

ECOSTRESS Geolocation Python Batch: User Guide

ECOSTRESS TUTORIALS

This tutorial will show you how to use the Batch Python ECOSTRESS Geolocation code.

Table of Contents

<i>ECOSTRESS Geolocation Accuracy</i>	2
<i>Where to obtain ECOSTRESS Data?</i>	2
<i>Necessary File(s)</i>	2
<i>ECOSTRESS Geolocation Setup</i>	3
Where to Download ECOSTRESS Geolocation Batch Python Code	3
Setting up a Geolocation Python Environment	3
<i>Running the Batch Python code</i>	5
Specifying your python environment	5
Editing the customizable variables	6
Set Variables	6
Running the Code	6
<i>Python Batch Code– Steps Overview</i>	7
File Setup	7
Water Mask Processing	7
Image Cropping	8
Primary Line Selection	8
Acquiring Possible Shifts	9
Final Shift Test	10
Applying the Shift	10

ECOSTRESS Geolocation Accuracy

The ECOSTRESS sensor aboard the International Space Station (ISS) provides hundreds of scenes every day and geolocating these scenes is crucial. The ECOSTRESS team at NASA JPL have algorithms to match scenes to their correct location using ground truth points. However, there are still scenes that may have geolocation errors of up to 7km. ECOSTRESS scenes located near bodies of water have noticeable errors as observers can easily find discrepancies between a base map and the ECOSTRESS scene. This code aims to use ECOSTRESS's water mask file to correctly place the ECOSTRESS LST scene.

The work uses the method developed by Soszynska, van der Werff, Hieronymus, and Hecker, "A New and Automated Method for Improving Georeferencing in Nighttime Thermal ECOSTRESS Imagery", 2023, <https://doi.org/10.3390/s23115079>

Where to obtain ECOSTRESS Data?

ECOSTRESS data is available on AppEEARS which requires an account to download images. Please refer to JPL's ECOSTRESS Tutorials ECOSTRESS Tutorial [06-Downloading From AppEEARS](#) for steps on setting up your account and downloading data. This code primarily works best with Collection 2 data as Collection 2 provides personalized water masks for each scene. AppEEARS provides each of the necessary files to a bounding box based on the users Region of Interest (ROI).

Necessary File(s)

This code was built to process ECOSTRESS Collection 2 from [AppEEARS](#). There are two versions of ECOSTRESS LST Collection 2 data with Swath and Tiled. To obtain Swath tiled products you may search for the product: "ECOSTRESS Swath Land Surface Temperature and Emissivity Instantaneous" or search for the tiled product using: "ECOSTRESS Tiled Land Surface Temperature and Emissivity Instantaneous". Below are the essential files with an optional cloud mask for additional edge removals.

- Land Surface Temperature (LST)
- Quality Control (QC)
- Water Mask
- Cloud Mask (**optional**)

*Note: Running the **python batch code** with the single water mask option for a folder that has swath and tiled scenes with their native projections will result in errors. Please request your scenes from AppEEARS in any projection besides "Native". The projection can be set under the "Projection: box*

