Pontus Olofsson, Christopher E. Holden, Eric L. Bullock
Boston Education in Earth Observation Data Analysis/
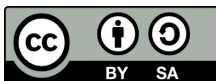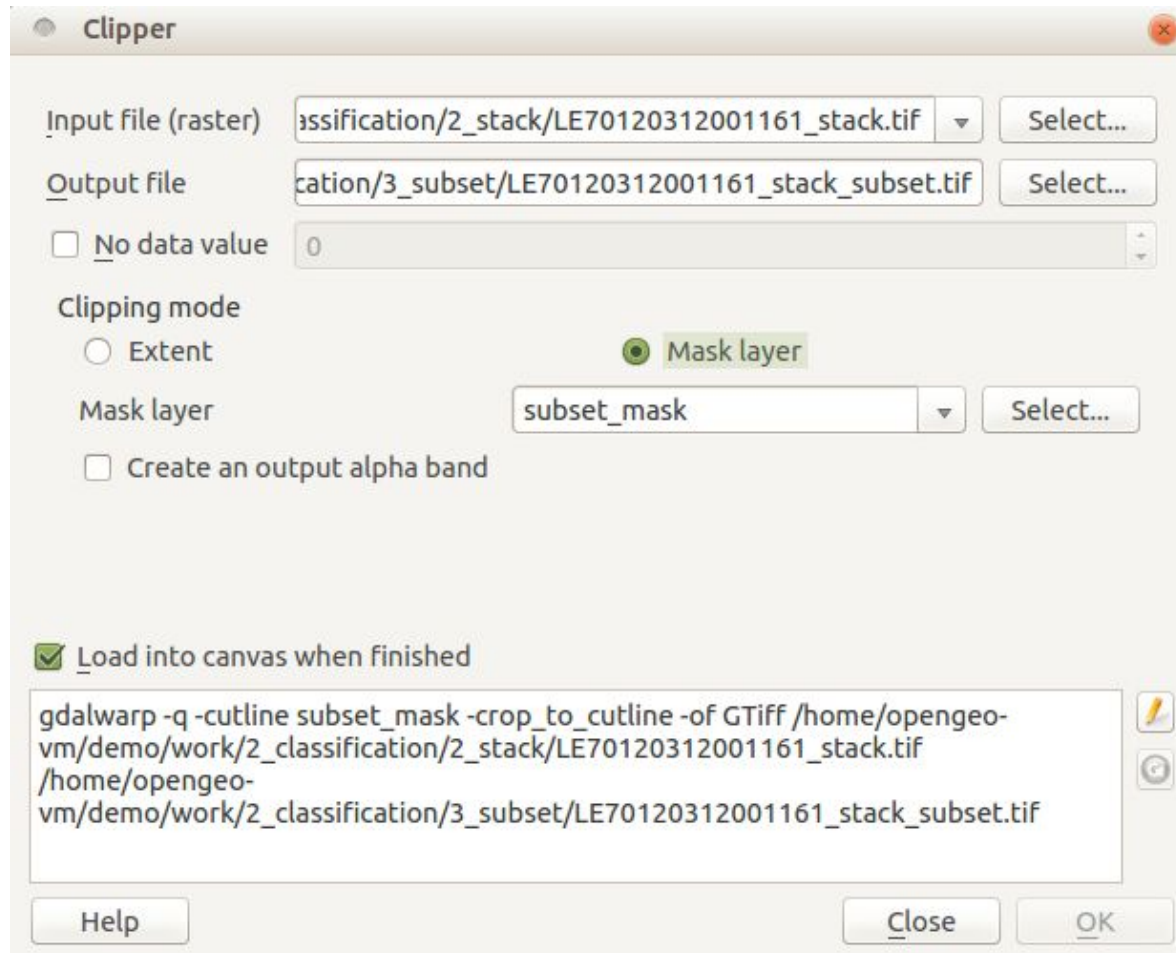*Department of Earth & Environment, Boston University*

