Marcel Rodriguez-Riccelli
2/18/2026
Satellite Image Analysis (GEOG 581)
LAB 5 - SUPERVISED CLASSIFICATION PART I

LAB DAY: THURSDAYS
LINK TO CODE: GOOGLE EARTH ENGINE

QUESTION 1. In your own words, what is our filtering strategy for identifying this image that we want to classify? This image isn't cloud-free, but how have we set up our filtering to ensure minimal cloud cover (Hint: look up what .mosaic() does)?

To understand the filtering strategy for our image, we can break down the following chain of functions which determines which image is selected from the Landsat 8 collection:

l8.filterBounds(nile_delta)

> Filter for imagery that contains the bounds contained in our *nile_delta* Multipoint type object. Our import dropdown tells us that nile_delta contains the coordinate points: 0: [31.2066650390625,30.562260950499443], 1: [30.9814453125,31.3864682695423], 2: [32.0635986328125,30.93992433102344].

.filterDate('2016-05-01', '2016-07-30')

> Filter for imagery captured between May 1st, 2016 and July 30th, 2016.

.sort('CLOUD_COVER_LAND',false)

> After the previous two filter criteria have been applied, sort the resulting remaining collection by the CLOUD_COVER_LAND variable for each image contained in the Landsat 8 metadata. The Image Properties for Landsat 8 list this as representing *"Percentage cloud cover over land (0-100), -1 = not calculated."*

.mosaic())

> This function composites all of the images remaining in the collection, by assigning each pixel from its first non-masked occurrence based on the order of images in the collection.

.select(bands, rename_bands)

> Select and rename the bands based on our previously defined array of strings:
>
> *var bands = ['SR_B2', 'SR_B3', 'SR_B4', 'SR_B5', 'SR_B6', 'SR_B7']*
> *var rename_bands = ['blue', 'green', 'red', 'nir', 'swir1', 'swir2']*

So, our filtering strategy is to first filter by our bounds and timespan of interest, sort by cloud coverage, and then compile a "mosaic" where each pixel selected is from the image with the least cloud cover for which that pixel is unmasked.

QUESTION 2. Looking at the `nile_img` for a moment, what concerns do you have about classifying this image into a land cover map?

Despite the mosaic function based on least cloud cover, we can still see some atmospheric constituents over land that will most likely result in non-intrinsic discrepancies in perceived surface reflectance. *(Fig. 1A)* We can also see some artifacts from the mosaic process where reflectance values change abruptly along this regular line, which appears to be the seam between two mosaiced images with varied atmospheric conditions. *(Fig, 1B)*

Fig 1. Artifacts in the "nile_img" mosaiced Landsat 8 image. 1A (Right): Haze overlaying the land made apparent by the coastal boundary. 1B (Left): An artifact of the mosaic function, possibly the seam between two images with varied atmospheric conditions.

QUESTION 3. Please visually compare nile_img with the reference satellite image basemap within the EE API by using the "Satellite" tab next to the "Map" tab in the upper right of the map view. What differences do you see between your nile_img and the satellite image basemap with regard to how the landscape is represented (i.e., image stretching and coloring) and legitimate differences in land cover or condition?

The satellite image basemap is of a much higher spatial resolution than *nile_img*, and seems to have a much higher dynamic range across all color channels. The area seems to have become significantly more developed—throughout the region, coastal management structures, roads, and agricultural plots have sprung up in the time between ranges from which the two mosaics were compiled.

QUESTION 4. For the remaining two new classes of *barren* and *river*, repeat the steps above so that each layer has 25 points at sites corresponding to a given land cover. Make sure to assign the correct 'landcover' property value for each land cover type as well.

