How Many Trees?

Introduction

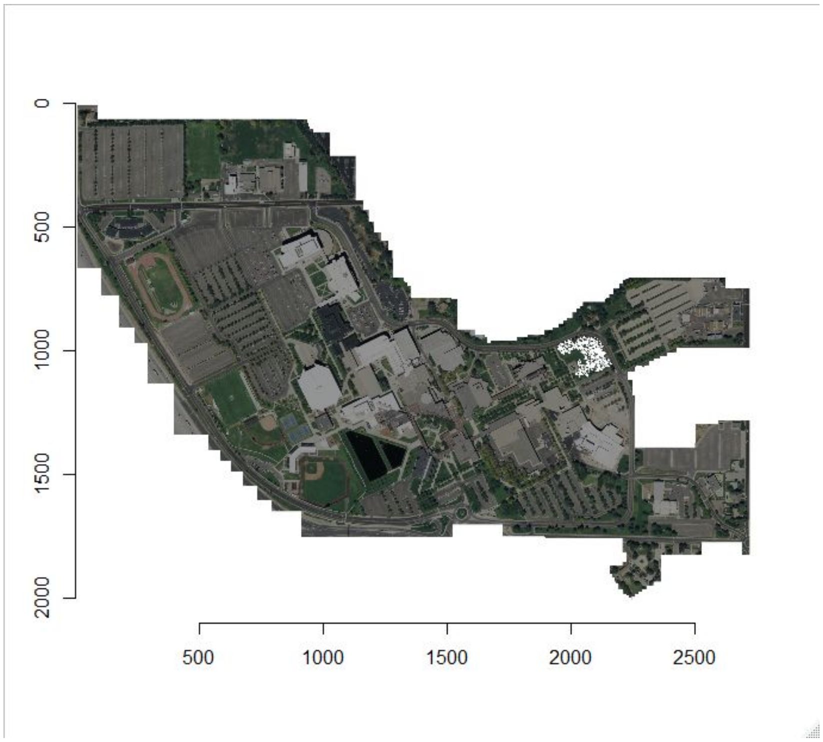
When given the offer, of traditional statistics versus the big data approach I took on the challenge. How can I estimate the total number of trees on UVU campus via the big data approach of satellite imagery and R.

Methods

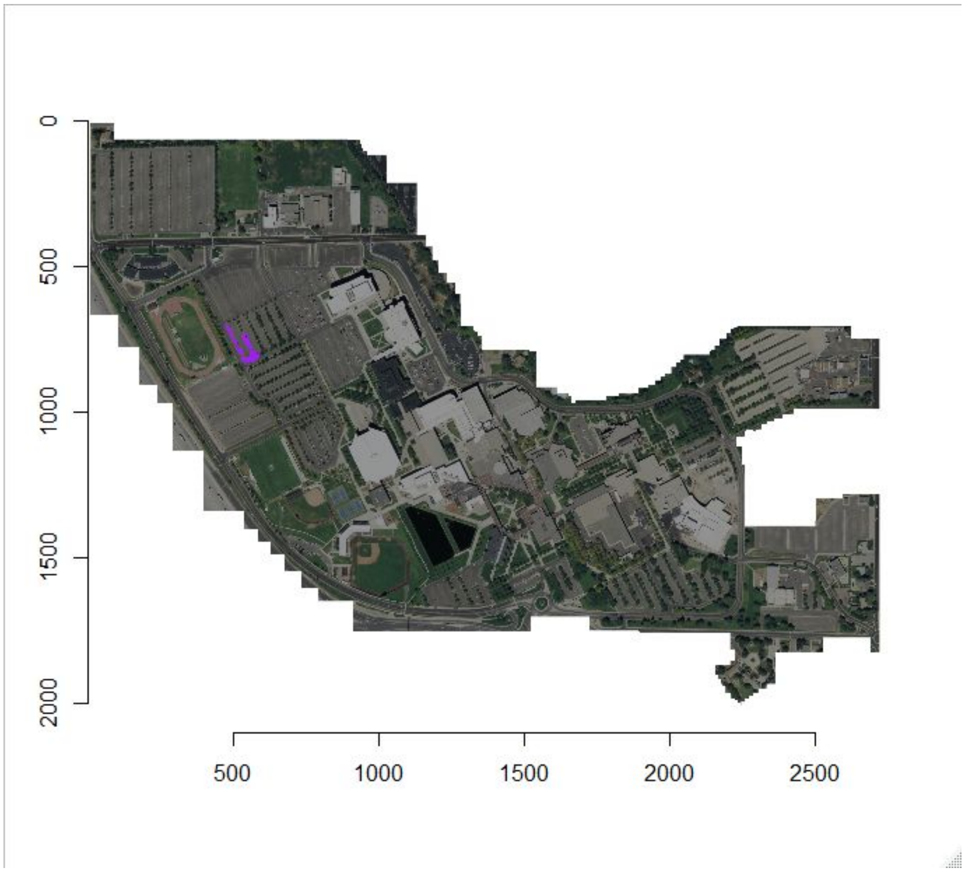
I went for the “big data” image processing methodology where I took an aerial photo from August of 2018 and used the imager package from R to do pixel based analysis of the photo. I cropped the aerial photo roughly in like with the map provided and discussed in class from this link: Found here. You can see the cropped image that I performed the analysis on here: Here. The cropped image has a width of 2721 pixel, and a height of 2077 pixels. Where as the initial image was 9460 pixels by 12140 pixels.

