How to Conduct Image Classification & Segmentation Using Google Earth Engine & QGIS

Designed by Emily Evenden and Max Enger

What you will need:

- Google Earth Engine Account
- A recent version of QGIS (Recommend version 3.18)
- Orfeo toolbox plug-in
- A Sentinel image of Rhode Island from May, 2020
- The data folder contains the output files from each step of this process. Some of these processes take a long time to run - therefore, you can see the output without running the process yourself.

This tutorial provides you with step-by-step instructions to conduct pixel-based classification of solar fields using Random Forest & Google Earth Engine, segment Sentinel imagery using Orfeo Toolbox, and combine the two outputs using Reclass & Zonal Statistics.

This methodology takes a long time to run. Therefore, the outputs of each step are provided in the folder with these instruction so you do not need to run each step yourself.

Google Earth Engine

Random Forest Pixel-based classification

Example: Random Forest Classification of Utility-Scale Solar Fields with Sentinel-2A imagery using Google Earth Engine

Random forest is a robust machine learning algorithm used to identify target classes in satellite imagery. For this example, you will utilize a random forest algorithm to classify two classes in the Sentinel-2A imagery: Solar Fields and Non-Solar. The first step is to navigate to the state of Rhode Island in the Google Earth Engine Map Viewer.

This link provides a script for classifying solar fields in Rhode Island. You can save this script to your GEE account for future reference.

https://code.earthengine.google.com/b1e74885147478aab06089b44803bcb9

In addition, here are some reference links to GEE tutorials for this methodology and the developer page (third link):

1. https://geohackweek.github.io/GoogleEarthEngine/05-classify-imagery/
2. https://medium.com/@punwath/mapping-land-cover-with-google-earth-engine-ad80af4c34e6
3. https://developers.google.com/earth-engine/apidocs/ee-classifier-smilerandomforest

The script should:

1. Generates a training dataset consisting of 17 polygons of solar fields (including arrays and the dirt edges immediately surrounding the arrays) and 18 non-solar in Rhode Island.
2. The random forest algorithm should be trained for 1,000 iterations using the most valuable electromagnetic bands for detecting solar arrays: Blue, Near-Infrared, Shortwave Infrared-1, Shortwave Infrared-2. 80% of the samples should be used to train the algorithm and 20% of samples were used to test the algorithm.
3. Finally, this trained model is applied to the entire Sentinel-2A scene to produce a hard classification of solar fields and non-solar. The output from the random forest classification is a raster layer of the hard classification. **Solar has a value of 1 and non-solar has a value of 2.**

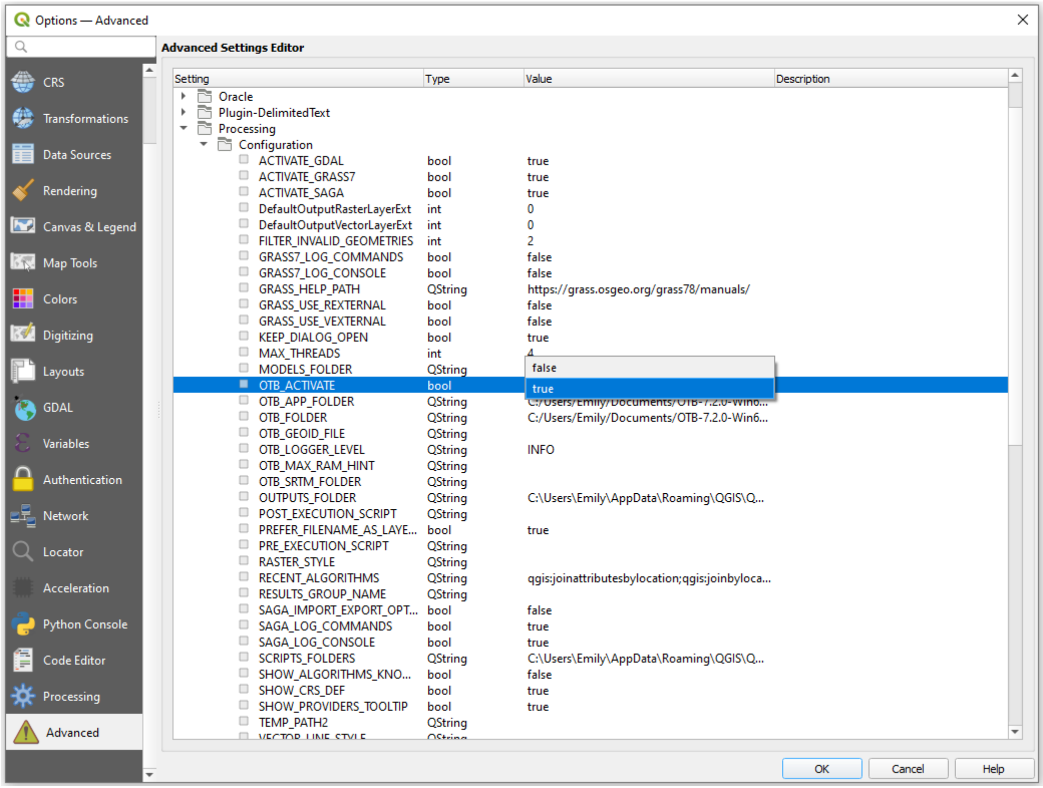
Open the Step1_RandomForest_RI_output.tif file to see the output from GEE.