I've assigned 25 points at sites for each added land cover type, made them of the type FeatureCollection, and assigned them each their respective value. *(Fig. 2)*

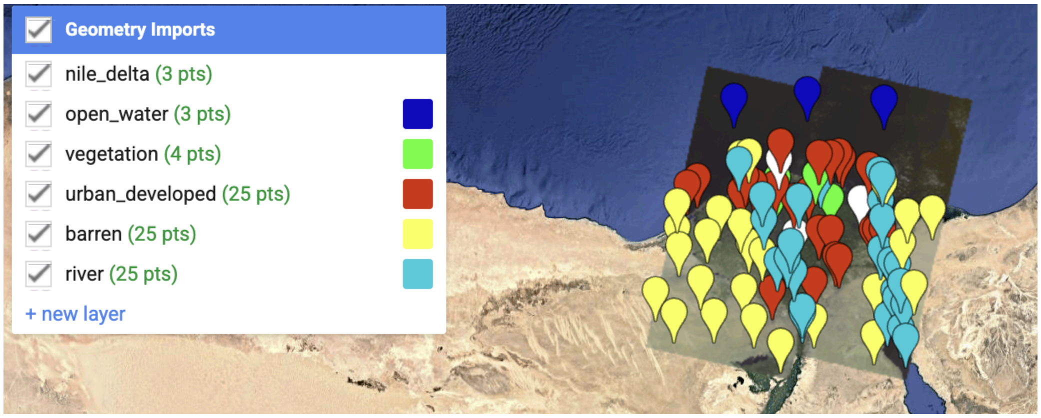

Fig. 2. Training data points for land cover types over the Nile Delta area of interest.

QUESTION 5. Are you including any sites at the 'edge' of a given land cover? Why or why wouldn't the inclusion of such 'edge' sites be beneficial for the classification?

I tried to include a bit of diversity by choosing from different sites of a single landcover, and differently colored pixels within a single feature extent (such as greenish and blueish colored sections of a river), but did not include any sites at the edge of a given land cover. This is because pixels at the edge of a land cover feature can integrate over two different types of land cover if they transition within the spatial extent of a single pixel, and would therefore muddy the spectral separation of our landcover classifications.

QUESTION 6. For the two original layers of *open_water* and *vegetation*, please add additional sites to these layers so that you have the same number of sites (25) for each land cover.

I've got 25 points at separate sites for each land cover type.

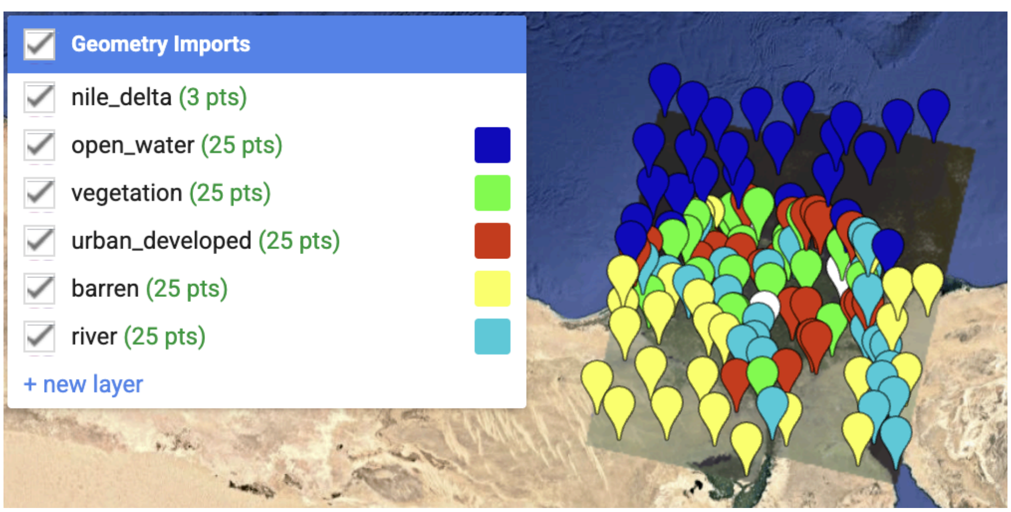

Fig. 3. Training data points for land cover types over the Nile Delta area of interest.

