Flurosat task (Garrick Paskos)

Task information

I was given the following task from Sowmya Manchem at Infopeople:

The task is to write a Python script that generates geo-jsons of the field anomalies (as per the screenshots provided) for labelling by an agronomist/field scout. Anomalies are defined as spatially homogeneous areas of significantly low performance of the crop in the field.

The candidate is encouraged to offer a solution that would suit all of the provided test cases (aerial imagery). Candidate can use as many/few layers of imagery as they would like.

The desirable solution is the one that is robust and scalable to different test cases, does not require an input from the user to produce the 'good guess'-type result. Good solution will also have parametric inputs to allow the user fine tune the result suggested by the system in real-time via a UI embedded into a FluroSense tool (shown on screenshots).

I was also given the Anomaly_PyCV_test_task.zip file as an attachment.

Environment

I developed/tested these scripts on x64 Windows using the Anaconda Python 3 distribution.

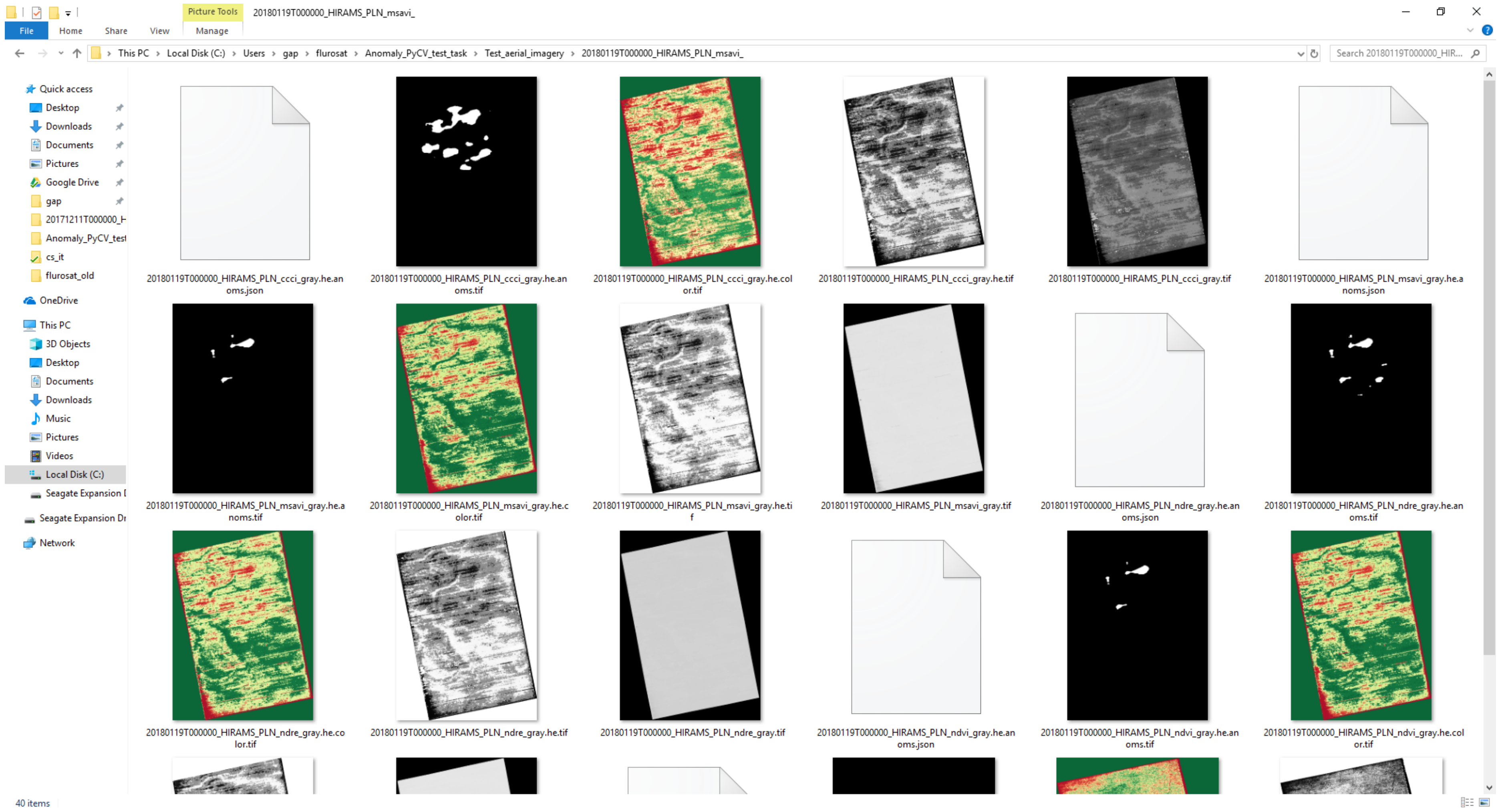
Reproducing environment

1. Download and install https://repo.continuum.io/miniconda/Miniconda3-latest-Windows-x86_64.exe
2. Start → Anaconda Prompt
3. (base) C:\Users\gap>conda update conda
4. (base) C:\Users\gap>conda create -n flurosat python=3.6.0 gdal opencv matplotlib geojson

I can set up a cloud environment if requested.