My initial plan for counting the total number of trees involved determining the total number of tree related pixels and then creating a normal distribution of tree sizes based on shape analysis, however this turned out to be much more difficult than expected. The color of both trees and campus ran across a near identical spectrum of greens, where we had grass in the soccer field the same color as trees next to the Sparks Automotive building. I spent hours trying to use px.flood and bucketfill from the imager package to outline small chunks of trees alone off a simgas(pixel neighbor color tolerance).

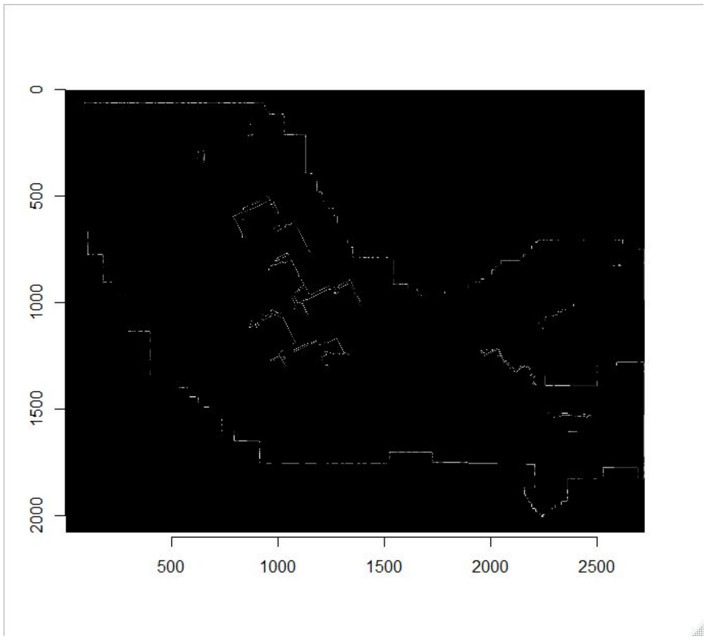
In some areas I had great success differing between trees and grass(see image)