QUESTION 7. In your own words, what does ee.Image.pixelLonLat() do? What information does it add to an Image, and where is this information stored within the Image?

ee.Image.pixelLonLat() generates two new bands, where the pixel values are the longitude and latitude of each pixel from the input image for each new band respectively.

QUESTION 8. What are your values for each of the 'missing' variables above: input_img, training_pts_collection, and scale_value?

I filled the missing variables accordingly:

var input_img = nile_img_vars; // <-- what value goes here?

This is the image containing our image with additional NDVI, NDWI, Latitude, and Longitude bands.

var training_pts_collection = training_pts; // <-- what value goes here?

This is the FeatureCollection of our merged land cover type point FeatureCollections.

var scale_value = 30; // <-- what is an appropriate spatial scale for your image sampling?

I chose the spatial resolution of Landsat 8 for my spatial scale value.

QUESTION 9. Print out your training_extract and training_pts featureCollections to the Console window. Given this information, compare training_extract to training_pts. What are the differences in individual feature properties within the two featureCollections? How are the geographic locations of the features in training_extract represented?

The individual feature properties inside of *training_pts* lists the land cover property value we had assigned before, while the training_extract feature properties list values of each band in *nile_img_vars*. In *training_extract*, the longitude and latitude coordinates are members of the feature properties key, as opposed to the geometry coordinates key as in *training_pts*.

QUESTION 10. What does the 'null' value refer to in our Feature creation? (Hint: look at the arguments for ee.Feature() in the Docs tab)

The null value for ee.Feature is input for the geometry argument, meaning that the resulting Feature will be purely data instead of having an attached geometry with spatial attributes. This is possible because we've converted our spatial coordinates to Feature properties, so geometry will not be needed for geolocation.

QUESTION 11. Which class' training points tends to have the highest NDVI values? What class' training points have the lowest mean NDWI value? Do these differences align with your expectations of the NDVI and NDWI of each land cover?

We can see that the vegetation land cover has the highest mean NDVI value and the lowest mean NDWI value. *(Fig. 4)* This does align with expectations, because healthy vegetation generally has high reflectance in the near infrared band, represented as a positive operand in the NDVI formula and negative operand in the NDWI formula.

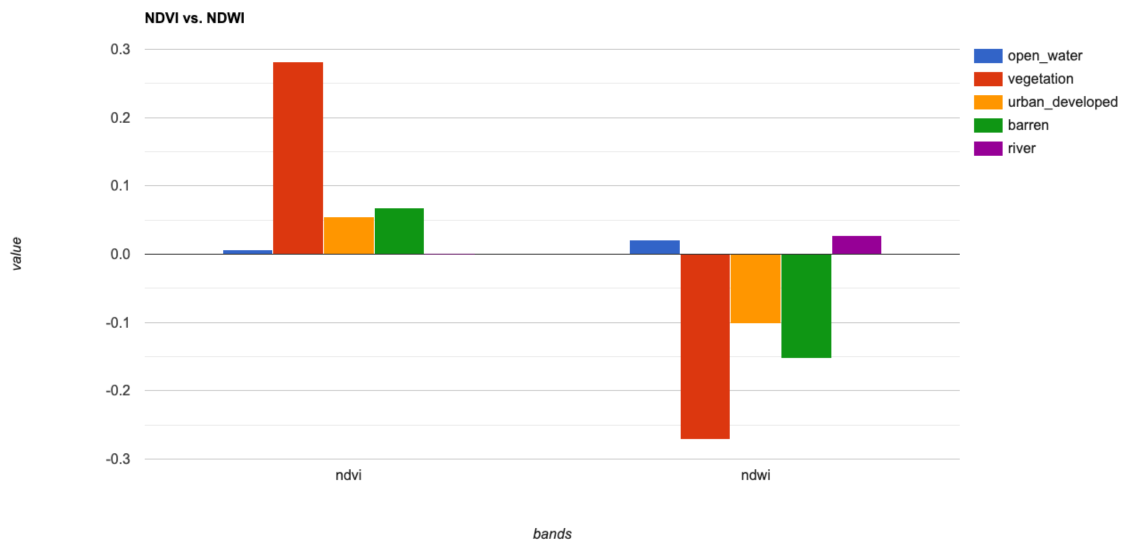

Fig. 4. A graph of NDVI and NDWI values for land cover points of each type.