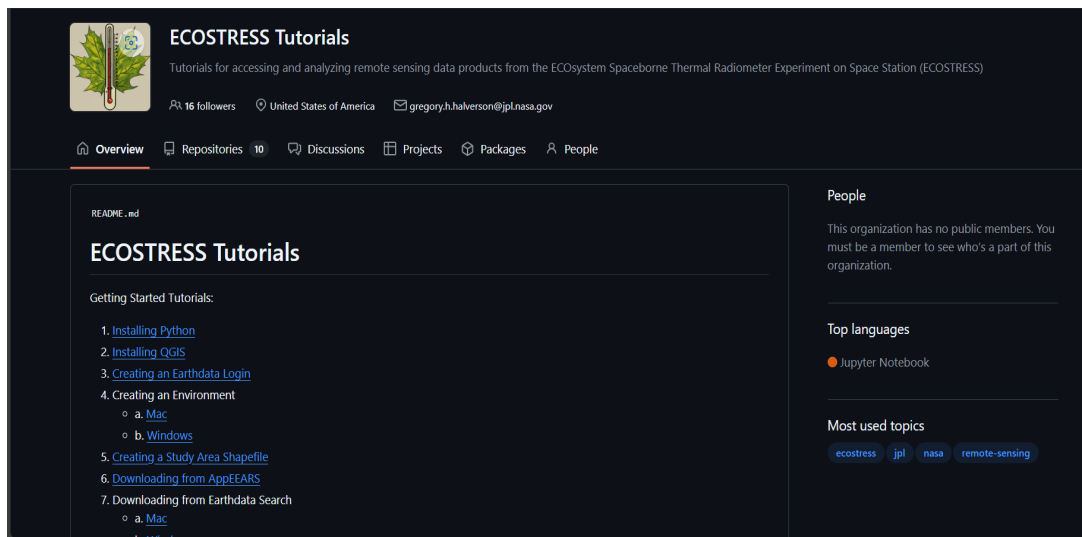


under Output Options. If you plan on using Swath only or Tiled only, then this should not be an issue.

ECOSTRESS Geolocation Setup

Where to Download ECOSTRESS Geolocation Batch Python Code

1. Start by accessing ECOSTRESS Tutorial [Github repository](#). This page should look like this:



2. Select the **ECOSTRESS Geolocation** link under “Python Tutorials:”
3. Navigate to **Python Geolocation Folder** and download the files.
4. Open the .py file with Visual Studio Code.

Note: If you do **not** have Visual Studio Code installed in your computer, please follow Tutorial Lesson [08-Downloading_Visual_Studio_Code](#) to set up your Visual Studio Code application.

Setting up a Geolocation Python Environment

1. If you are unfamiliar with setting up Python Environments, please follow Tutorial Lesson [04-Creating an Environment](#) to set up a python environment up until the **CREATING AN ENVIRONMENT** segment. This Tutorial has a Window and Mac version.
2. In the Anaconda Prompt: type the command



```
mamba create -y --name Geolocation python=3.12
```

- a. **mamba create** is the command to make the environment.
 - b. **-y** confirms changes being made.
 - c. **--n Geolocation** is used to name our environment. In this case the environment is being named Geolocation but if you would like a different name, you can change it. Just make sure to keep the **-n** and not use spaces or special characters in your name.
 - d. For the end of the command, we list the most recent python version as of January 29th, 2024:
 - i. **python=3.12** connects to python, in this case setting it to version 3.12.
3. Let the command run. You will know it is done when you get these instructions on how to activate and deactivate the environment

```
To activate this environment, use

$ mamba activate Geolocation

To deactivate an active environment, use

$ mamba deactivate
```

4. Next we will activate the Geolocation environment using **mamba activate Geolocation**. This will replace (base) with (Geolocation)

```
(base) C:\Users\alamillo>conda activate Geolocation

(Geolocation) C:\Users\alamillo>
```

5. With the Geolocation environment active, copy and paste the following commands after each is complete:

```
(Geolocation) C:\Users\alamillo>pip install numpy opencv-python scikit-image pandas matplotlib ipykernel
```

```
pip install numpy opencv-python scikit-image pandas matplotlib ipykernel
```

```
(Geolocation) C:\Users\alamillo>conda install gdal
```