Process all files

1. Ensure you're using the flurosat environment
    a. (flurosat) C:\Users\gap\flurosat>conda activate flurosat
2. Ensure the Anomaly_PyCV_test_task.zip file is in the flurosat directory
3. (flurosat) C:\Users\gap\flurosat>python unzip.py
4. (flurosat) C:\Users\gap\flurosat>python prep.py
    a. This will get histogram equalize all images (they will have the extension '.he.tif')
    b. It also removes the black 'background' though cv2.floodFill on each corner of the image
        i. This is necessary because 'black' (or lower valued) pixels indicated anomalies
5. (flurosat) C:\Users\gap\flurosat>python anoms.py
    a. Without an argument this will create images of anomalies ('.he.anoms.tif' files)
6. (flurosat) C:\Users\gap\flurosat>python gjs.py
    a. This will create geo-jsons ('.json')
7. Optionally, create color images (with the ramp in Anomaly_PyCV_test_task/Screenshot 2018-08-10 00.55.26_labelled.png) of the '.he.tif' files



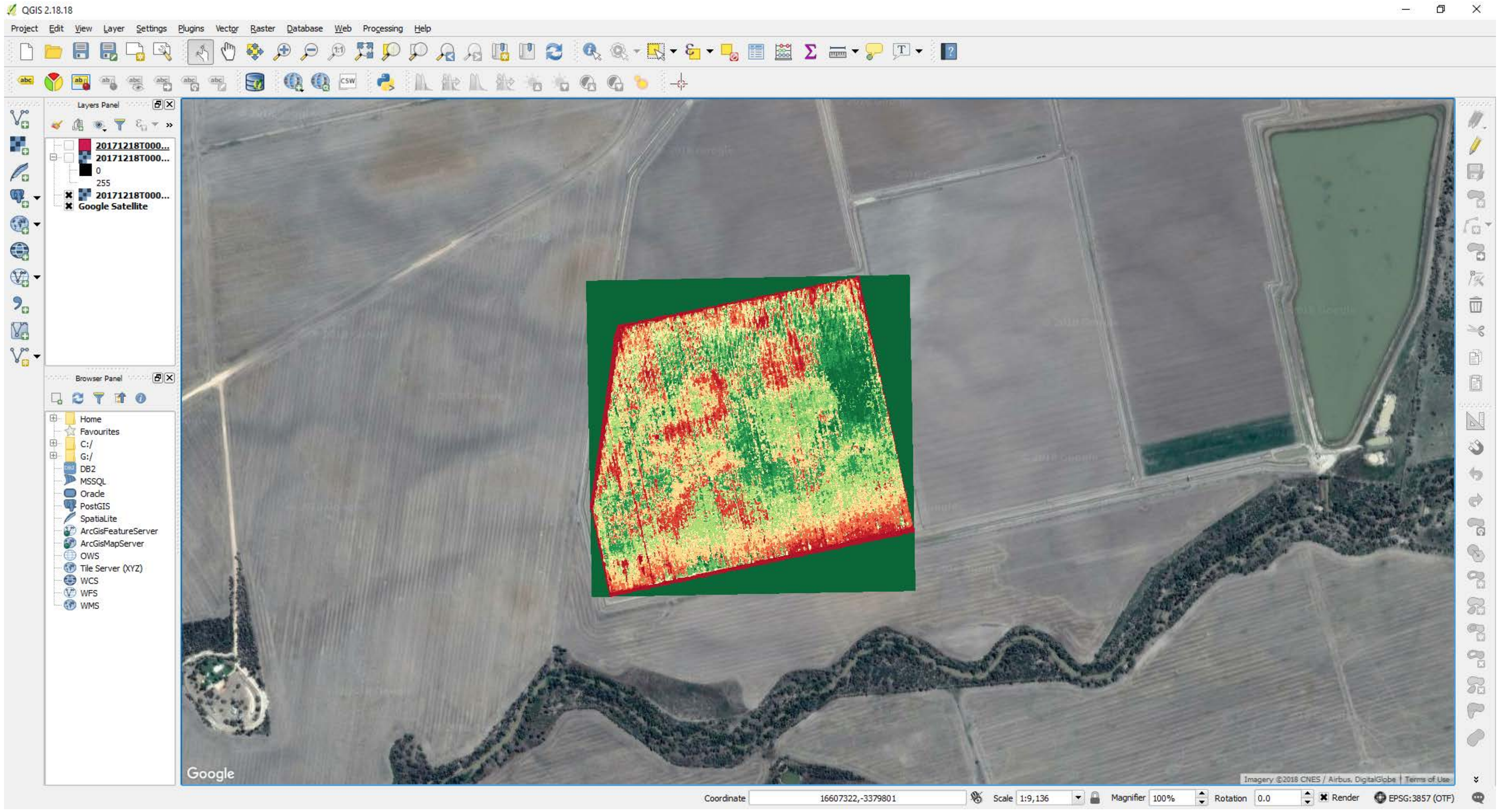
20180119T000000_HIRAMS_PLN_msavi_

This PC > Local Disk (C:) > Users > gap > flurosat > Anomaly_PyCV_test_task > Test_aerial_imagery > 20180119T000000_HIRAMS_PLN_msavi_

20180119T000000_HIRAMS_PLN_ccci_gray.he.anoms.json
20180119T000000_HIRAMS_PLN_ccci_gray.he.anoms.tif
20180119T000000_HIRAMS_PLN_ccci_gray.he.color.tif
20180119T000000_HIRAMS_PLN_ccci_gray.he.tif
20180119T000000_HIRAMS_PLN_ccci_gray.tif
20180119T000000_HIRAMS_PLN_msavi_gray.he.anoms.json
20180119T000000_HIRAMS_PLN_msavi_gray.he.anoms.tif
20180119T000000_HIRAMS_PLN_msavi_gray.he.color.tif
20180119T000000_HIRAMS_PLN_msavi_gray.he.tif
20180119T000000_HIRAMS_PLN_msavi_gray.tif
20180119T000000_HIRAMS_PLN_ndre_gray.he.anoms.json
20180119T000000_HIRAMS_PLN_ndre_gray.he.anoms.tif
20180119T000000_HIRAMS_PLN_ndre_gray.he.color.tif
20180119T000000_HIRAMS_PLN_ndre_gray.he.tif
20180119T000000_HIRAMS_PLN_ndre_gray.tif
20180119T000000_HIRAMS_PLN_ndvi_gray.he.anoms.json
20180119T000000_HIRAMS_PLN_ndvi_gray.he.anoms.tif
20180119T000000_HIRAMS_PLN_ndvi_gray.he.color.tif