QUESTION 12. In your opinion, what makes a variable/band useful for discriminating land cover classes? Which land cover types do you find that NDVI is useful for? How about NDWI?

High separation of values between land cover types makes a band useful for discriminating between land cover. From the graph, it appears that both NDVI and NDWI will do well for differentiating between vegetation and everything else, and aquatic bodies from terrestrial bodies in general, but might struggle to differentiate between urban developed and barren, and open water and river.

QUESTION 13. How many nodes are in the decision tree? (Note that the maximum node number may not equal the total number of nodes, and that there are two branches per node in a decision tree.) After the root node, on what spectral variable does the decision tree split into two branches? And at what spectral value does this split occur?

There are 20 nodes in total, ten decision nodes including the root node, and ten class nodes. *(Fig. 6)* The root node splits on the near-infrared band at a threshold of 11709, dividing pixels into those with a spectral value in the NIR band ≤ 11709 and those with a spectral value in the NIR band > 11709.

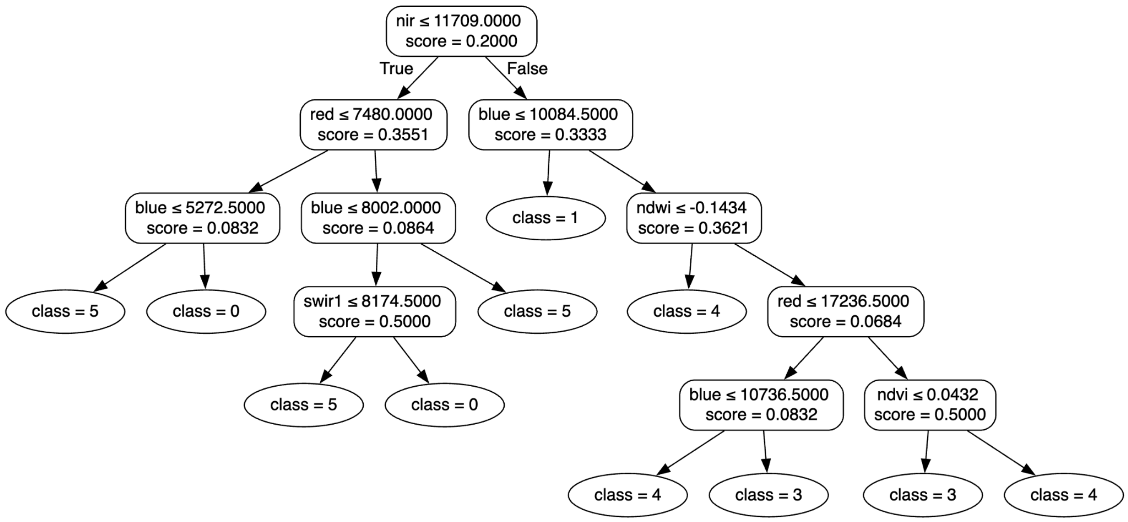

Fig. 6. Decision tree for our supervised land cover classifier.

QUESTION 14. Include this map in your assignment hand-in.