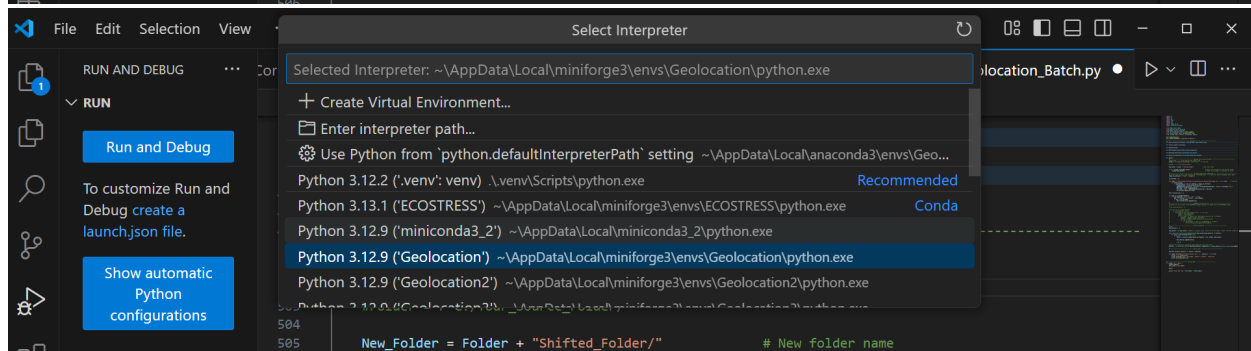
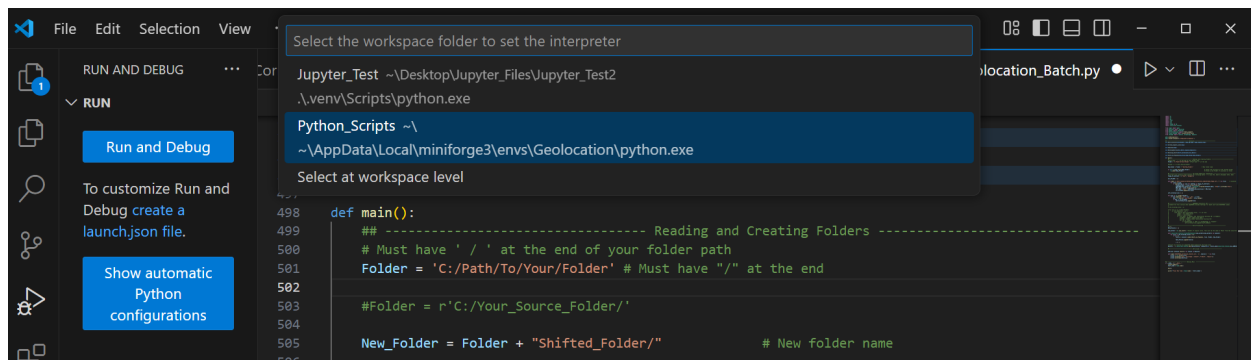
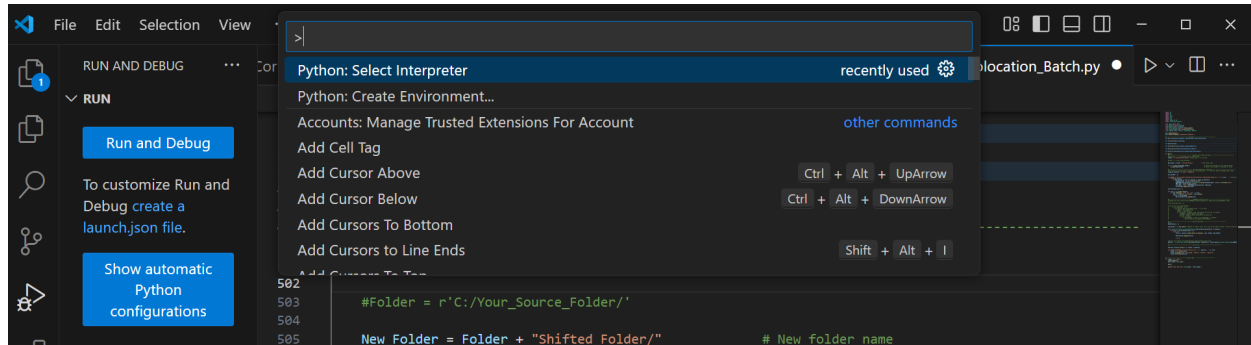
```
conda install gdal
```

- Once each line has finished running, the environment is ready for the ECOSTRESS Geolocation Jupyter Notebook or Python Batch

Running the Batch Python code

Specifying your python environment

With the ECOSTRESS_Geocorrection Batch Code in Visual Studio, Press and hold the “Ctrl”, “Shift” and “p” keys to open the python interpreter and navigate to your python environment.