QGIS

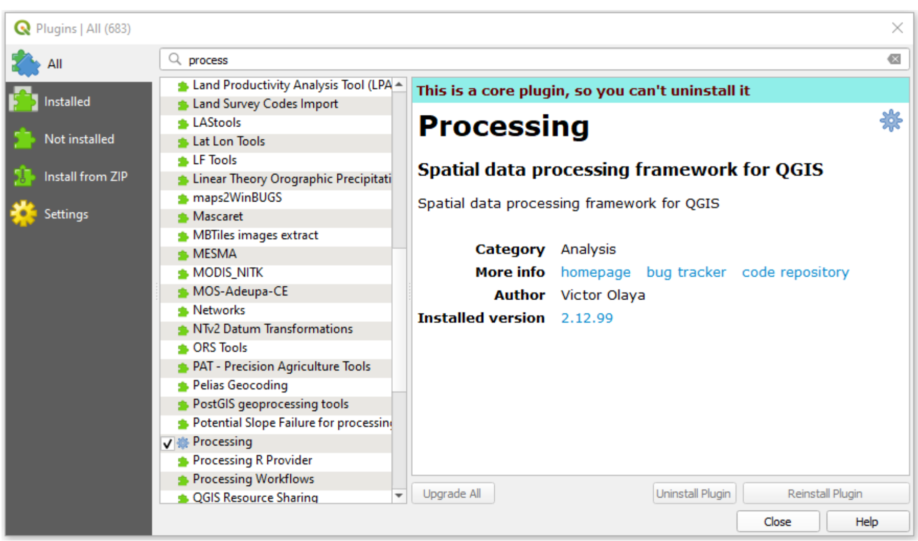
1. Installing Orfeo Toolbox

Downloaded Orfeo Toolbox here: **https://www.orfeo-toolbox.org/download/.** Unzip & extract toolbox in a known file location.

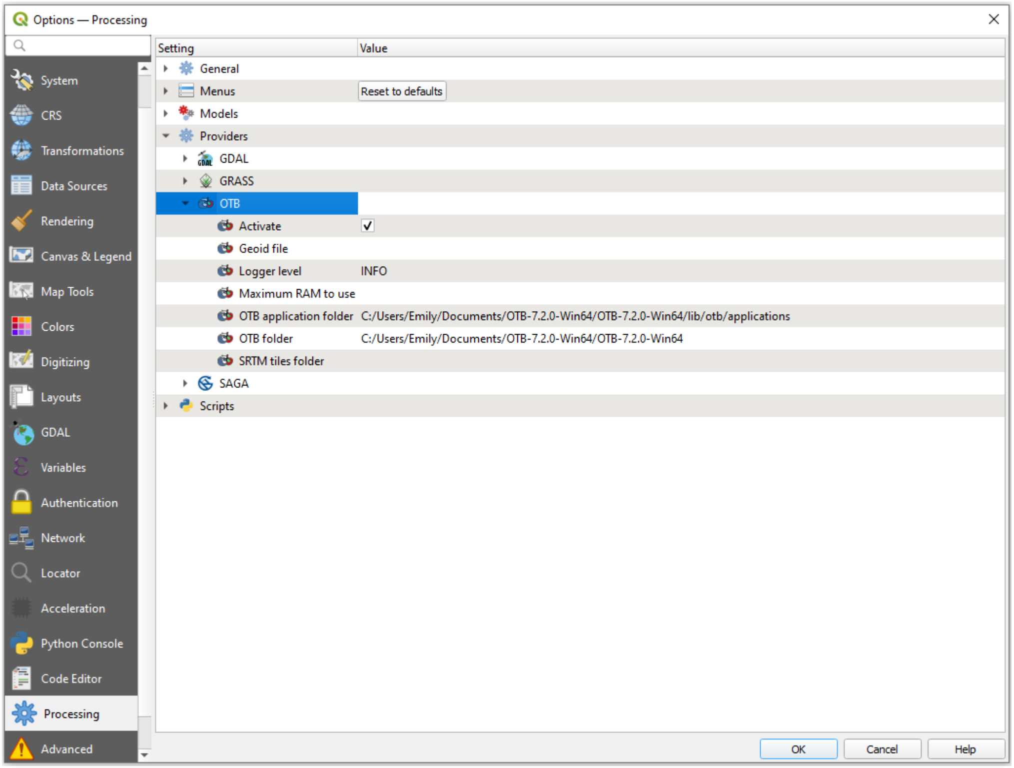
Open QGIS and navigate to **Settings > Options > Advanced.** Click “I will be careful, I promise!”. Navigate to **Processing > Configuration > OTB Activate**. Click on the cell that says “false”. Using the drop down menu, change the status to “true”.