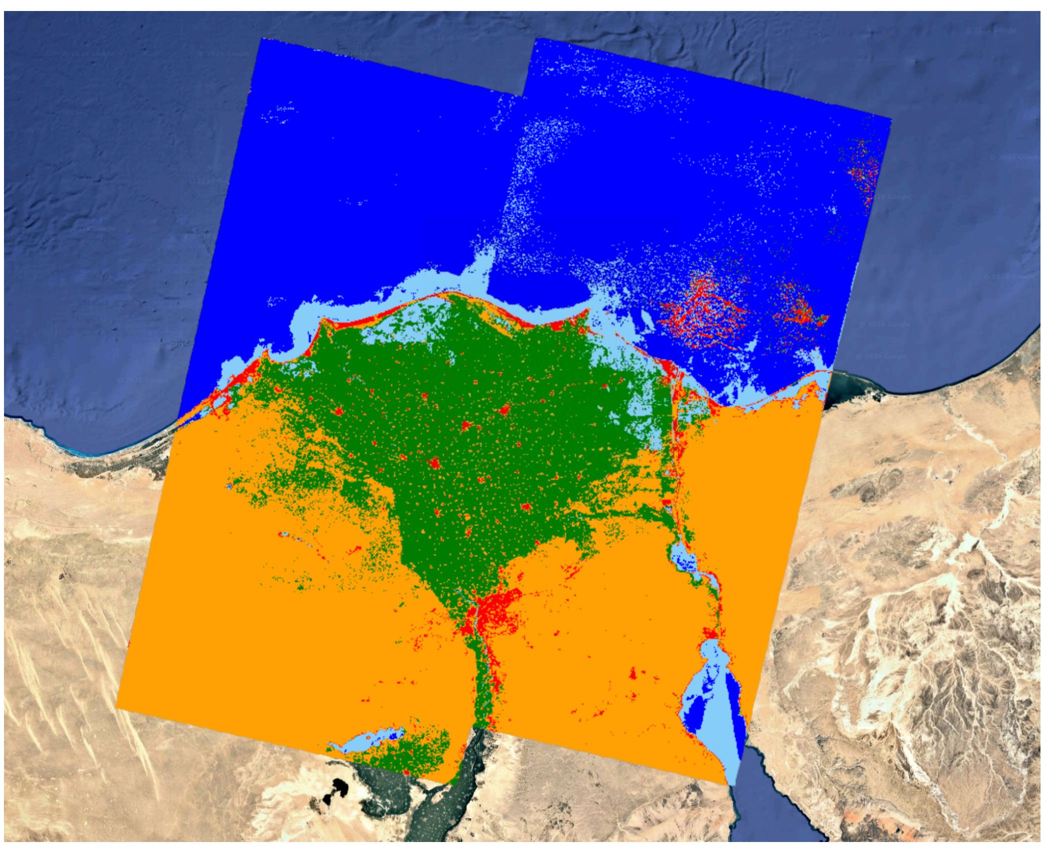

Fig. 7. Output of the supervised classifier.

QUESTION 15. How well is your coastline represented at the boundary of the 'open water' class?

The coastline itself is clearly and accurately defined between aquatic and terrestrial land cover types, but the classifier has misclassified open water pixels closest to the shoreline as belonging to the river class.

QUESTION 16. How are green spaces within human settlements commonly classified?

Depending on their extent, green spaces that are larger than the spatial resolution of the Landsat 8 input dataset are generally classified as belonging to the vegetation class. *(Fig. 8)*

Fig 8. Left: Google Earth Engine satellite basemap, Right: Classified output from the same extent.

QUESTION 17. How are clouds and their shadows classified with the above classification approach?

Many of the thickest clouds have been classified as urban developed, lighter clouds and cloud shadows have been classified as belonging to the river class. *(Fig. 9)*

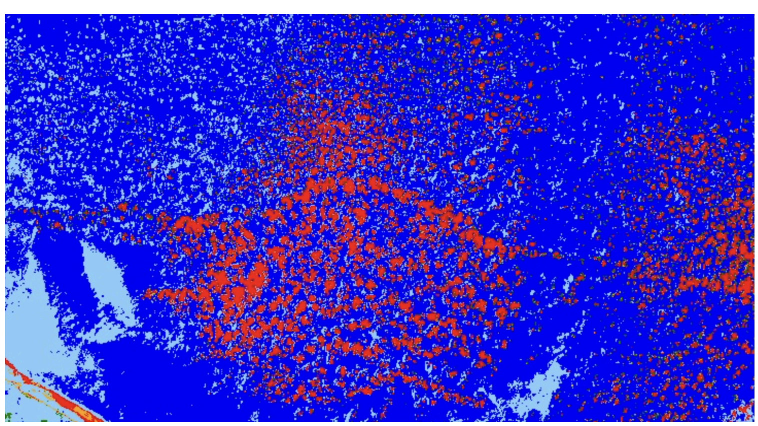

Fig. 9. Clouds and their shadows over open water in the classified output.

