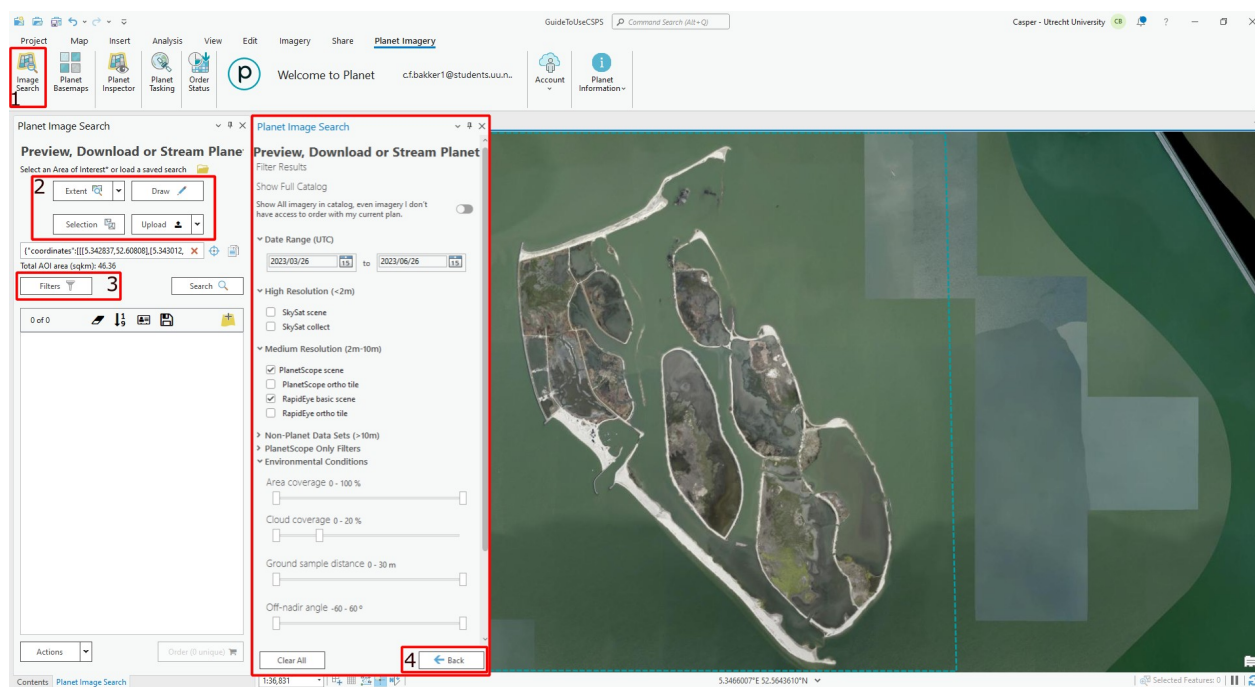


Stap 1:

Krijg toegang tot PlanetScope-beelden door een account aan te vragen. Studenten kunnen een gratis account aanvragen via deze [link](#). Als je eenmaal een account hebt, kun je beelden opvragen via QGIS, ArcGIS Pro of Google Earth Engine. Meer informatie is te vinden op deze [site](#). Hieronder vind je een stappenplan voor het gebruik van ArcGIS Pro, waarvoor de [plugin voor ArcGIS Pro](#) geïnstalleerd moet zijn.

1. Selecteer de image search window
2. Selecteer het interesse gebied, dit kan door een gebied te tekenen of door een .json-file te uploaden
3. Selecteer de filters
4. Selecteer een medium-resolutie PlanetScope-scène en een RapidEye-basic-scène (Op dit moment zijn de high-resolutiebeelden niet beschikbaar zijn in Nederland). Zorg ervoor dat de bewolgingsgraad minder dan 20% is, hoewel de locatie van de wolken individueel kan verschillen dus het is het belangrijkste dat het strand volledig zichtbaar.
5. Bevestig de instellingen.
6. Maak een keuze voor de satellietbeeld, voeg de beelden toe aan de kaart en inspecteer ze visueel.
7. Na alle beelden te hebben gekozen, kan de bestelling worden gemaakt
8. Bevestig de order, waar het beeld in het format: 'Analytic radiance (TOAR) - 4band GeoTIFF' en kies voor 'clip to AOI.'
9. De order is te download van een zip-file, waarvan de .tiff-bestanden worden gebruikt.