Next, navigate to **Plugins > Manage and Install Plugins > All**. In the search bar, type "Processing". Activate the Processing plugin by clicking on the white box.



Finally, navigate to **Settings > Options > Processing > Providers > OTB**. Click on the white box next to "Activate". Double-click on the cell next to "OTB application folder" and navigate to the **"x/OTB-7.2.0-Win64/lib/otb/applications"** folder. Double-click on the cell next to "OTB folder" and navigate to **"x/OTB-7.2.0-Win64".** Press OK.



If you press **CTRL + ALT + T**, the processing tool box will appear in a panel on the right side of the screen. There should be a tab labeled "OTB" which houses all of the tools available in Orfeo Toolbox.

2. Image Smoothing for Preparation for Segmentation

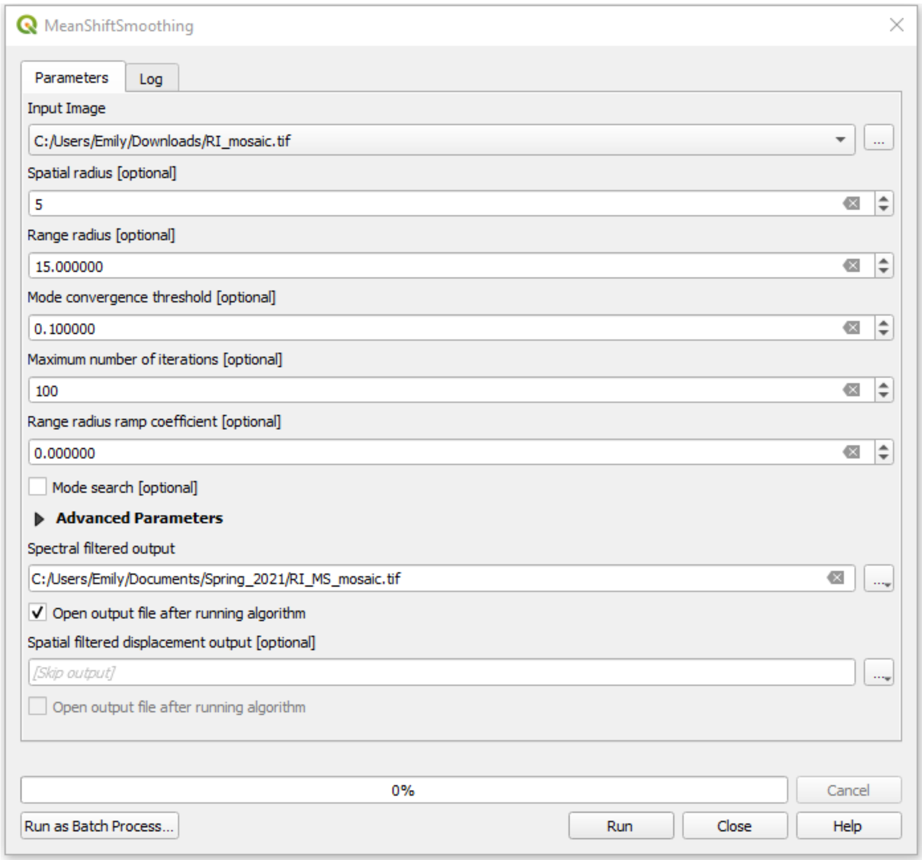
Semantic Segmentation is a method of analysis which divides an image into polygons based on pixel similarity. To prepare your satellite imagery for segmentation, you must first run the imagery through a mean-shift smoothing algorithm to limit the extreme values of outlier pixels.