QUESTION 18. Are there other conditions (e.g., atmospheric haze) present in the mosaicked image that inhibit accurate classification?

The effects of haze on classification can be seen most prominently at the seam between pixels that have been derived from different images from the mosaic process over the open ocean. *(Fig. 10)* Hazy pixels that should have been classified as the type open water, have instead been classified as belonging to the river class.

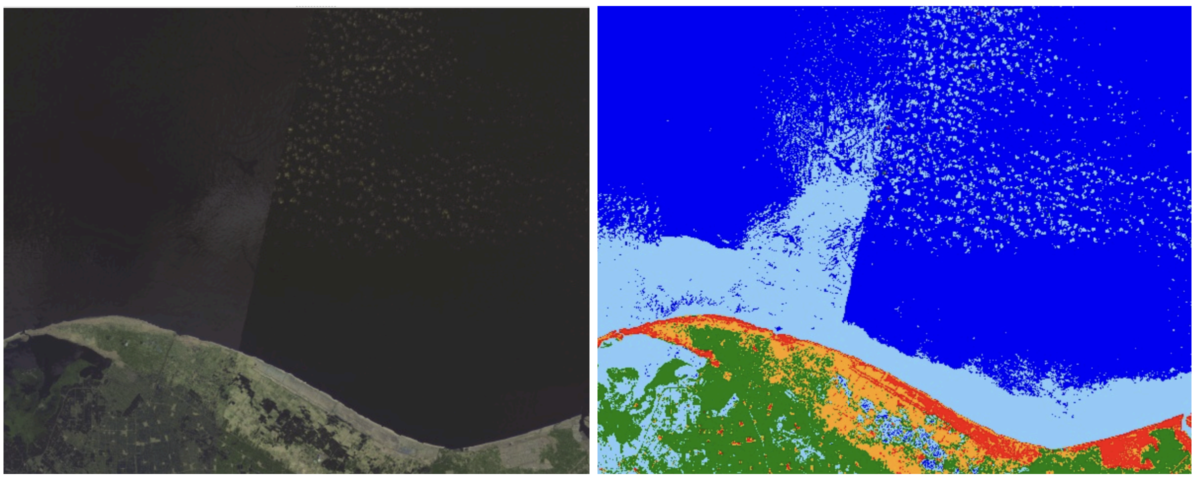

Fig. 10. The effects of haze on the classification process. Right: Landsat 8 true-color imagery, Left: Classification output.

QUESTION 19. Considering the entirety of the image, what additional classes would have improved overall classification? What criteria are you considering in your answer?

I think that the vegetation class is quite broad, and includes several different types of vegetation that may be helpful to distinguish depending on the analysis being done, by inclusion of classes like developed open or agricultural space. The vegetation class also has room for further division because there is high spectral separability between that class and the others based on the mean NDVI and NDWI values. I also think that a wetlands classification would have helped. We can see features that do not belong to the river being classified as the river type, especially in the agricultural plots surrounding the river and in the estuarine sections of the upper delta. *(Fig. 11)*

QUESTION 20. Which of the five existing land cover classes, if any, should be merged into a single class? What criteria are you considering in your answer?

I most likely would merge the river and open water classes into a single class. They’re very similar spectrally, and are the most frequent subjects of misclassification in our classified output.