Note: Clicking on the bottom right of your Visual Studio Code window shows the same prompt

Editing the customizable variables

With your python environment selected, you will be able to expand the “Main” function within the script. Within this function you may alter the “Folder” variable to your desired folder path.

```
Folder = r'C:/Path/To/Your/Folder/' # Must have "/" at the end
```

Set Variables

This python batch code can be run in four different ways depending on the usage of cloud masks for each scene or the use of a single water mask to match all scenes. These two variables “Use_Cloud_Mask” and “Use_Parent_WaterMask” are used to guide the code into using certain sections of the code. These variables were put in place due to some inconsistencies with the cloud mask values as well as the quality of the Water Masks. Cloud masks may identify land temperatures as clouds and water masks may have irregular shorelines due to unidentified clipping. It is up to the user if they would include cloud masks into their results and if they want to use a single water mask as the main file for shift calculation. To use a single water mask for your entire scene, you may set a valid water mask file name within your folder to mark “Use_Parent_WaterMask” as True. To use the cloud masks for each scene, simply alter the “False” value for the variable “Use_Cloud_Mask” to True.

```
Use_Cloud_Mask = False

Main_WM_File = 'My_Parent_Water_Mask.tif'

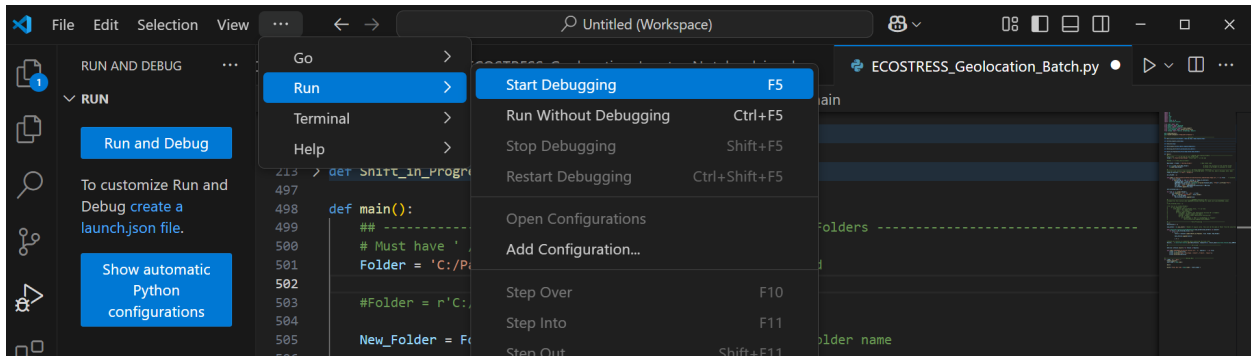
Main_WM_Full_Name = os.path.join(Folder, Main_WM_File)

Use_Parent_WaterMask = os.path.exists(Main_WM_Full_Name)
```

Note: If a single water mask is used to shift every scene in your folder, then only scenes that overlap with your defined water mask.

Running the Code

With the customizable variables set to your specifications, you may run the script by going to the run tab in the top right and select “start debugging”

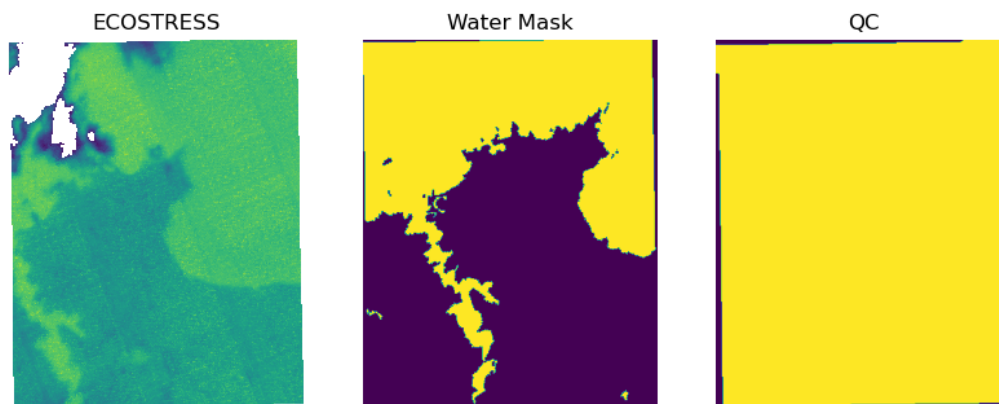


Python Batch Code– Steps Overview

This will go over the main functions and steps for finding the shift. For any specifications on how each file is processed within the code, visit the Batch Python Documentation docs which contains a detailed description of each step of the code.