Within the Processing Toolbox, navigate to **OTB > Image Filtering > Mean Shift Smoothing.** Double-click to open the dialogue box.

A full description of the mean-shift smoothing tool is here: https://www.orfeo-toolbox.org/CookBook/Applications/app_MeanShiftSmoothing.html.

Short description: Mean-shift smoothing averages the pixel values within a user-defined spatial neighborhood and a user-defined spectral neighborhood. By limiting pixel aggregation via distance and spectral signature, mean-shift smoothing can preserve the edges of image features while minimizing spectral noise within the features.

Change the input image to “RI_mosaic.tif”. Leave the smoothing parameters as the default values. In the “Spectral Filtered Output” parameter, navigate to your chosen folder and create a file name for the smoothed output TIF file. For the “Spatial filtered displacement output”, click on the “...” button and select “Skip output.” Press “Run”. This will take a bit to complete.



Open the RI_MS_mosic.tif to see the output from this process.

3. Segmentation of the Smoothed Image

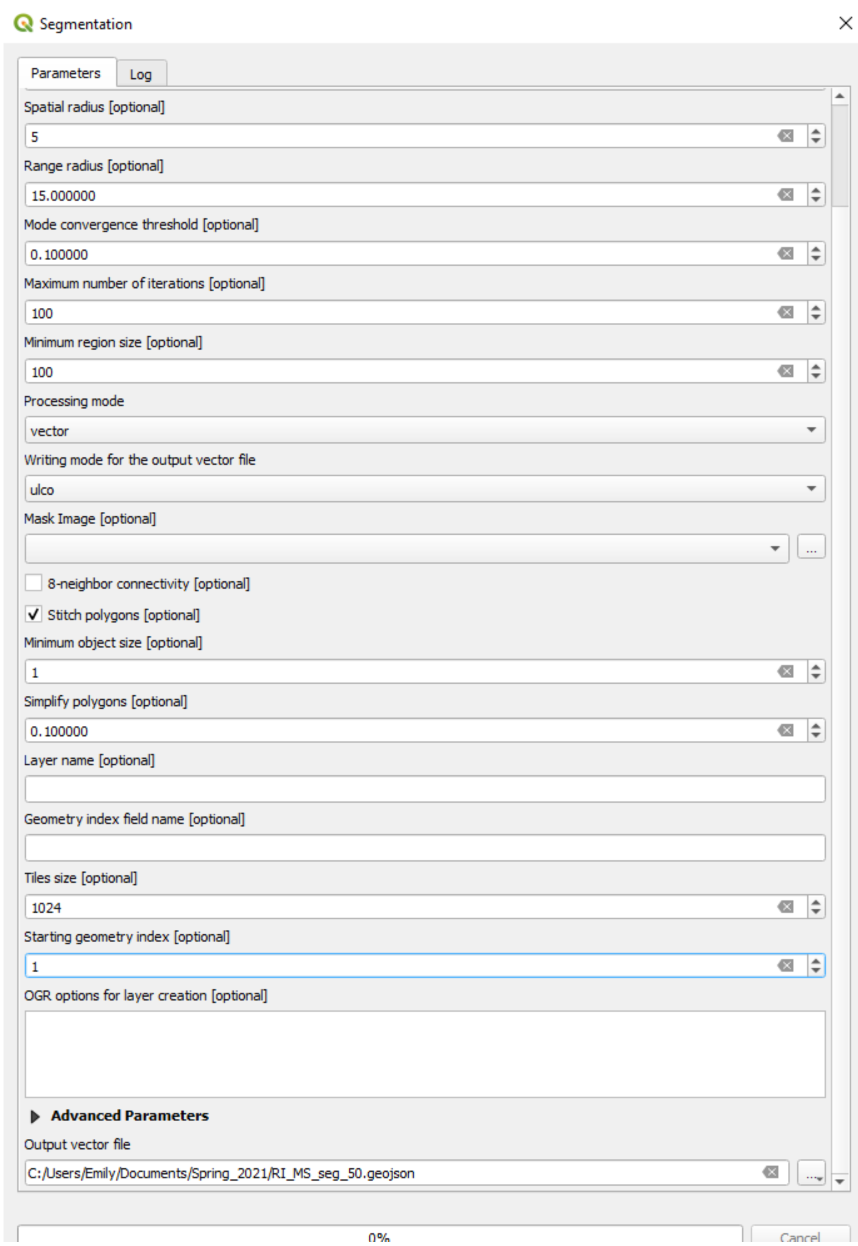
Now, the smoothed imagery will be fed into the segmentation application. A full description of segmentation tool is here:
https://www.orfeo-toolbox.org/CookBook/Applications/app_Segmentation.html