S2. Methods: Classification

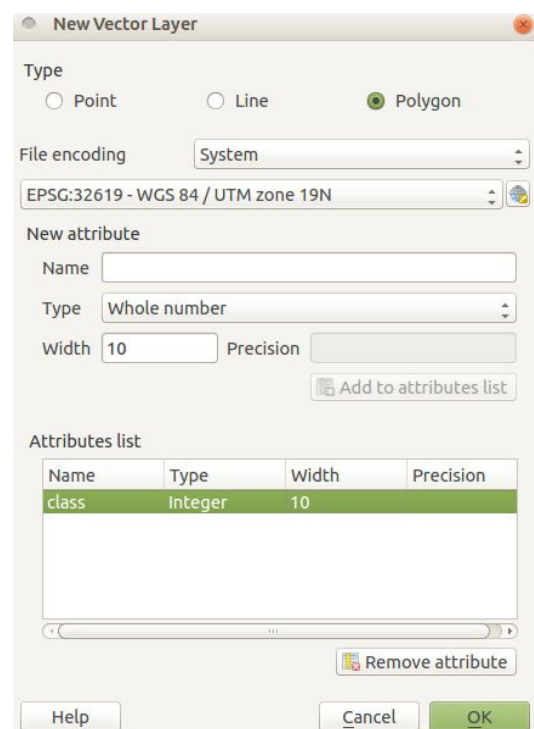## S2.1 Subset an image using *Mask layer*

1. Display the raster image that you want to subset in addition to the subset produced in the **Introduction.** If you have not created a single subset yet, create one by selecting *Extent* and *Clipping mode* and simply drawing a rectangle on the image canvas by clicking the mouse. This is detailed in **Tutorial 1 *Introduction* subsection 1.8**.
2. The subset raster will be used to create a 'clipper' vector. This vector can be used to clip all future images to the same extent. This is necessary for much of the analysis done in this tutorial.
3. Go to *Processing > Toolbox.* In the *Processing Toolbox* go to *Orfeo Toolbox > Geometry > Image Envelope.*
4. For the input image select the subset you made in the **Introduction.**
5. For *Output Vector Data* choose a filename and save it to your "/datasets" folder. This belongs here because you will likely use this file in multiple exercises.
6. Click *Run.*
7. To extract the subset using the vector layer, go to *Raster > Extraction > Clipper.* For *Input file (raster)* put the Landsat image you want to subset. For *Output file* click *Select…* and save your output as "[imagename]_stack_subset". For the **Introduction** exercise, it is recommended saving the subsets into the image subfolders in "5_composite".
8. Under *Clipping mode* select *Mask layer* and select the shapefile you just created. Click *OK.* (See screenshot below.)

Clipper

Input file (raster)   ssification/2_stack/LE70120312001161_stack.tif   ▼   Select...

Output file   cation/3_subset/LE70120312001161_stack_subset.tif   Select...

☐ No data value   0

Clipping mode
  ○ Extent   ● Mask layer

  Mask layer   subset_mask   ▼   Select...

  ☐ Create an output alpha band

☑ Load into canvas when finished

```
gdalwarp -q -cutline subset_mask -crop_to_cutline -of GTiff /home/opengeo-
vm/demo/work/2_classification/2_stack/LE70120312001161_stack.tif
/home/opengeo-
vm/demo/work/2_classification/3_subset/LE70120312001161_stack_subset.tif
```

Help   Close   OK

S2.2 Create your ROIs

1. Now that you have a subset to easily work with, use what you used in the previous exercise to apply an appropriate 3-band stretch on the image.
2. To collect ROIs in QGIS you need to create a shapefile with features corresponding to your different ROIs. As you did before, go to *Layer > Create Layer > New Shapefile Layer.* Under *Type* click *Polygon*. Specify the same coordinate reference system as the image you intend to classify. Now add an attribute to your Polygon with the *Name* 'Class', and *Type* 'Whole Number'. You can delete the other attribute in the *Attribute List* by clicking on the attribute with the name 'id'

New Vector Layer

Type
  ○ Point   ○ Line   ● Polygon

File encoding   System

EPSG:32619 - WGS 84 / UTM zone 19N

New attribute
  Name
  Type   Whole number
  Width   10   Precision

Add to attributes list

Attributes list

Name	Type	Width	Precision
class	Integer	10	

Remove attribute

Help   Cancel   OK

and clicking *Remove attribute*. Your box should be filled out like the screenshot above. Click *OK* to create the shapefile. It will ask you where to save the ROI. Save it to "/2_classification/4_training".

3. Start drawing your ROIs by right clicking the newly created shapefile in the *Layers* panel > *Toggle Editing*; click the *Add Feature* button (or *Layers > Add Feature*); left click in the image to draw a polygon, and right click to complete the polygon.