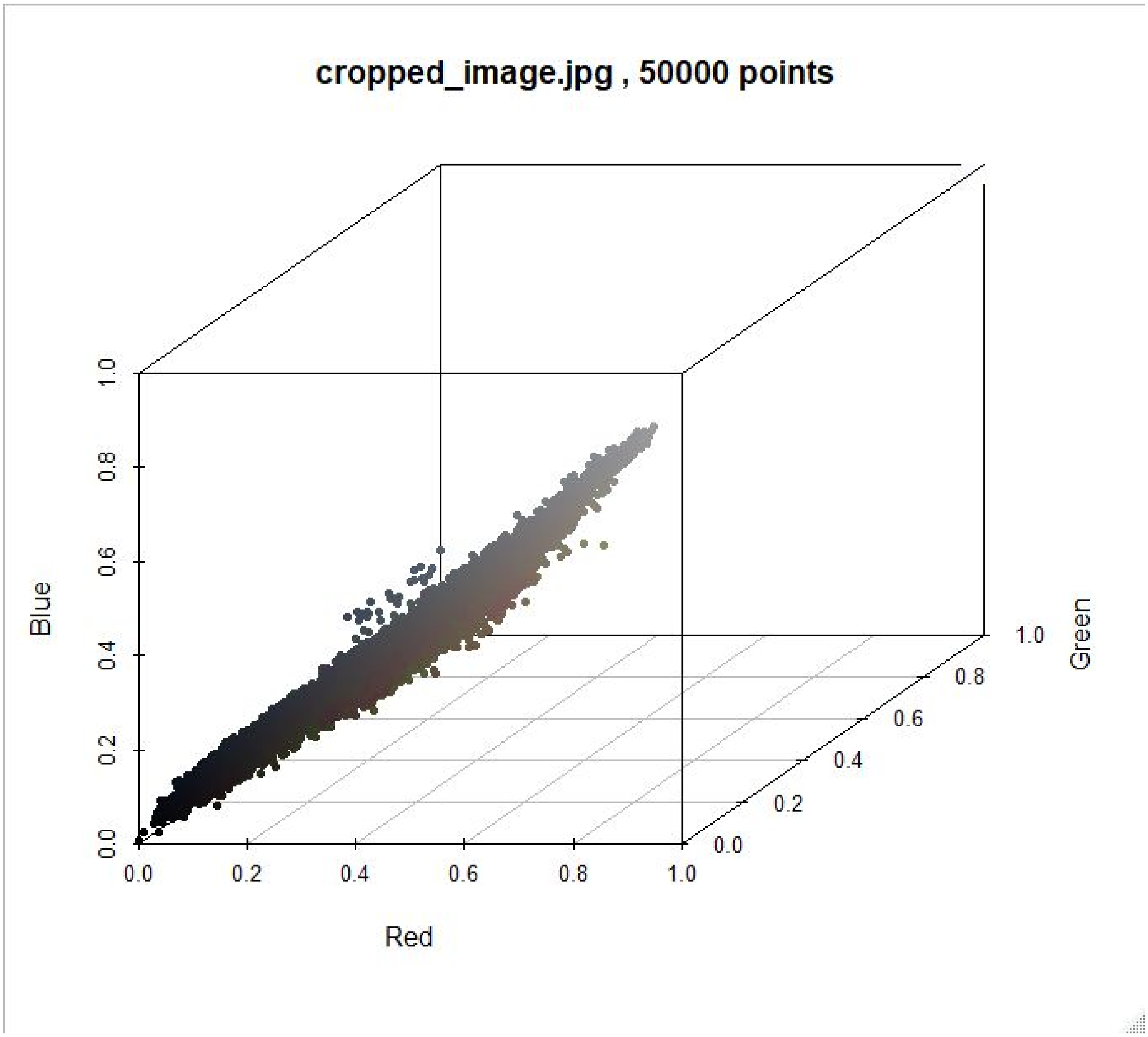
However in other areas I ended up with miniscule chunks outlined(see image), despite the fact that it seemed like it should detect more trees around.



All of this was further thwarted when I attempted to switch to edge detection and ended up the image below using the cannyEdges function.