Short description: The segmentation tool contains several segmentation algorithms. We’ll be using the “meanshift segmentation” algorithm. Similar to mean-shift smoothing, the meanshift segmentation tool groups pixels into “segments” based on user-defined spatial and spectral parameters. In addition, the user can set a “minimum” segment size as **X** pixels. Therefore, each segment will be at least **X** pixels.

Navigate to **Processing > OTB > Segmentation > Segmentation**. Double-click on Segmentation to open the dialogue box.

Enter your smoothed TIF image as the “Input Image”. Leave the algorithm parameters as defaults. This particular run will produce segments with a minimum of 100 pixels.

Under the “Processing mode” parameter, select “vector”. In the “Output vector file” parameter, select a location and filename for the segmentation vector file. We need to set the file type for the output. The native file type for QGIS vector layer is an ESRI shapefile. Therefore, make sure to add **.shp** to the end of your file. This will take a long time to run.



Open the RI_Seg_50_layer.shp to see the output from this process.

4. Reclass & Zonal Statistics

We need to prepare the random forest TIF for input to the Zonal Statistics module by reclassifying the valus. If you remember, the Google Earth Engine script classified Solar as 1 and NonSolar 2. We want to reclassify NonSolar to a value of 0.

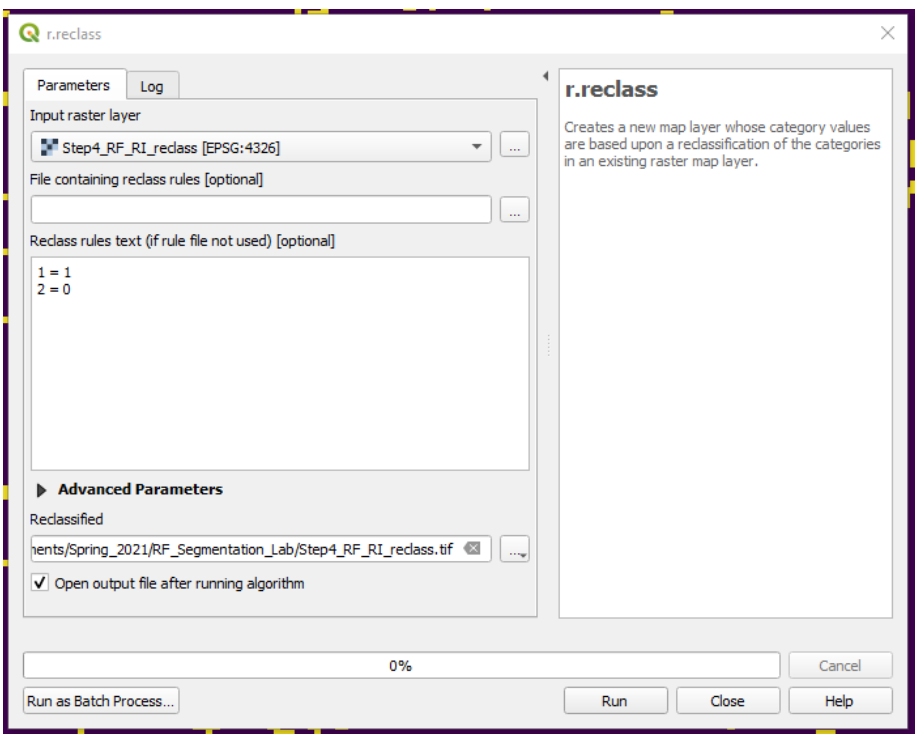
Search for "reclass" in the Processing Toolbox. Double-click on **r.reclass**. This tool is under the GRASS plug-in, an open-source toolbox for raster analysis.