4. When you finish drawing an ROI, it will ask you to fill in a value for 'Class'. Each class in your map will have a corresponding number. If you have 5 classes, then your classes will be labeled 1-5. Make sure to keep track of which number corresponds to which class. To the right is an example of an ROI for a water class.

5. After you enter the number for the class, click *OK.* Repeat this multiple times for each class. **When collecting ROIs, pay attention to the notes below.**

6. When you are done creating ROIs, right click on your shapefile in the *Layers* panel and click *Toggle Editing* to stop editing and save your edits.
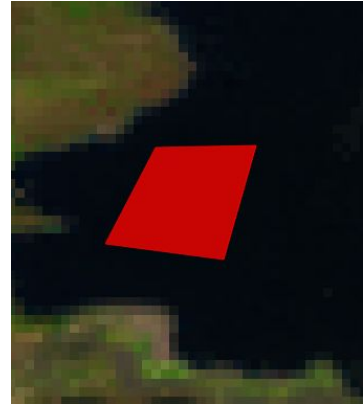
NOTE 1: You can have more than one polygon that represents the same class. In fact, you should typically have many polygons for each class! Many small ROIs are better than a few large ones (a few hundred pixels tops for each ROI). This can help account for the spectral variations in each land cover class. If you provide just one example of forest, it will be hard to recognize the variety of spectral signatures associated with that class. If you have multiple sub-classes under the same class (such as different tree species all being considered forest), your ROIs should reflect the full breath of possible spectral signatures.

NOTE 2: You will find that you will need to adjust your training data a number of times (maybe many times!) in order to produce a good quality classification. The actual classification part is just pressing a button, what reflects the accuracy of your results is the quality of your training data. Make sure to take your time when collecting ROIs.

NOTE 3: The amount of ROIs for each class should be proportional to the total area of that class in the image. If you have more forest than water, then you should have more ROIs for forest than you have for water.

S2.3 Analyze your ROIs

1. To visualize the spectral signatures of your training data ROIs, load the *ROI Explorer* plugin within QGIS. Open the *Plugins* menu and click *Manage and Install Plugins*. Under the *Installed* tab, enable the *ROI Explorer* plugin. If this plugin is not currently installed, you may need to install it from the *Available* tab.
2. The user interface of the *ROI Explorer* plugin should appear docked in the right side of QGIS. If this user interface did not appear, right click on any QGIS toolbar and show the user interface by clicking the *ROI Explorer* dropdown item.