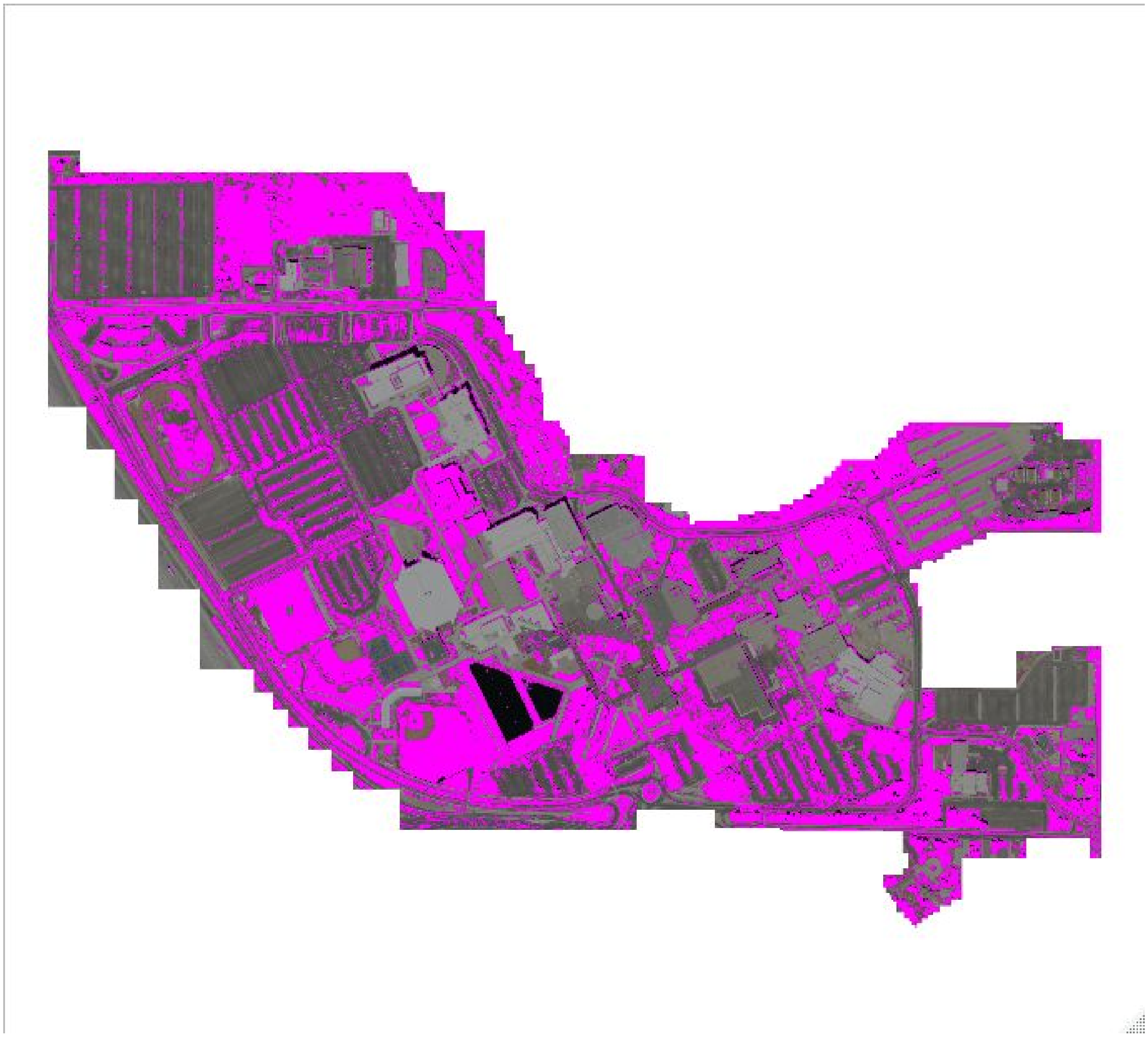


I then spent several hours experimenting with the countcolors package. I was able to do things such as plotting the pixel color distribution