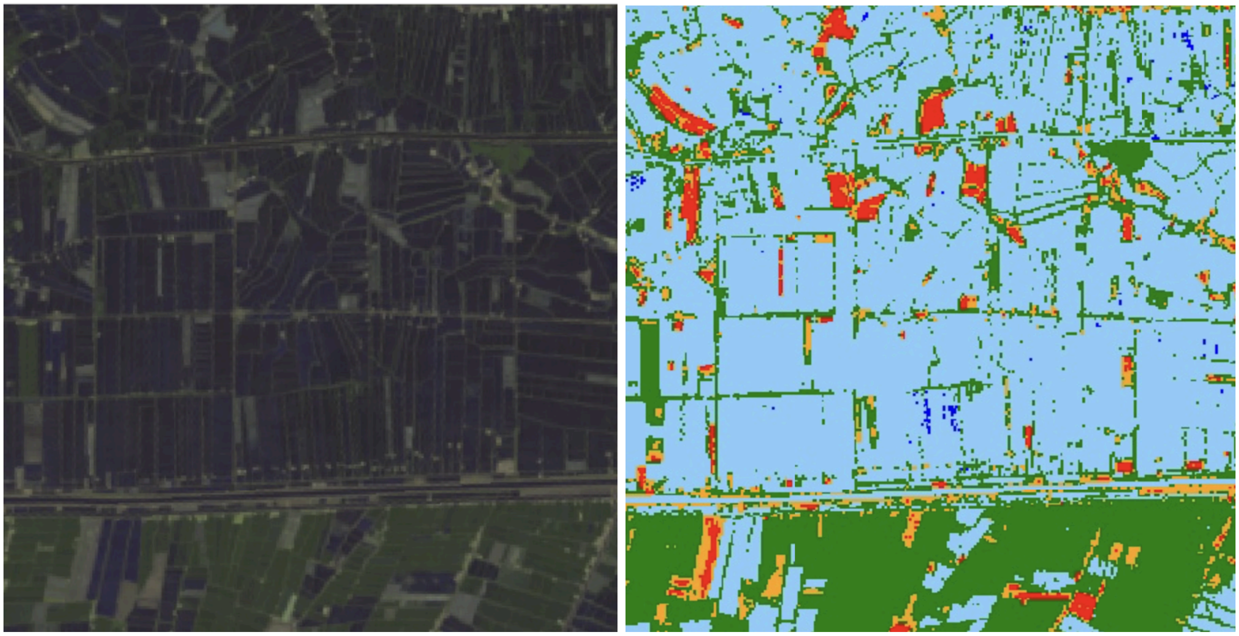

Fig. 11. Commission of pixels representing agricultural plots into the river class. Right: Landsat 8 true color imagery. Left: Classifier output.

QUESTION 21. Download this CSV from your Google Drive, and submit this CSV file alongside your answers to the above questions to the assignment.

I was able to download the CSV file, and will submit them alongside this document.

QUESTION 22. Visually compare your classified maps before and after filtering by a minimum mapping unit. Briefly describe the differences between the classified maps. Did the threshold approach work as you expected?

The threshold approach didn't work exactly as I had expected; it makes sense after checking the help documents for the functions in the code block more, but for some reason I expected pixels that failed the threshold to be assigned to a neighboring class, but instead they're just masked out and made transparent. I did expect for small groupings of pixels to not meet the threshold criterion and be removed in one way or another, and we can see this in expected places like for thinner sections of river and in the cloudy area over the open ocean. *(Fig. 12)* Some pixels that you would expect to see grouped, such as long lengths of river, have unexpected breaks that cause them not to pass the threshold.

QUESTION 23. Include a zoomed-in screenshot of this `nile_classified_mmu` map in your assignment hand-in that shows the absence of pixel clusters removed because they fell below the MMU size threshold.