File Setup

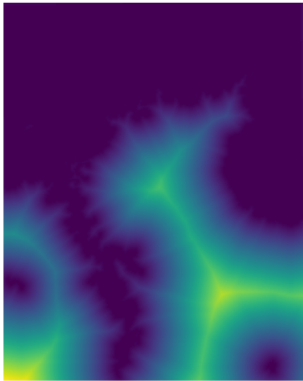
Your Land Surface Temperature, Water Mask, and QC Mask images will be read and opened using `gdal.Open()` and read as an array.



Water Mask Processing

The `cv2` function `distanceTransform` is used to identify the distance of the water mask pixels to another pixel (proximity of 0's to 1's). This step will give a shoreline with the scene border. The QC mask is used to eliminate the scene border (Seen in the top right section of Water Mask Edges and Water mask w/o Img Boundary).

Water Mask Distance to Edges



Water Mask Edges



Water Mask w/o Img Boundary

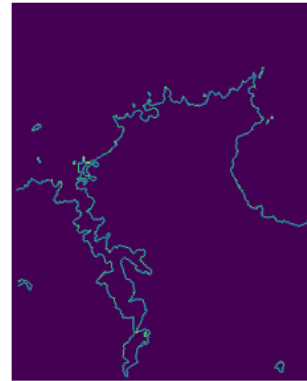
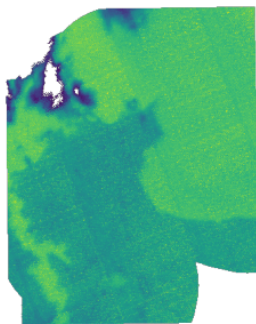


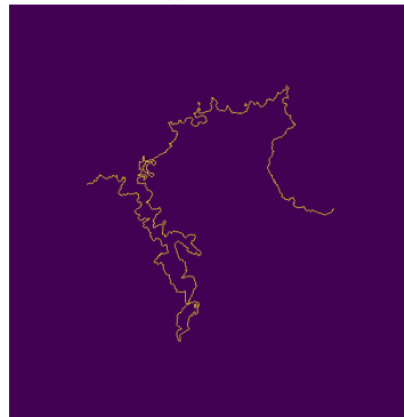
Image Cropping

Using the new water mask shoreline, both the water mask and LST can be clipped to their bounding box with a 100-pixel buffer on all sides. Additionally, the LST image will be masked using a 100-m buffer of the water mask shoreline. This reduces the area of interest within images, especially in large scenes.

Clipped ECOSTRESS

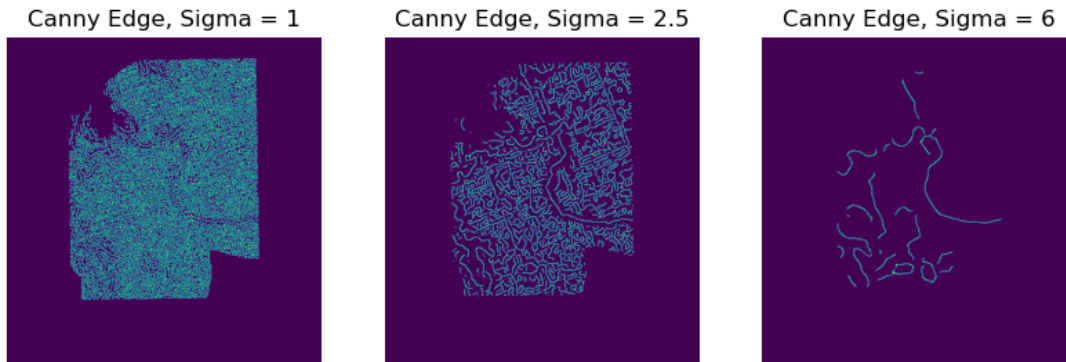


Clipped Water

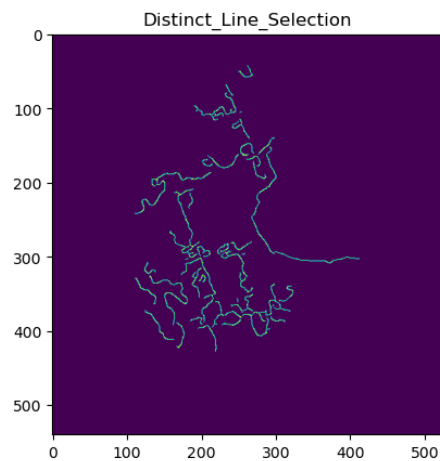


Primary Line Selection

The LST image will have edge detection applied to identify all the edges in the scene. A gaussian smoothing of 2.5 (known as sigma) is applied to the scene to help reduce noise and help identifying noticeable change in values within a certain pixel distance from the pixel of interest.