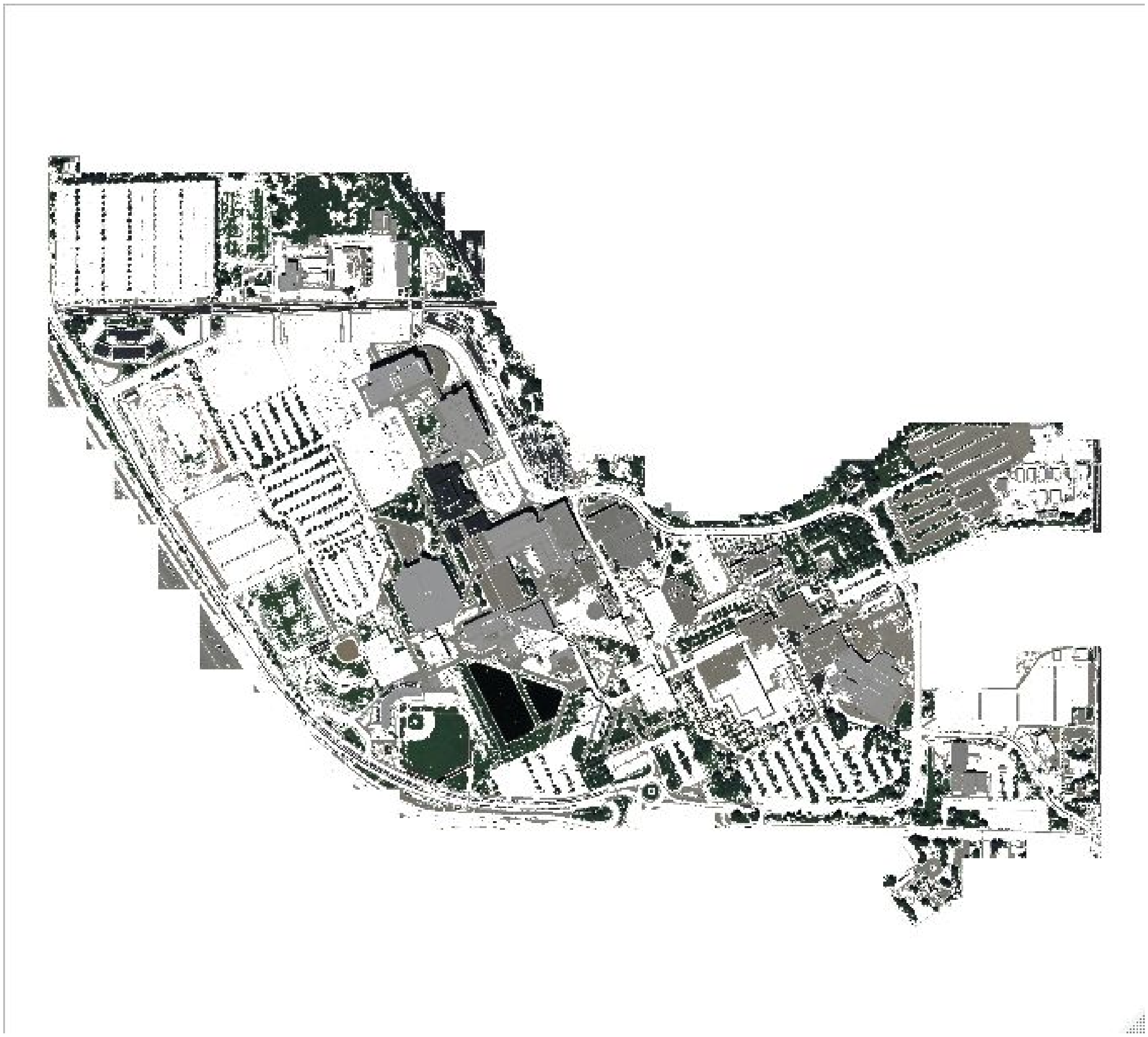


Running some k-means clusters allowed me to determine the proportion of several different color bases. 20.22722% of the image is some form of green, 40.17038% is white(the cropped out parts on the outside), 26.99302% is a black/dark grey, mainly roads and parking lots, and 12.60938% is buildings, cement walkways, and cement parking lots. I've highlighted these observations in some images below.