Planet Image Search

Preview, Download or Stream Planet Imagery

Select an Area of Interest* or load a saved search

Extent Draw Selection Upload

["coordinates":[[[5.342837,52.60808],[5.343012,52.564467],[5.414093,52.564467]]]]

Total AOI area (sqkm): 46.36

Filters Search

74 of 74

Jun 25, 2023 PlanetScope scene 2 Images

Jun 24, 2023 PlanetScope scene 4 Images

09:45:23 UTC PSB.SD Satellite 24ce 2 Images

09:43:43 UTC PSB.SD Satellite 2460 2 Images

Jun 24 2023 09:43:43 UTC PlanetScope scene Cloud UDM2: 0% GSD: 3.7m

5 Jun 24 2023 09:43:41 UTC PlanetScope scene Cloud UDM2: 0% GSD: 3.7m

Jun 23, 2023 PlanetScope scene 1 Images

Jun 22, 2023 PlanetScope scene 1 Images

Jun 21, 2023 PlanetScope scene

Actions **6** Order (1 unique)

Bekijk de satellietbeelden van tevoren

8 Order Imagery

Order Imagery

1 Name Order 2 Select Assets 3 Tools & Review

Order Name: GuideToCueCSPS 14/000

Order Imagery

1 Name Order 2 Select Assets 3 Tools & Review

1 Items PlanetScope scene

RECTIFIED ASSETS

Visual
Optimized for visual analysis - RGB only
GeoTIFF NTF

Surface reflectance - 4 band
Corrected for surface reflectance: recommended for most analytic applications - includes RGB NIR
GeoTIFF NTF UDM2

Surface reflectance - 8 band
Corrected for surface reflectance: recommended for most analytic applications - also includes coastal blue, green II, yellow, red edge
GeoTIFF NTF UDM2

Analytic radiance (TOAR) - 4 band
Calibrated to top of atmosphere radiance - includes RGB NIR
GeoTIFF NTF UDM2

Analytic radiance (TOAR) - 8 band
Calibrated to top of atmosphere radiance - also includes coastal blue, green II, yellow, red edge
GeoTIFF NTF UDM2

Show More

Back Continue

Order Imagery

1 Name Order 2 Select Assets 3 Tools & Review

1 Items PlanetScope scene - Analytic radiance (TOAR) - 4 band

CLIPPING: Only get imagery defined within your AOI

Clip Name to AOI

METADATA: COG metadata provides a standardized format for describing geospatial information so that you can more easily understand and use your data

STAC

REVIEW ITEMS: We recommend downloading items that appear to have no errors

Download All

Back Order

Order Summary

Order name: GuideToCueCSPS

Number of Orders: 1

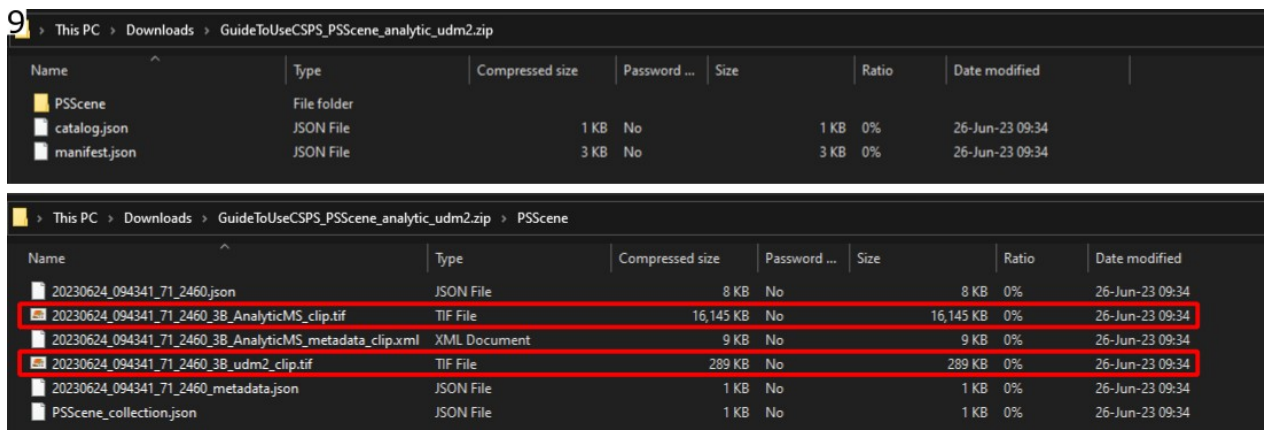
* Each asset will be placed as a separate order