3. The *ROI Explorer* user interface tracks all raster and vector data opened within QGIS. Select from the *Raster* drop down box the raster image you want to use for your classification. Using the *Vector* drop down box, select the vector layer containing your training data regions of interest. Use the *ROI Label Field* drop down box to select the field (column) within your training data vector file that contains the training data labels. This *ROI Label Field* selects the field that is used to aggregate features into class labels.
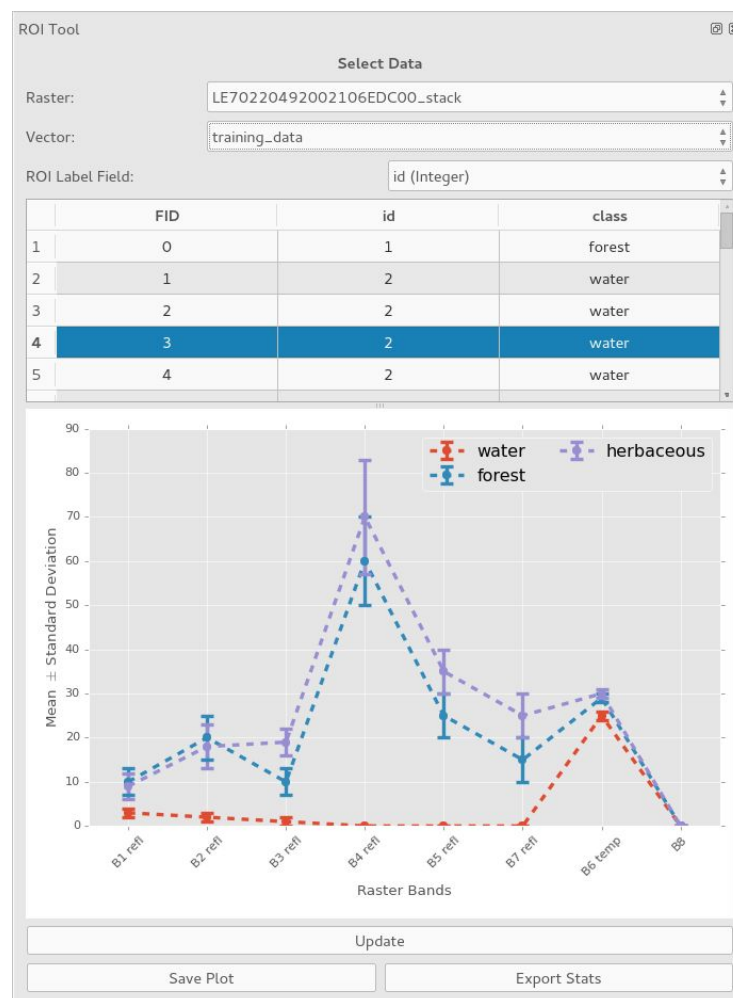
4. Now that your raster and vector files are selected, you may view each training data feature within the included attribute table. Selecting features from this attribute table will select them for visualization within the plot. You may select more than one feature by using the Control key and clicking additional features, by using the Shift key to select a range of features, or by using the Control + A shortcut to select all features.
5. Click the *Update* button below the plot to calculate the mean and standard deviation of all raster bands for each unique class label. The plot will update and display the mean as a point on the plot with the error bars above and below the point representing plus or minus one standard deviation.
6. Continue to analyze your ROIs for separability. Highlighting ROIs from different classes in the attribute table provides an understanding of the across class variability.
7. You might also want to analyze the within-class variability of your ROIs. You might create another field within your vector layer that provides additional class information, such as "degraded", "growing", or "mature" labels for examples of "forest', or simply provide a unique identifier for each example of "forest". If you select this additional field within the *ROI Label Field* drop down box, then you could analyze the differences among these example forest sub-classes by highlighting only features labeled "forest" and by clicking *Update*.
8. You can export the currently displayed plot to a variety of image formats (PNG, JPEG, EPS, etc.) using the *Save Plot* button. A dialog will appear allowing you to specify the filename and file format. Click *Save* to save your plot. Additionally, the computed statistics for the currently plotted classes may be exported as a CSV file using the *Export Stats* button.