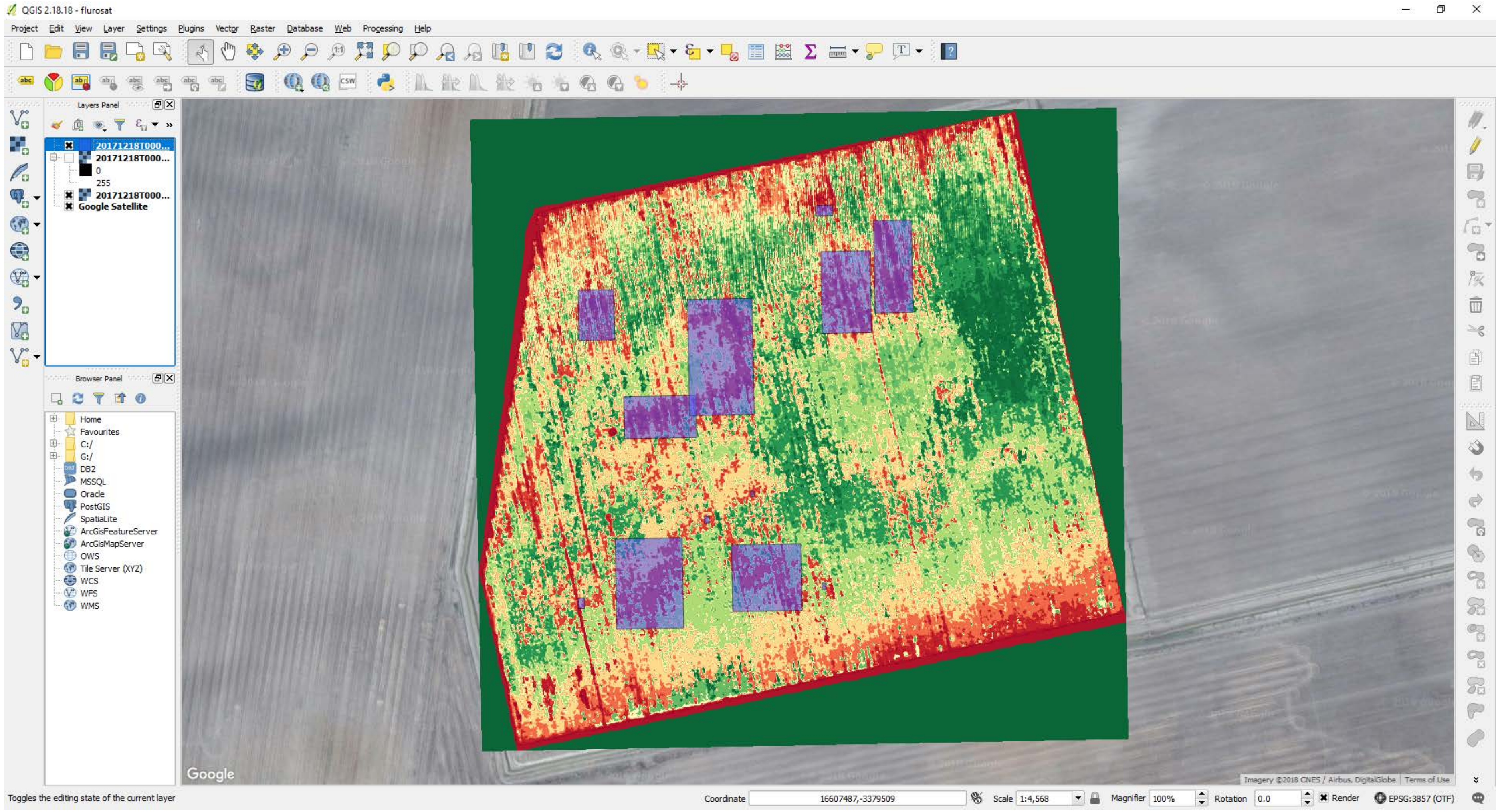
40 items

Results



20171218T000...
20171218T000...
0
255
20171218T000...
Google Satellite

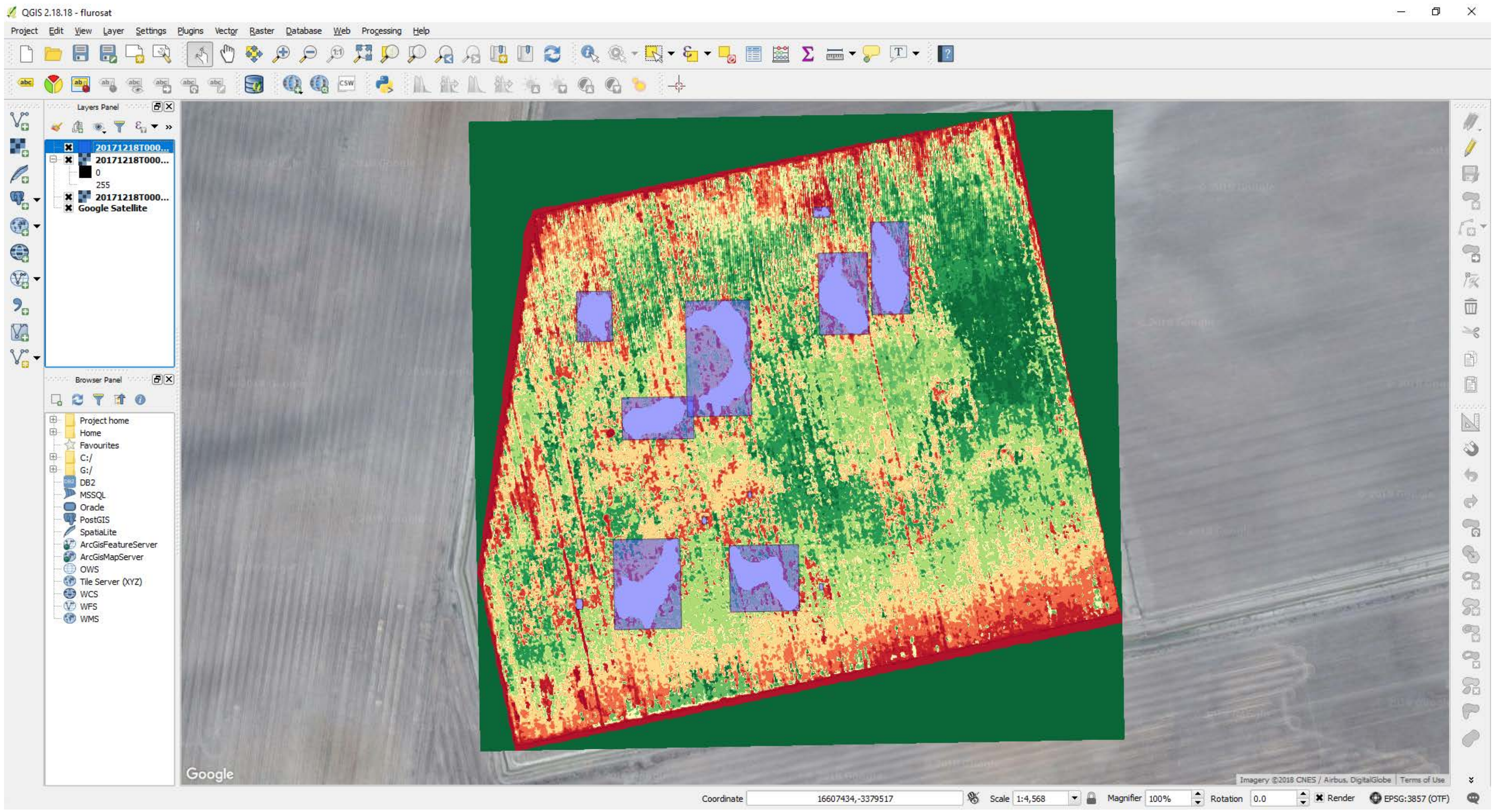