If you have problems with the GRASS modules not loading, follow the suggestions in this link: https://gis.stackexchange.com/questions/313724/grass-not-working-with-qgis-3-6. I had to follow this step to get GRASS to work properly.

Go to `C:\Program Files\QGIS 3.12\apps\grass\grass78\` and copy the "bin" folder directly into `C:\Program Files\QGIS 3.12\bin\`.

Now the GRASS modules are in the specified path `C:\Program Files\QGIS 3.12\bin\bin`.

Once the r.reclass module is open, set the input raster layer to Step1_RandomForest_RI_output.tif. Enter the reclass rules: **1= 1, 2 = 0**. Select an output location and file name. Press OK.



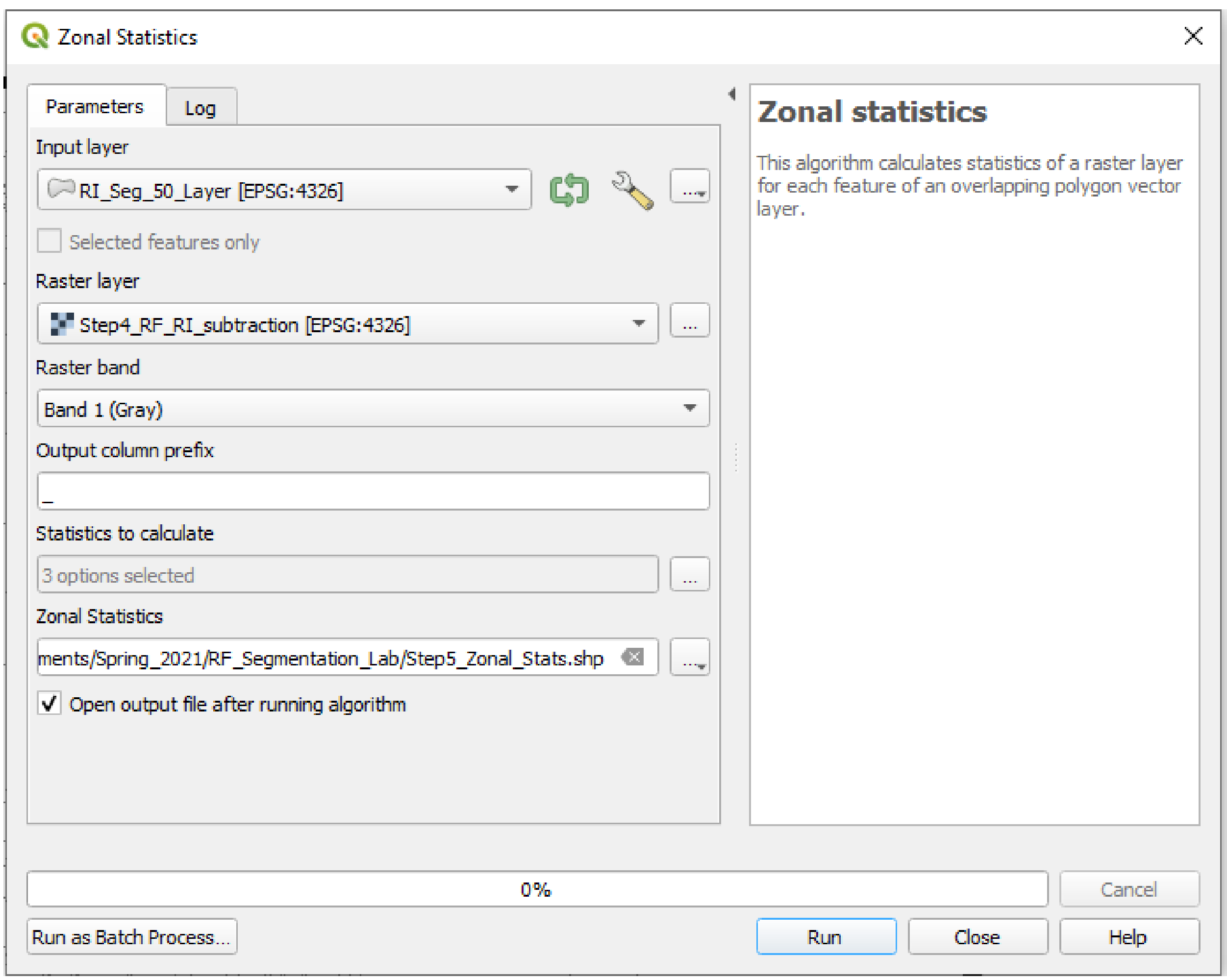
Open Step4_RF_RI_reclass.tif to see the output from this process.