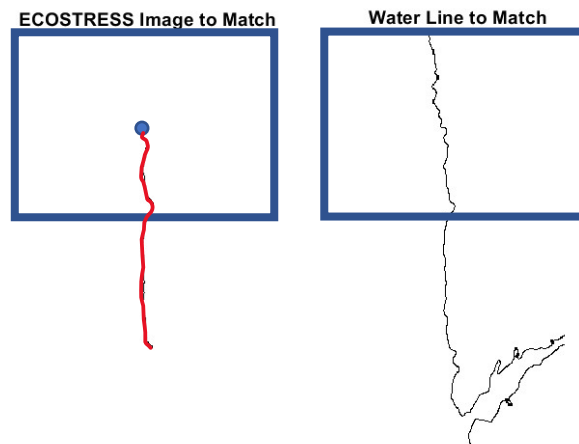


Using a gaussian smoothing of 6, the edge detection process has less noise reducing the number of lines, but decreasing the shoreline accuracy. The 2.5 sigma smoothed scene and 6 sigma smoothed scenes are compared for overlapping lines. The lines that overlapped are used for match selection.



Acquiring Possible Shifts

Each line will be isolated and will be used to clip their water mask counter parts. Using the top right most point of the ECOSTRESS line, the line will travel along the water mask (right). The

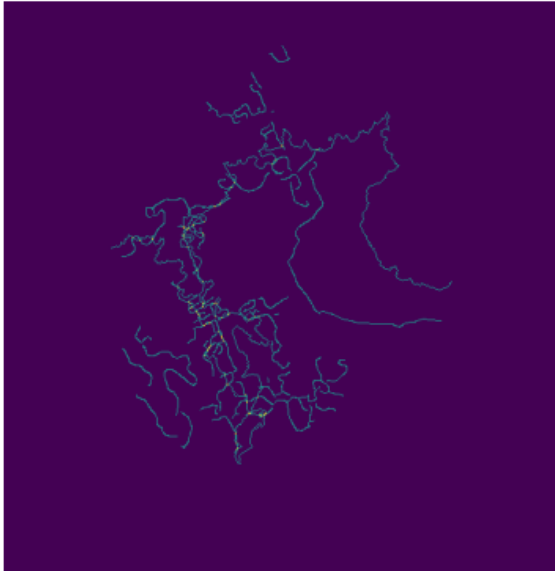


shift values will be limited to water mask pixel within a box resembling the 100-pixel error (7km error) shown below as the blue box:

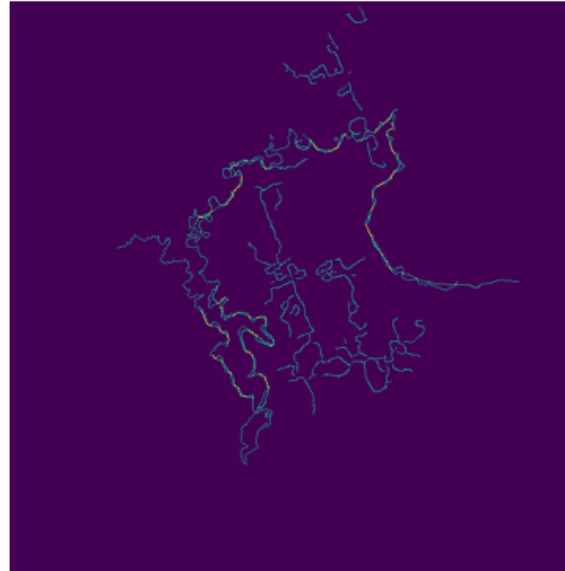
Final Shift Test

This step is done for individual water mask shorelines (in the case of islands). Once this step is done for every water mask shoreline, the final image can be used for matching using the best shifts calculated from every individual water mask shoreline shift tests.

Original Image Overlay



Shifted Image Overlay



Applying the Shift

The final shift will be applied to a **copy** of your original LST image within a subfolder named “Shifted_Folder”.