Coordinate 16607322,-3379801 Scale 1:9,136 Magnifier 100% Rotation 0.0 Render EPSG:3857 (OTF)



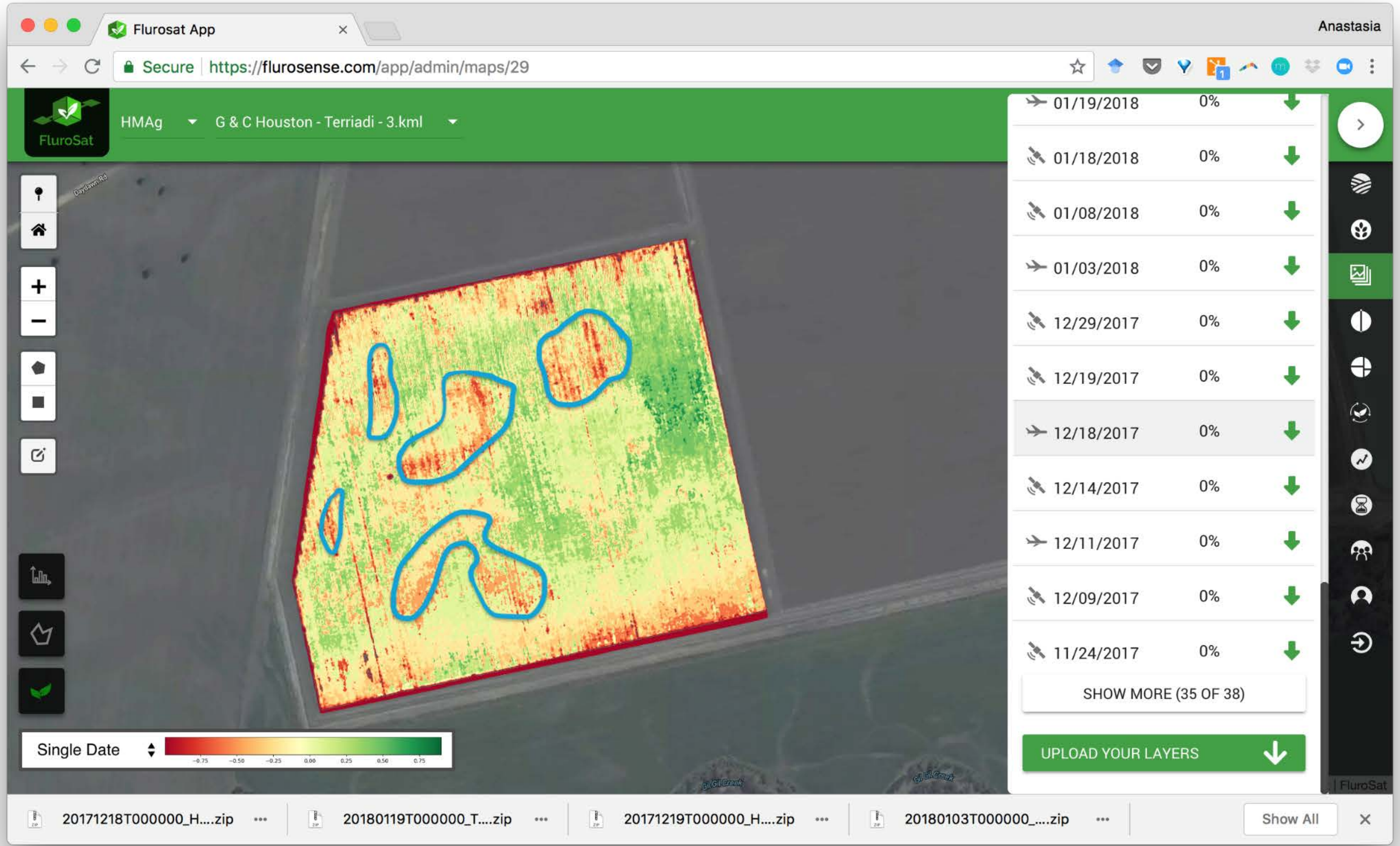
20171218T000...
20171218T000...
0
255
20171218T000...
Google Satellite