S2.4 Train the classifier

1. Make sure that the image you intend to classify and the vector file containing the ROIs are added to the *Layers* panel.
2. To find the tool needed, go to *Processing > Toolbox*; in the toolbox that pops up on the right side of the screen double click *Orfeo Toolbox > Learning > TrainImagesClassifier(rf).*
    a. **Note:** Orfeo has many different types of machine learning classification algorithms. The technique can be changed by choosing a different classifier to train in this step.
3. For *Input Image List* select your Landsat image. For *Input Vector Data List* select your ROI shapefile.
4. Make sure the *Name of the discrimination field* is the name of the attribute you used to differentiate the classes (it should be 'Class').
5. For *Output Model*, click *Save to file…* and save it as a ".txt" file in "/2_classification/4_training/".
6. For now, leave the other fields as they are. These are different parameters for what goes into the Random Forest classification. If you'd like, you can play around with the different parameters and see how they affect the end results. When doing so, you will

need to retrain the classifier with new output models. To train the classifier click *Run.*

2.5 Classify the map

1. To find the tool needed for classification, return to the *Processing Toolbox* and go to *Learning > Image Classification.*
2. For *Input Image* select your Landsat image; for *Model file* select the text file you created in the step above; and for *Output Image* select *Save to file…* and save to "/2_classification/5_map/". Leave the rest of the fields as they are. Your *Image Classification* box should look something like the image below. Select *Run.*
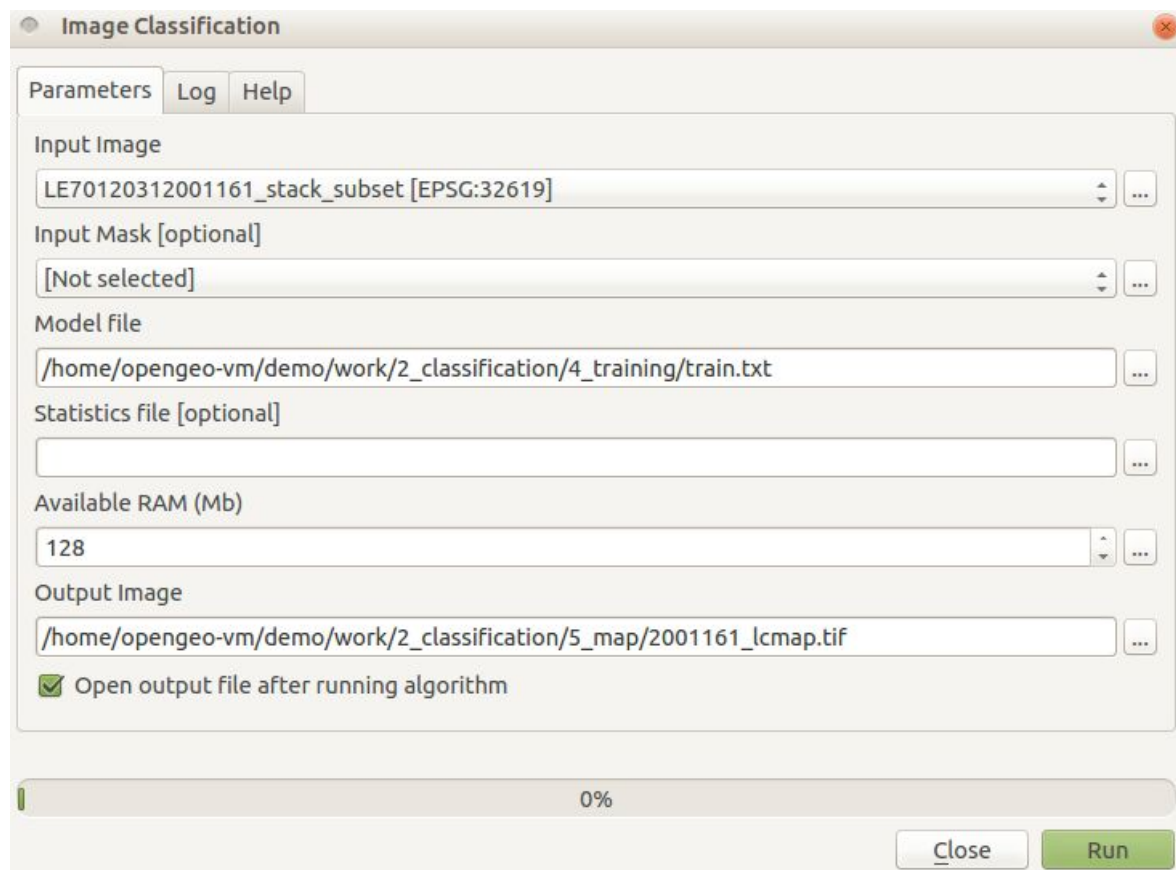
3. Right click the map in the Layers panel > *Properties* > *Style* to change the name and color of the classes. Most likely, the map will not be satisfactory and you will need to go back to step S2.2 to revisit and edit your ROIs. Remember, image classification is iterative trial and error process and you might have add/remove/edit your ROIs several times before you have a map that you are satisfied with.