PlanetScope scene

1 Items

Analytic radiance (TOAR) - 4 band

GeoTIFF UDM2 Clip



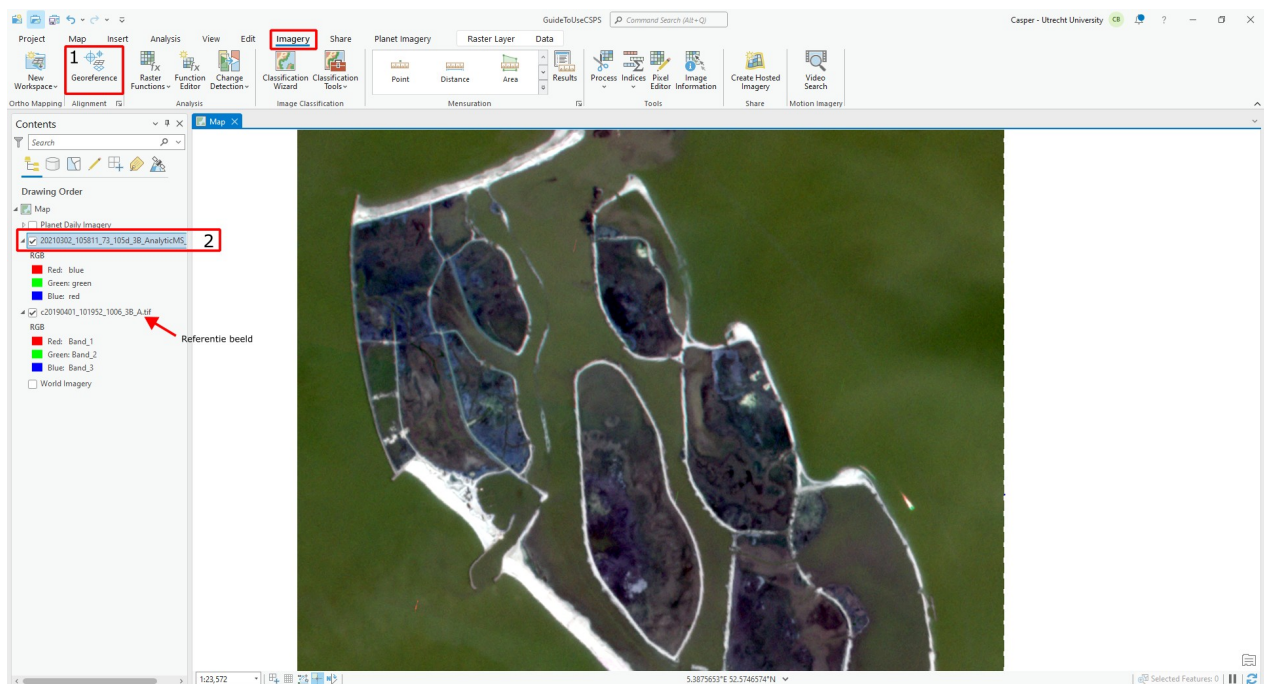
Stap 2:

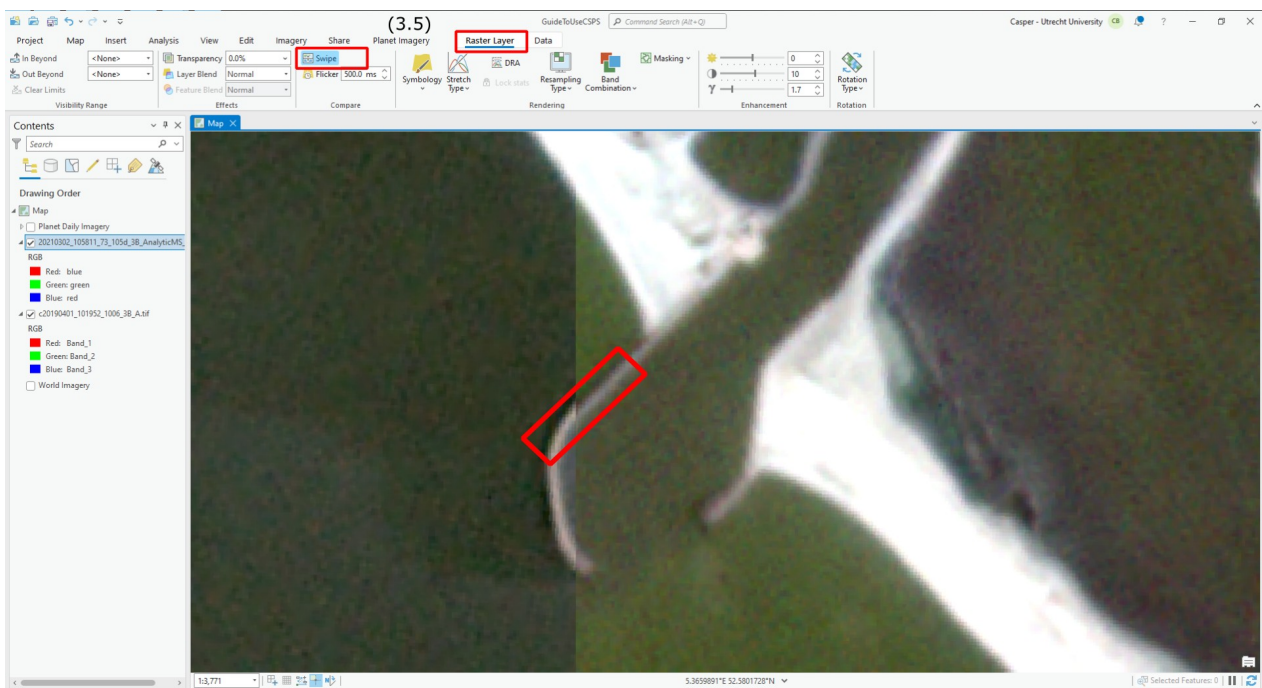
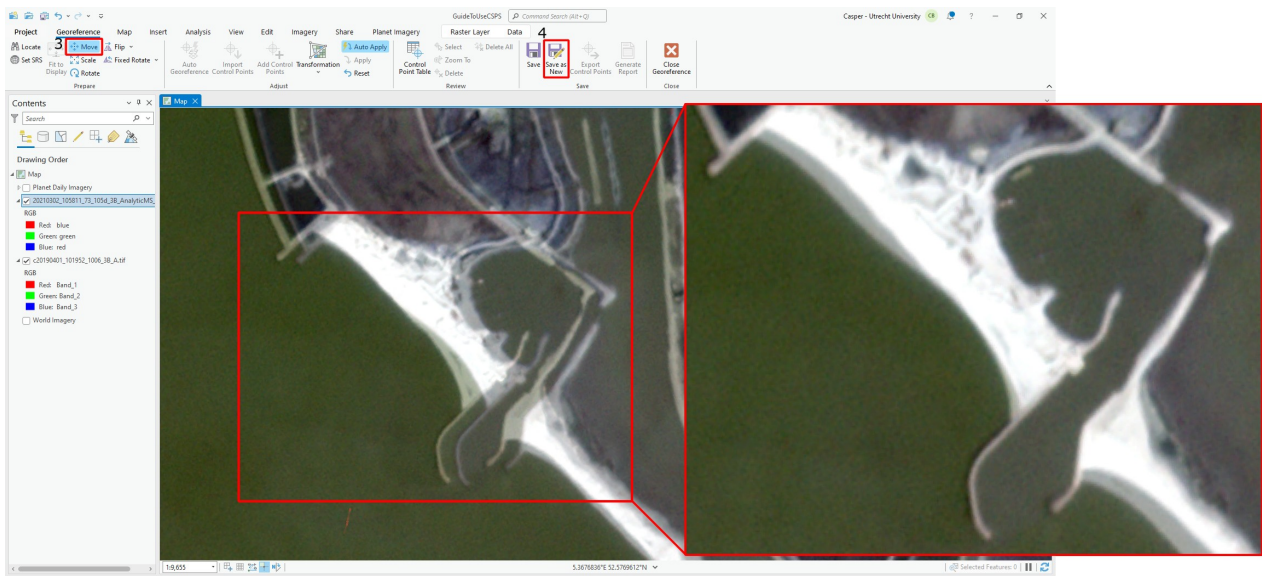
Selecteer het referentie beeld, dit is in het geval van [Mijn Stuide] het beeld van PlanetScope '20190401_101952_1006'