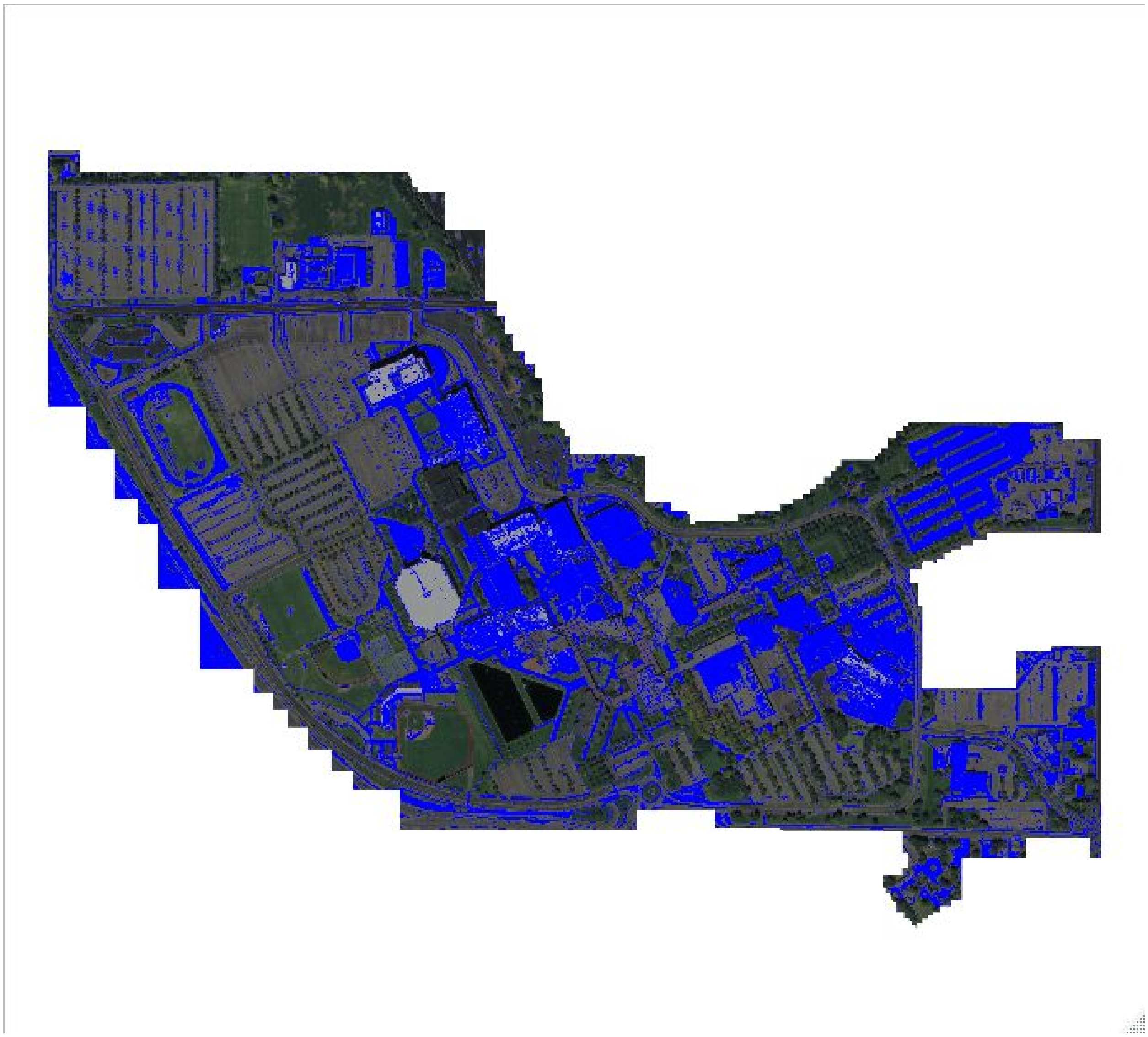
All the green pixels turned magenta



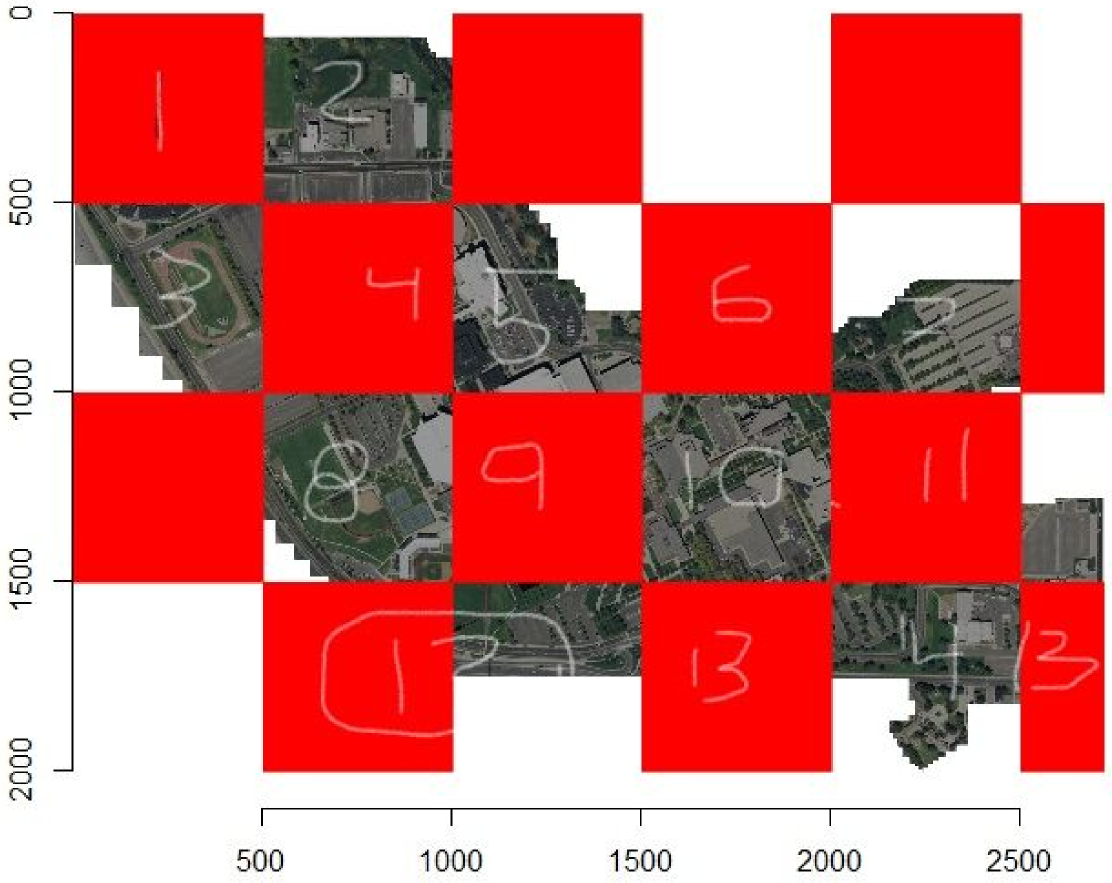
All the “roads” turned white:



And all the “buildings” turned blue:



Since the buildings take up less than 12% of campus, with the presumed restriction that trees indoors are a potential fire hazard I am assuming that there is an insignificant amount of trees indoors to affect my final count. To attempt to estimate the amount of outdoor trees I drew a checkerboard of boxes across my image to cut it into smaller easier to manage chunks. And numbered each box/ approximate full box.



I then used R to randomly select two squares(8 and 10 in this case). I then counted the number of trees in each area from the satellite image. 8 had approximately 166 trees and 10 had approximated 256 trees. I then averaged the two together getting ~211 trees per square. To estimate the total number of trees I did some basic math, original image is 2721x2077 pixels, with ~.41 being white, so I have 3,381,281 pixels that aren't white. Each square is 250,000 pixels so I have a total of 13.5 squares worth of pixels in my image. That means I have approximately 2848.5 easily visible trees on campus.

Results

My estimated count for the total number of trees on campus was 2848 trees, plus ~1,500 trees or minus ~250 trees.

Discussion

The big data turned out to be much worse than I was hoping for. First, because if my inexperience with image processing, and secondly because of all the error it introduced. Looking back, if I wanted to make the sampling more precise, after drawing the squares I could have done k-means clustering per square and obtained the approximate number of “green” in each square, and used that to do some stratification.

Just like with the last test I am near clueless as to how I could estimate the error, the “big data” and satellite imagery introduce a ton of error. The image is ~16 months old so there are both new trees my image didn’t have, and old trees on my image that have since been cut down. Due to the nature of satellite imagery there are hundreds if not thousands of trees I could have missed. I’m not sure how any method short of manually counting them could get you precise numbers of miniscule trees jammed into one single area(as discussed in class). This is all definition error, where I can only count trees of a significant enough size to be distinguishable from their surroundings.

The most important lesson I learned from all of this was it’s never as easy as it seems, even just randomly sampling from squares like I did you run the risk of selecting one(or more) squares that have an extremely high or low concentration throwing off my final estimate. My “plus ~1,500 trees or minus ~250 trees.” is just an eyeball guess where I assume that there aren’t fewer trees than I guessed, but there could very possibly be a ton more trees to estimate due to the definition and time errors

References

- UVU Map: https://www.uvu.edu/maps/map.html?id=382#!ct/18632?mc/40.27936196816123,-111.71381235122682?z/16
- Aerial Photo: https://drive.google.com/file/d/1LM3KIEfUH_B0QcnafBXts2GmpuRVouGR/view?usp=sharing
- Cropped Photo: https://drive.google.com/file/d/12I-LfErvJgNqpM73-f9o02TOIk44D-ex/view?usp=sharing
- Imager Documentation: https://www.rdocumentation.org/packages/imager/versions/0.41.2
- autocrop documentation: https://www.rdocumentation.org/packages/imager/versions/0.41.2/topics/autocrop
- draw_circle documentation: https://www.rdocumentation.org/packages/imager/versions/0.41.2/topics/draw_circle
- px.flood documentation: https://www.rdocumentation.org/packages/imager/versions/0.41.2/topics/px.flood
- bucketfill documentation:
- https://www.rdocumentation.org/packages/imager/versions/0.41.2/topics/bucketfill
- Countcolors package documentation: https://www.rdocumentation.org/packages/countcolors/versions/0.9.1
- countColors function documentation: https://www.rdocumentation.org/packages/countcolors/versions/0.9.1/topics/countColor
- cannyEdges function documentation: https://www.rdocumentation.org/packages/imager/versions/0.41.2/topics/cannyEdges
- Hough_circle function documentation: https://rdrr.io/cran/imager/man/hough_circle.html