S2.6 Map smoothing/clean-up

1. Open up the map and apply a single-band stretch.
2. In the general QGIS toolbar, go to *Raster > Analysis > Sieve.*
3. For *Input file* select your classification image.
4. For *Output file* save your file to "/2_classification/5_map/".
5. For *Threshold* put 10. This defines the minimum size of pixel groupings. Adjust this if you would like to remove larger or smaller groups.
6. For *Pixel connections* select 8. Press *OK* to run the sieving process.
7. Notice a change in your classification map? It should look less noisy. Below you can see a subset of a classification unsmoothed (left) and smoothed using the sieving process (right).
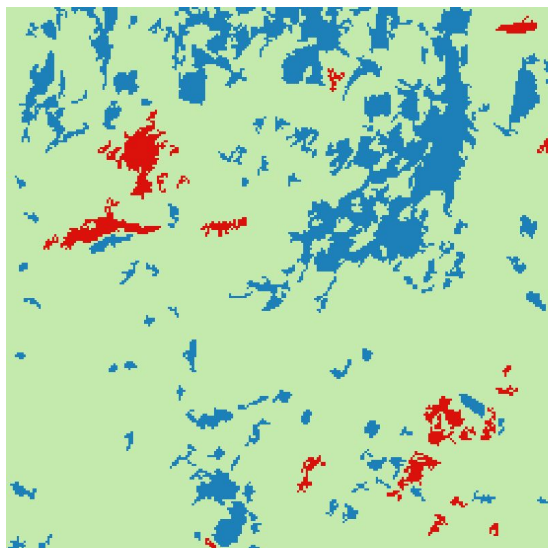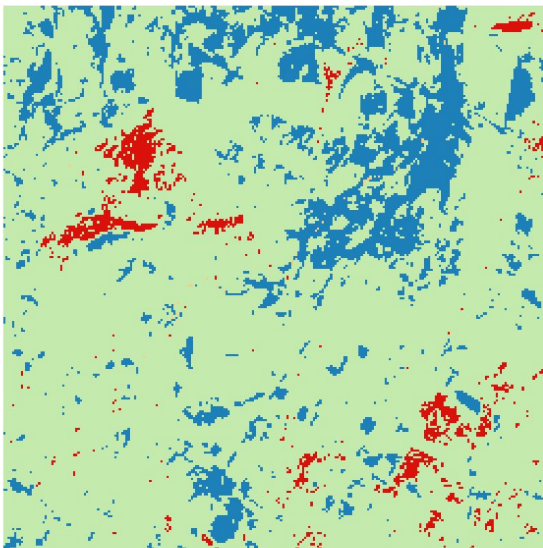
S2.7 Image Segmentation

1. Be sure that you have the subset image open in QGIS**.** Since image segmentation is a computationally expensive process, it is best to first work with a small subset to test the parameters you will use. At most, start with an image that is a few thousand pixels. If your image is significantly larger that, refer to **Method S2.1** to make a smaller one. This will save you significant time in the long run. Once you have determined appropriate parameters then segmentation can be run on the full scene.
2. In the *Processing Toolbox,* go to *Orfeo Toolbox > Segmentation > Segmentation (meanshift).*
3. Under *Input Image* select the subsetted image.
4. You might have to tweak the parameters a bit to get the results you are looking for. Here are some suggestions:

a. *Spatial radius*: the size of a window (in pixels) being considered in mean calculation; to start leave this at 5.

b. *Range radius:* specifies how close, in terms of euclidian distance, neighboring pixels need to be to be grouped together; to start, try keeping it at 15.

c. *Mode convergence threshold:* Neighboring pixels whose multi-spectral distance lies below this threshold with converge into one mode. Leave this at .1.

d. *Maximum number of iterations:* Algorithm will stop if convergence has not been reached at this amount of iterations. This can impact computation time. If you want to speed up the process at the expense of segment quality, decrease this value. To start, keep this at 100.

e. *Minimum region size:* This is intuitively the smallest an output segment can be. This will depend on the types of regions you are trying to find. Increase this to 150.

f. *Tile size:* Since the algorithm is so computationally expensive, it will break up in the image into specified tiles to reduce the memory needed. This can result in the output segment image looking 'blocky' since segments are not accurately calculated at the lines between the segments. This will often not matter, but if the blocks are jeopardizing your results, increase this value. If you are running into memory issues, decrease this value. To start, keep it at 1024.

g. Save the *Output vector file* to "/2_classification/5_map/".