Toggles the editing state of the current layer

Coordinate 16607487,-3379509 Scale 1:4,568 Magnifier 100% Rotation 0.0 Render EPSG:3857 (OTF)



Coordinate 16607434,-3379517 Scale 1:4,568 Magnifier 100% Rotation 0.0 Render EPSG:3857 (OTF)



01/19/2018	0%
01/18/2018	0%
01/08/2018	0%
01/03/2018	0%
12/29/2017	0%
12/19/2017	0%
12/18/2017	0%
12/14/2017	0%
12/11/2017	0%
12/09/2017	0%
11/24/2017	0%

Anomaly detection

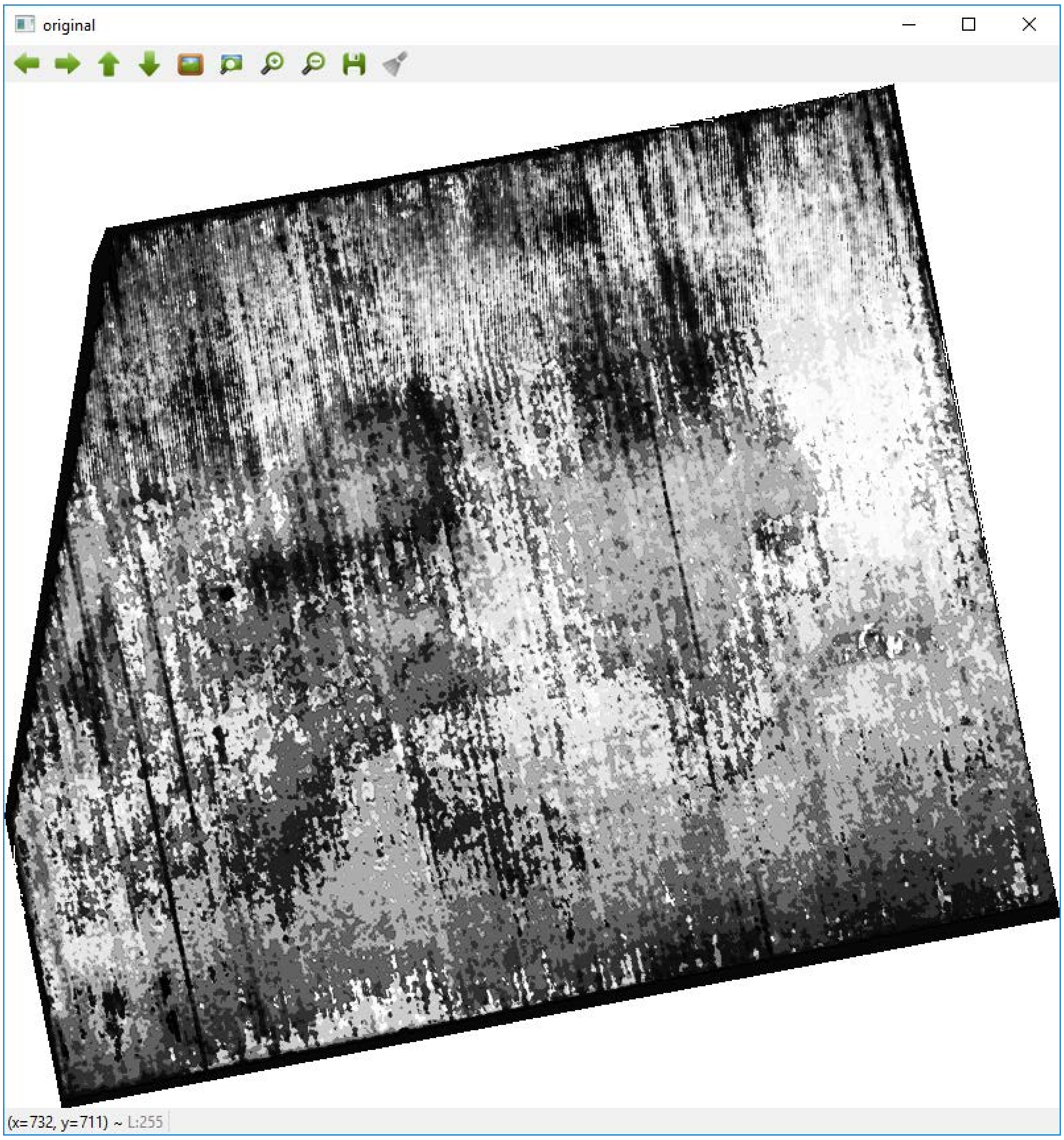
- Without any experience with vegetation indices I could not do any advanced detection that made use of multiple indicies
- ‘articles\’ contains some literature I skimmed
    - Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications (Jinru Xue and Baofeng Su)
        - ‘As mentioned before, the Normalized Difference Vegetation Index (NDVI) is the most widely used as VI’
    - Improving estimation of summer maize nitrogen status with red edge-based spectral vegetation indices
        - ‘The results indicated that red edge-based canopy chlorophyll content index (CCCI) performed the best across different bandwidths for estimating summer maize plant N concentration and uptake at the V6 and V7 and V10–V12 stages.’
- CCCI seemed to have the most contrast, but other indices had similar levels of contrast after histogram equalization