Finally, Zonal Statistics allows us to combine the pixel-based classification from Random Forest with the vector segments produced via Segmentation. In this case, we will calculate the average number of pixels identified as “solar” within each segment.

In the Processing Toolbox and search for “Zonal statistics”. Double-click on the “Zonal statistics” tool.

As you input file, select the segmentation vector file (**RI_Seg_50_Layer.shp**). As the raster layer, select the raster calculator output (**Step4_RF_RI_reclass.tif**). Click on the  icon. For the “Invalid feature filtering” parameter, select “Skip Features with Invalid Geometries”. (Otherwise it will not run).

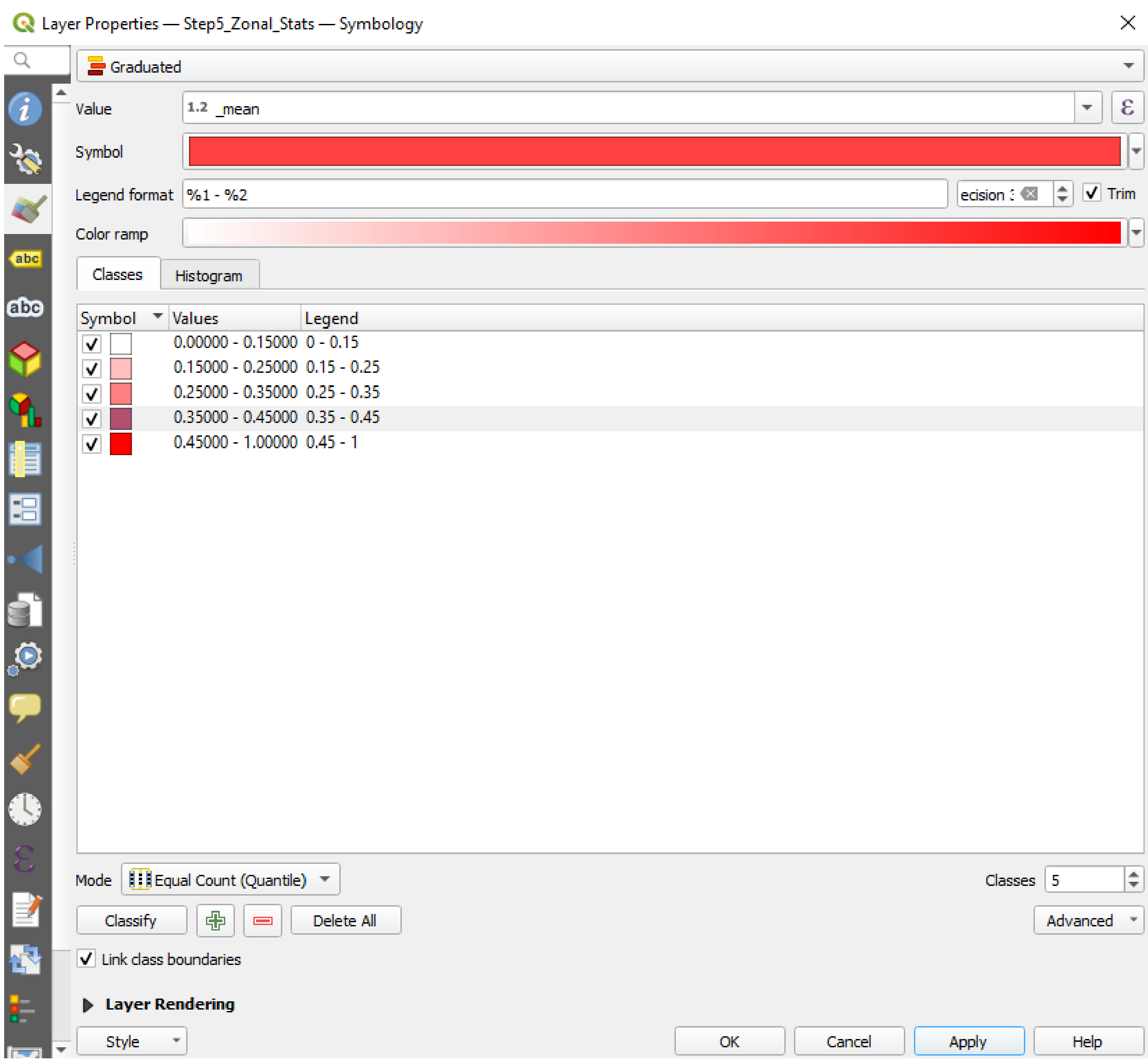
Finally, under the “Zonal Statistics” parameter, choose a location and name for your output file. You can select from many file types. I chose to save my file as an ESRI.shp file.