The lightest pixels in the image below show pixel clusters that have been removed. We can see that some stretches of river have been removed, while others have remained, as well as other small contiguous pixel patches throughout. *(Fig. 12)*

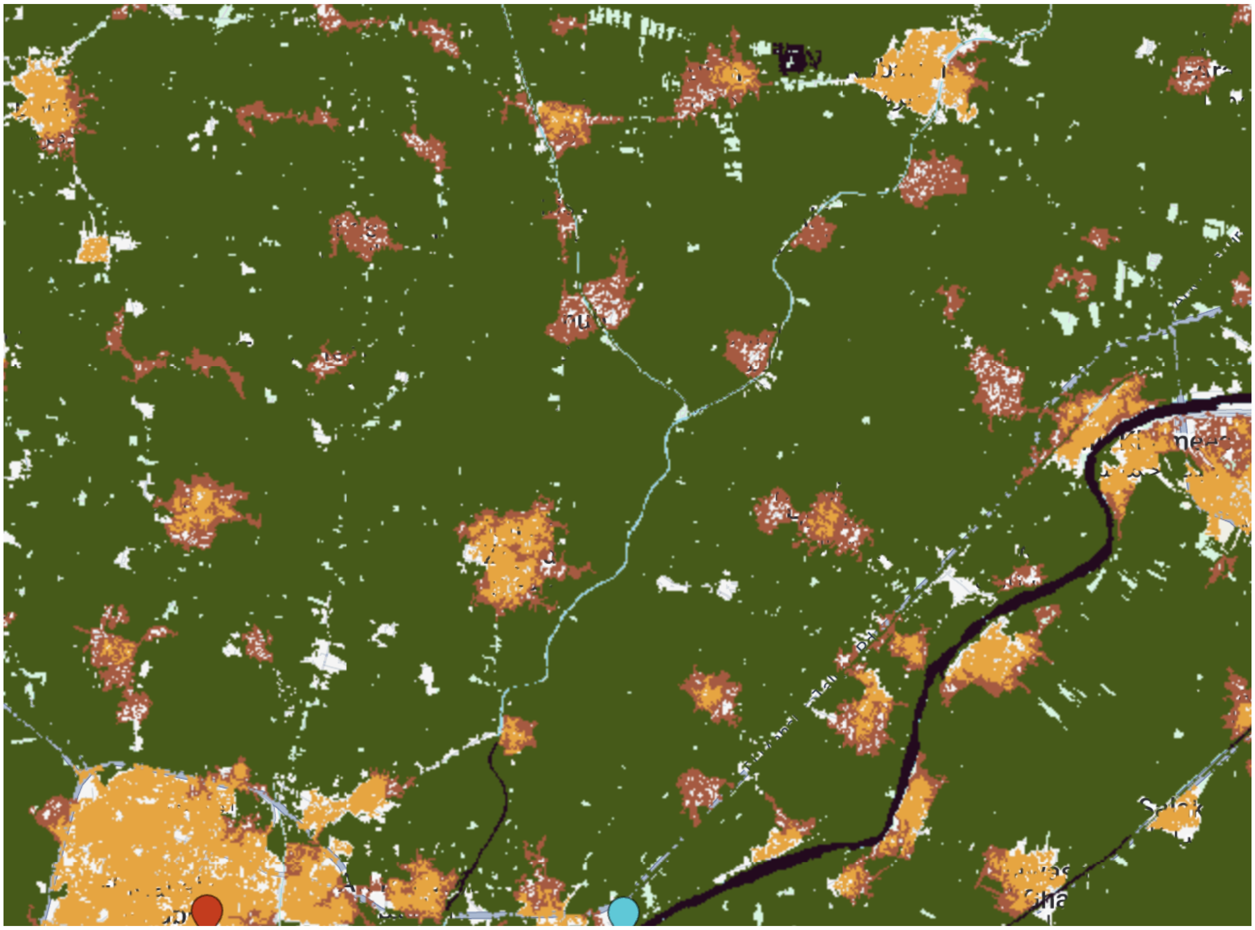

Fig 12. Classification output after Minimum Mapping Unit (MMU) filtering. The classes are represented by a random color palette, and the "Map" basemap used to emphasize masked pixels.