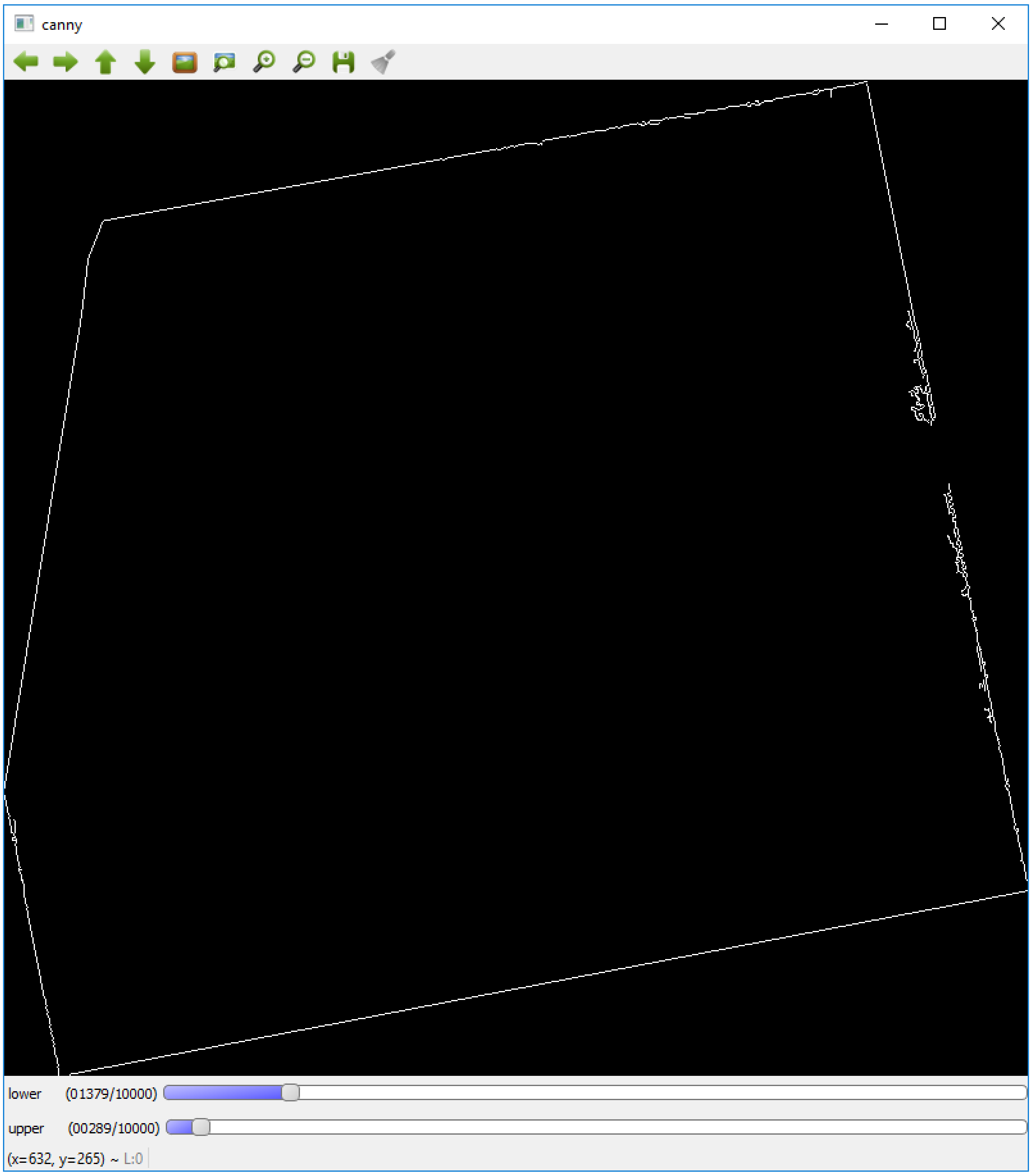
Interactive OpenCV parameter adjustment

There is further explanation of the OpenCV pipeline in the next section.

Parameters can be adjusted for an individual file with:

```
(flurosat) C:\Users\gap\flurosat>python anoms.py -f
"Anomaly_PyCV_test_task\Test_aerial_imagery\20171218T000000_HIRAMS_PLN_msavi_\2017121
8T000000_HIRAMS_PLN_ccci_gray.he.tif"
```

