N
Selected Features: 1
Catalog
Geoprocessing
Modify Features





To autimate this process python code can be used. The provided code gives an example using a 'for-loop'

```
import os

lst = ["SLPos_2019-04-01_101952.csv", "SLPos_2019-04-21_085532.csv"]
names = ["SLPos_2019_04_01_points", "SLPos_2019_04_21_points"]

directory = "...\\toGeoDataBase_of_GIS_project.gdb"

for index, value in enumerate(lst):
    # Loop through the list and do for every item in the list this operation
    output_feature = os.path.join(directory, names[index])
    arcpy.management.XYTableToPoint(value, output_feature,
                                    "x", "y", None, 'PROJCS["RD_New"] ... ')
    # Creating points from table
```

```
# Functions to call for the different operations:
arcpy.management.PointsToLine(value, output_feature, None, None, "NO_CLOSE")
arcpy.management.FeatureToPolygon(value, output_feature, None, "ATTRIBUTES", None)
arcpy.analysis.Clip(value, "Backline_Poly", output_feature, None)
arcpy.analysis.Clip(value, "Section_5", output_feature, None)

for index, value in enumerate(names):
    fc = os.path.join(directory, value)
    field = "Shape_Area"
    cursor = arcpy.SearchCursor(fc)
    for row in cursor:
        print(row.getValue(field))
```