h. Keep everything else the same, and click *Run.*

5. Depending on the size of your input file, this process can take a while. Once it is done, you should get something like the following image, the red being the output segment vector overlaid on the original Landsat image (to display the segment outlines: right click the shapefile in *Layers* panel > *Properties* > *Style* > *Simple Fill* > set *Symbol layer type* to *Outline: Simple line*):
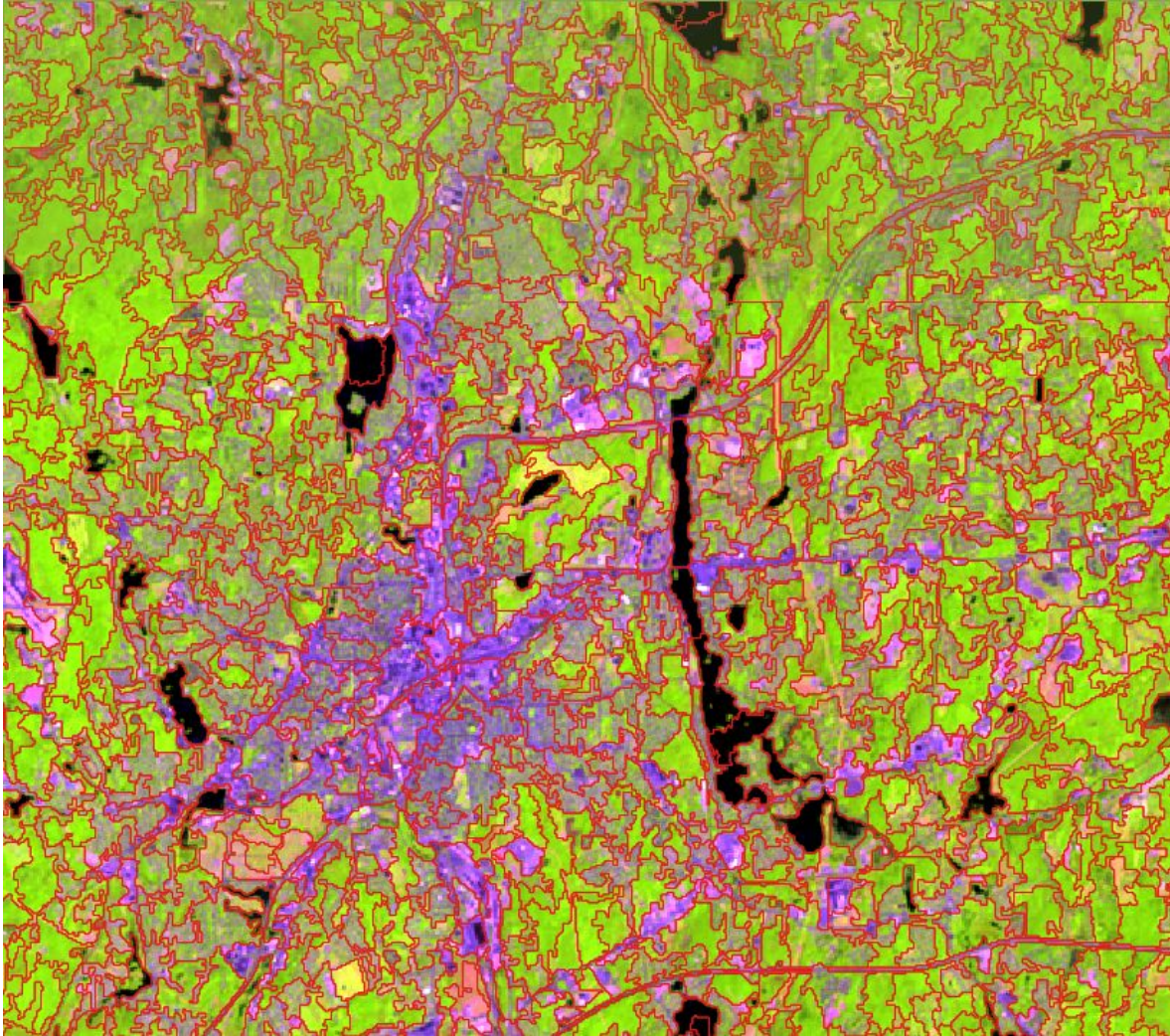
6. For visual analysis, this alone can be useful. One use of image segmentation is to identify spectrally varying regions in a normally homogenous land cover. Through visual interpretation, this can help detect areas of change.
7. For classification, the vector file needs to be converted to a raster. Using zonal statistics, a raster can be created that contains the means and variance of the pixels within each segment as the output bands. This 'object' image can then be classified.
8. To calculate mean and variance for the objects in your image, we have written a script that must be ran outside of QGIS on the command line.
9. Navigate to "/2_classification/5_map/" in the *MATE Terminal*. After specifying the script through 'object_stats.py' you enter the Landsat image, the segment vector, and then the statistics to be calculated (mean and variance in this case):

```
object_stats.py ../3_subset/LE70120312001161_stack_subset.tif
2001161_segmentation.shp 2001161_seg_stats.tif mean var
```

10. This will generate a multiband image where each segment contains the mean and variance for the pixels in that segment in the following band order:

        1. Mean Band 1
        2. Variance Band 1
        3. Mean Band 2
        4. Variance Band 2
        etc…

This raster can now be used in an object-based classification approach; display it as false color composite of choice, it should look like below with the original Landsat image on the left and mean segment image (5-4-3 color composite):
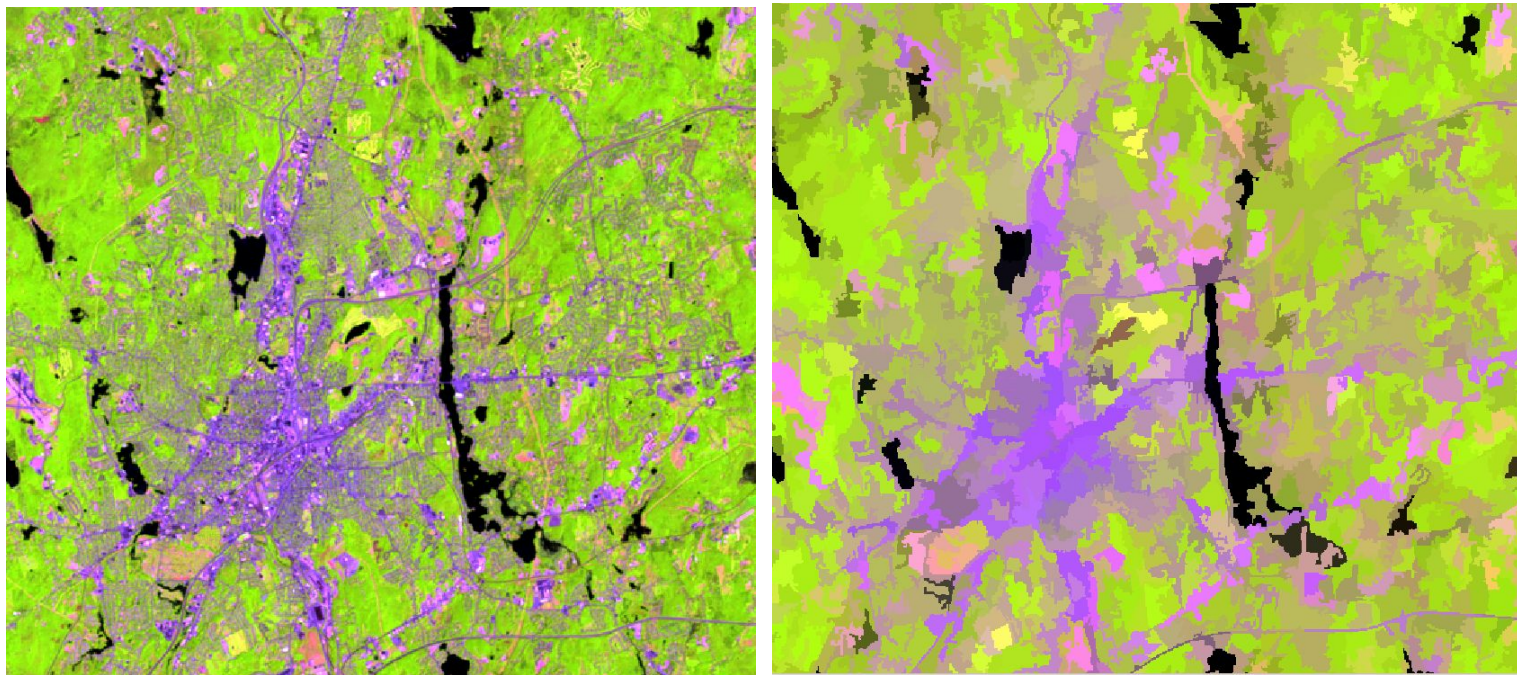
S2.8 Map Clean-up Through Segmentation

1. Return to the *MATE Terminal* and navigate to "/2_classification/5_map".
2. Once again, we will use the object_stats.py script. This time, instead of extracting statistics from the Landsat image use the original (non-segmented) classification map from step **S2.5.**
3. The segmentation vector will again be used, but this time specify 'mode' as the statistic. What this is doing is calculating the classification that has the highest likelihood of being assigned inside that segment and applying that classification to the entire segment. This has the effect of "smoothing" out the classification map based on calculated segments. An example command would be:

```
object stats.py 2001161 lcmap.tif 2001161 segmentation .shp
2001161 lcmap seg.tif mode
```

which will generate a "clean" map like below to the right, with the original unsegmented

map to the left (note: if this removes too much detail, you will need to redo the
segmentation and decrease the *minimum region size*):