1. Zoek binnen de tab van 'Imagery' naar de tool 'Georeference'.
2. Selecteer het beeld dat moet worden gecorrigeerd.
3. Kies de 'move' optie en sleep het beeld, zodat het exact overeen komt met het referentie beeld.

(3.5) Controleer de beelden met behulp van de 'swipe' tool.

4. 'save as new' voor het gecorrigeerde beeld, in een folder met alle gecorrigeerde beelden.





Stap 3:

Dit zijn de exacte instellingen die gebruikt zijn en die moeten worden aangepast in het hoofd-script van CoastSat.PlanetScope. Voeg de gecorrigeerde satellietbeelden, samen met de originele UDM-.tiff bestanden, toe aan de 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:

Voor het vinden van de oeverlijn wordt de code van CoastSat.PlanetScope gebruikt. Omdat CoastSat.PlanetScope niet precies aan de eisen voor deze toepassing voldoet, is er een nieuw script geschreven dat aan de originele code kan worden toegevoegd. Deze nieuwe code is vrij beschikbaar via deze [link](#). Hieronder staat het 'main'-script dat in de terminal wordt uitgevoerd en ervoor zorgt dat alles in één keer wordt gedaan.

```

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 unnecessary 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()

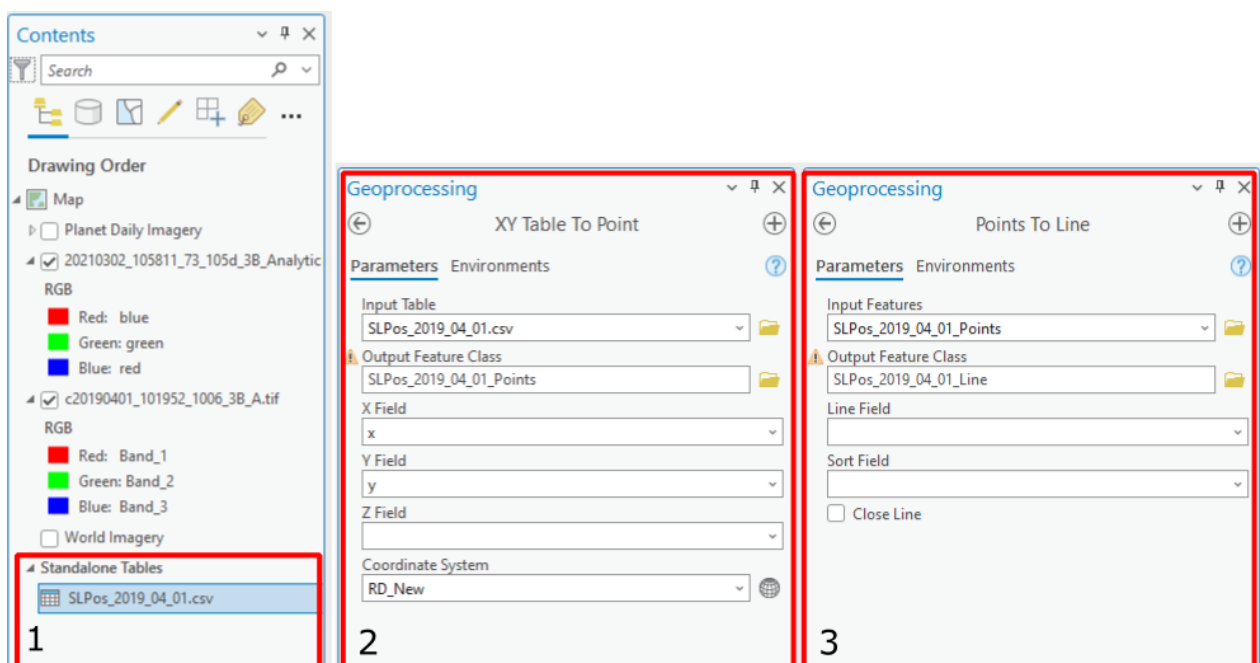
print('===== Step 3 =====')
print('Storing the output shorelines as .csv-files: ')
pkl_file = 'shoreline outputs/Local Coreg/NDWI/Peak Fraction/' + str(Input_Settings['site_name'])
data = os.path.join('outputs', str(Input_Settings['site_name']), pkl_file)
Storage_location = os.path.join('Shoreline_Positions/',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