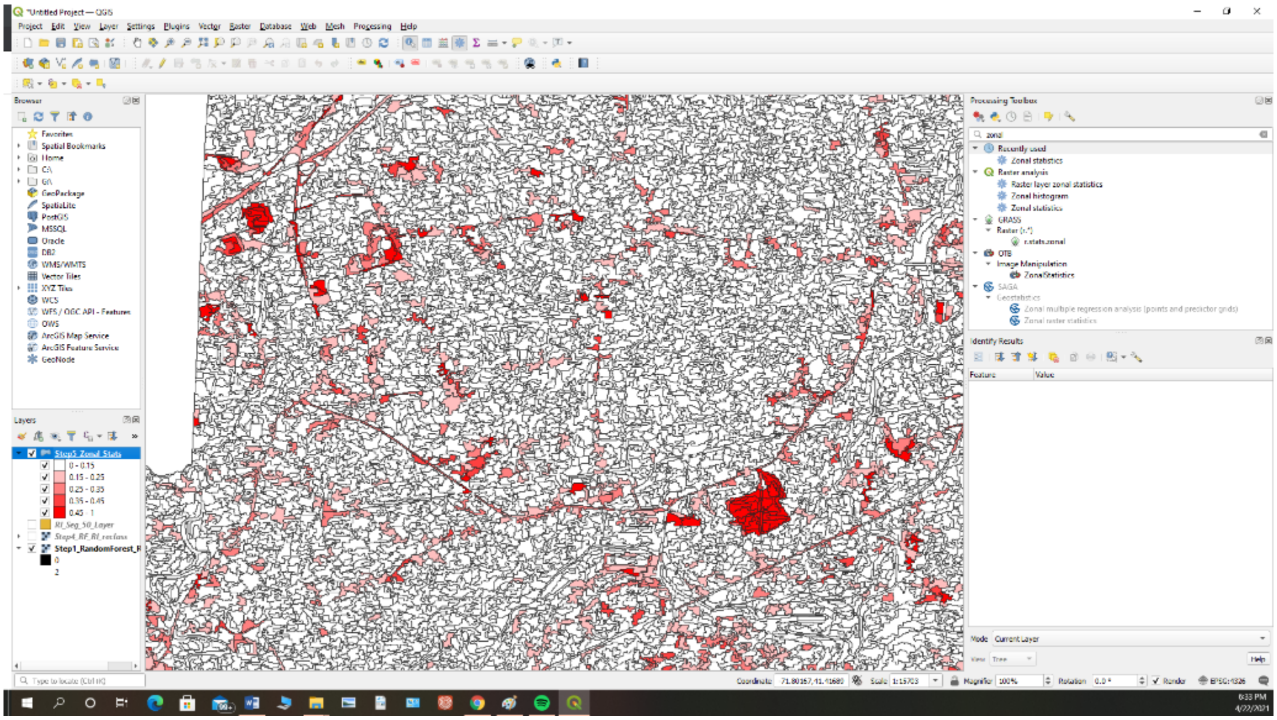
Open Step5_Zonal_Stats.shp to see the output from this process.

Display the “average solar value” by right-clicking on “Step5_Zonal_Stats” in the Layer panel and going to **Properties > Symbology**.

Change “Single Symbol” to “Graduated”. For the value, select “_mean”. At the bottom, click “Classify”, this will divide the “_mean” values you into 5 quintiles. Customize the breaks (see image below) to display the solar fields better.



Here is a zoomed-in example. While panning around, you’ll notice that roads and clouds receive large “_mean” solar values.



Extra Info: Now what?

This data product requires manual clean up. Clean up can be made faster by identifying threshold value to eliminate segments with a low probability of being solar.

In case, we eliminated all segments with a “_mean” solar value < .20. You can do this by choosing “Select by Attribute” and selecting all segments where _mean < .20, then press the DELETE key.

After that begins the painful process of manually verifying the output with the underlying Sentinel imagery. You can also use other imagery sources as a reference such as Google Maps. We used some high resolution imagery available through the University of Rhode Island.