```

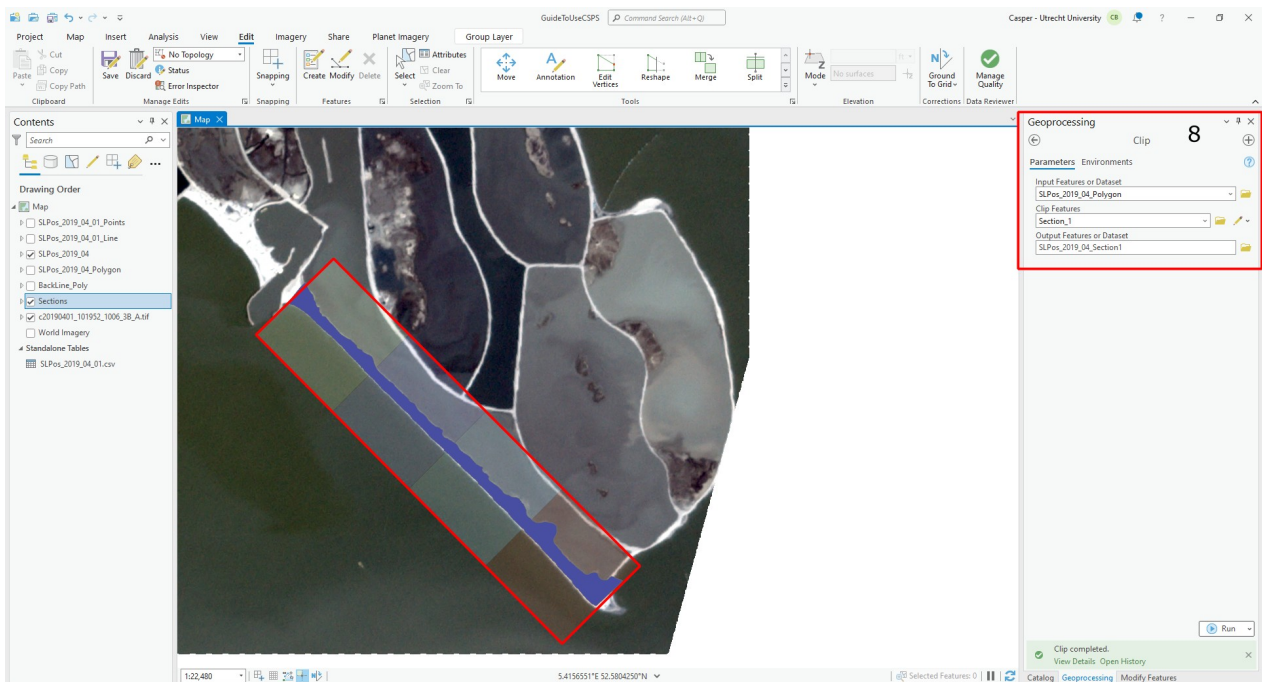
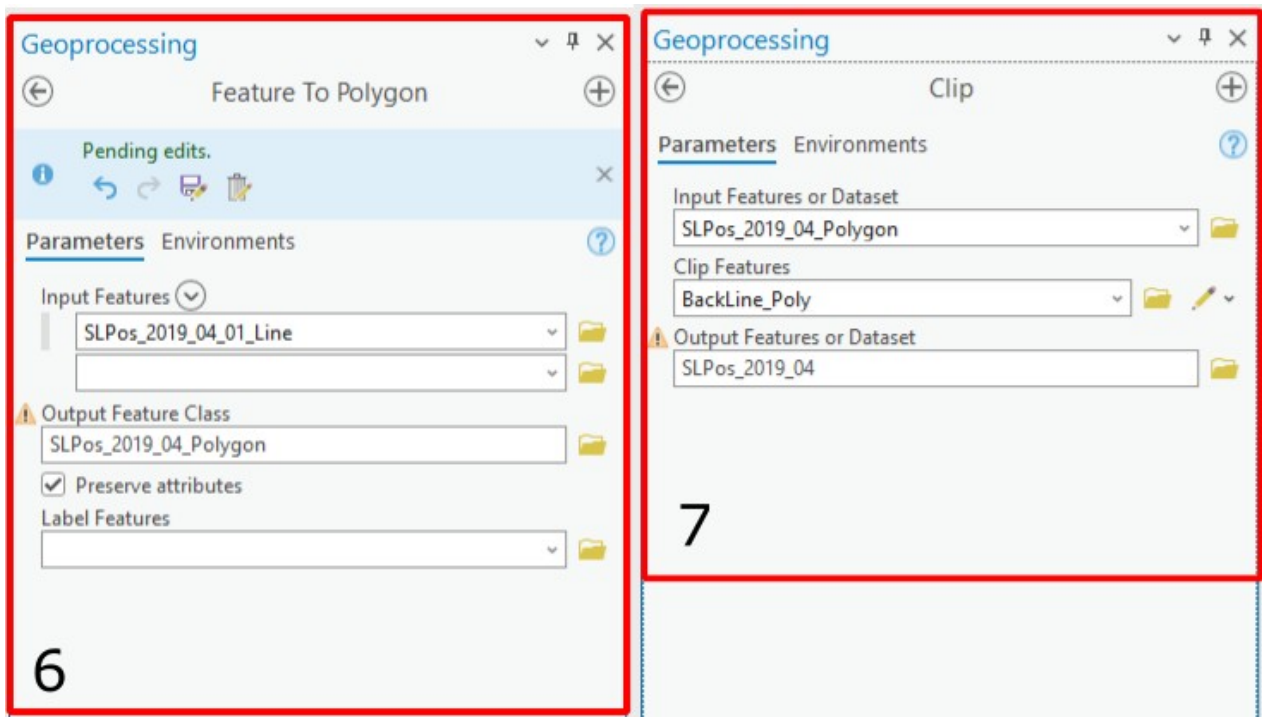
Stap 5:

De laatste stap is het verwerken van de oeverlijn naar het oppervlakte. Dit wordt gedaan in ArcGIS Pro waar de lijnen tot sluitende polygoon worden omgezet en daarna

1. Sleep de .csv-bestanden met de oeverlijn positie in het GIS-project, zodat de bestanden als 'standalone tables' zijn toegevoegd.
2. Zoek in 'Geo-processing' (Analysis Toolbox) naar 'XY Table To Point', zodat de punten van de oeverlijn worden toegevoegd.
3. Zoek in 'Geo-processing' (Analysis Toolbox) naar 'Points To Line', zodat de punten een lijn vormen.
4. Zoek in 'Edit Tools' naar 'Copy parallel', selecteer de lijn en vul de correctie waarde in. (Let op of de correctie-waarde positief of negatief en dat de *method* links of rechts overeenkomt). Werk hierna verder met de nieuwe lijn en verwijder de ongecorrigeerde lijn.
5. 'Edit' de lijn, door op 'Modify Feature' en 'Continue Feature' te klikken en vorm een sluitende lijn.
6. Zoek in 'Geo-processing' (Analysis Toolbox) naar 'Feature To Polygon', zodat de lijn een polygoon vormt.
7. 'Clip' de polygoon met een polygoon waar de lijn overeenkomt met de achterlijn van het strand. Zodat er een nieuwe polygoon vormt die precies vorm van het strand volgt.
8. Extra stap: 'Clip' de polygoon van het strand met extra secties langs het strand om de kustlangse variatie te bestuderen.
9. Open de 'Attribute Table' (CTRL-T) en de waarde van de 'Shape-Area' is het oppervlakte van het strand (in m²).







Om het proces in ArcGIS Pro te automatiseren kan python worden gebruikt. Hieronder is een voorbeeld hoe met een 'for-loop' alles kan worden gedaan.

```
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))

```