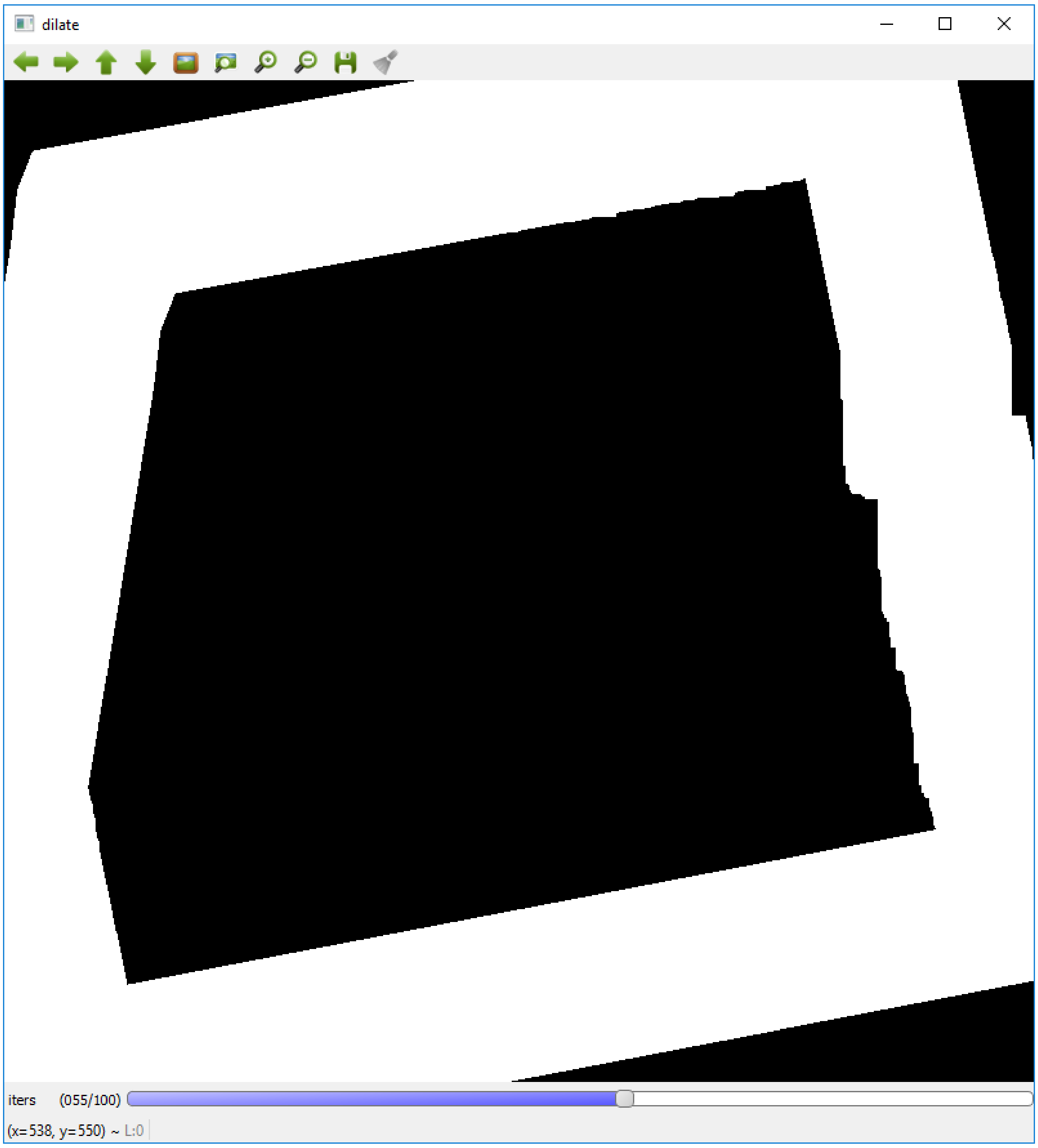

original
(x=732, y=711) ~ L:255

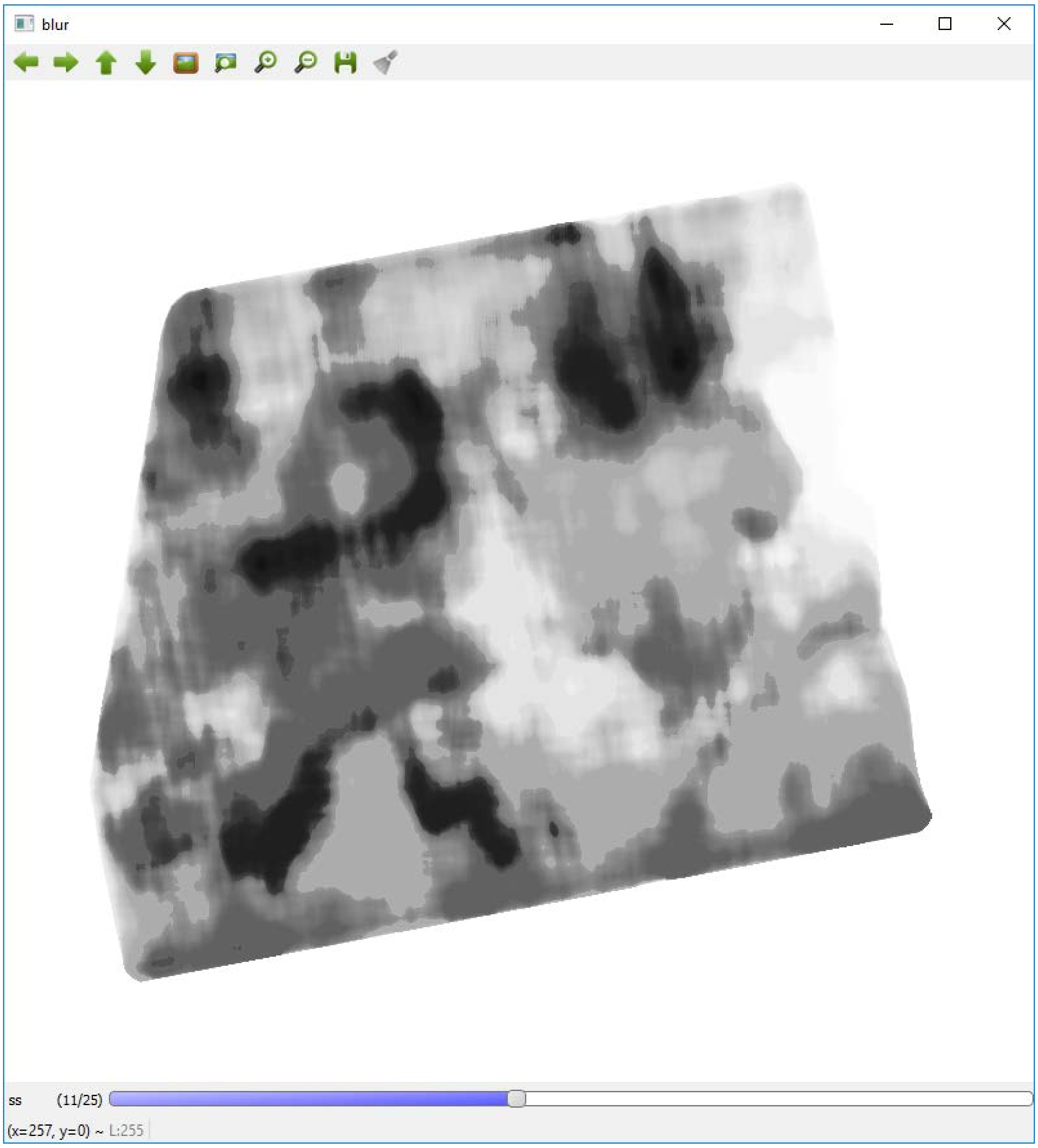


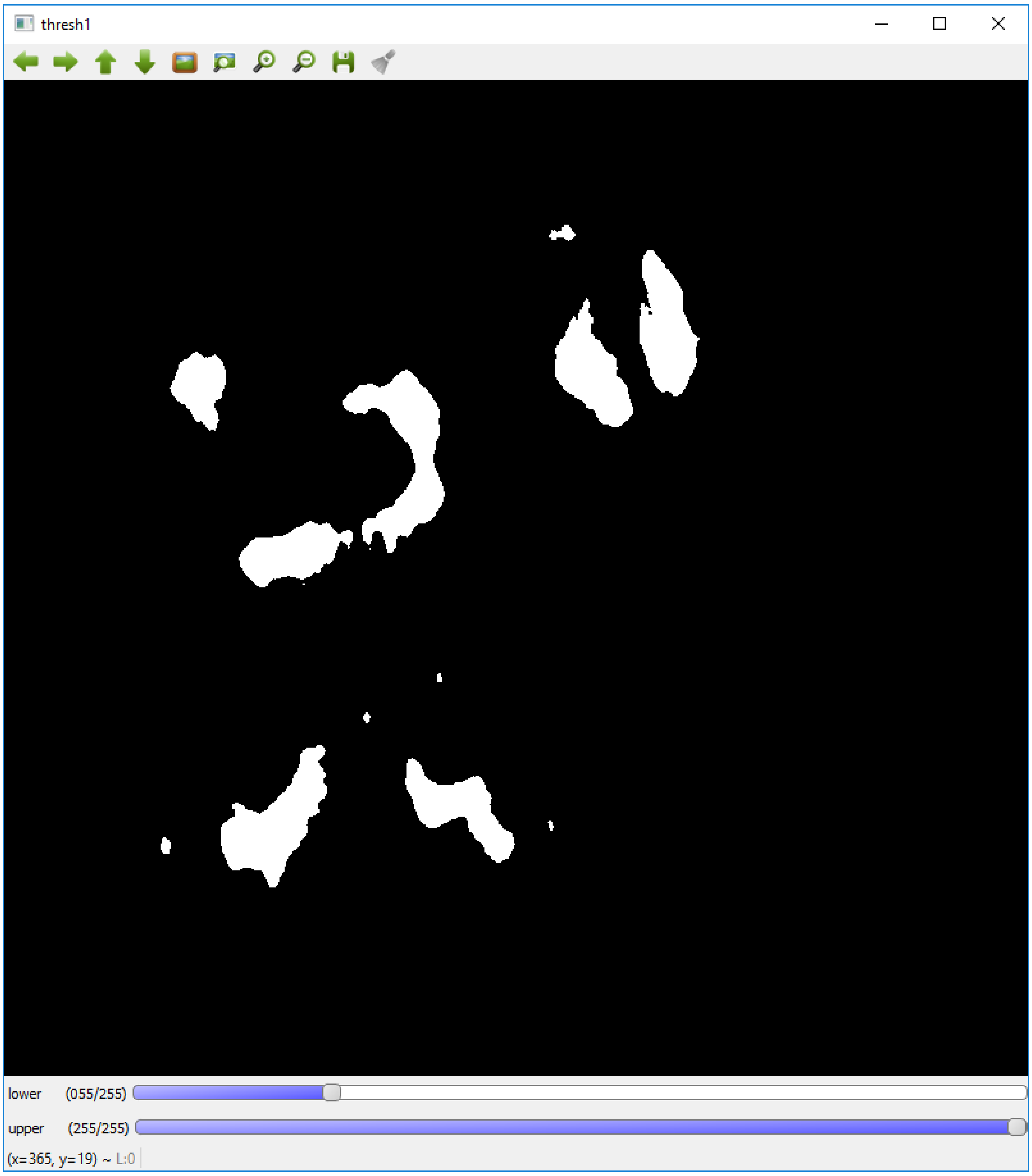
lower (01379/10000)

upper (00289/10000)

(x=632, y=265) ~ L:0


dilate
iters (055/100)
(x=538, y=550) ~ L:0


blur
ss (11/25)
(x=257, y=0) ~ L:255



lower (055/255)
upper (255/255)
(x=365, y=19) ~ L:0

OpenCV pipeline

- I based the pipeline off the image featured in the results section since it had an example of anomalies
- Issues with background/mask
    - The geotiffs didn't have a mask channel I could use
    - It wasn't so much of an issue in this case, but it can be an issue (see mask.py for the issue and how to deal with it in the case of blurring)
- Canny and dilate large to get rid of the periphery
    - Could most likely used gradient instead of canny because of the big periphery gradient (cheaper)
    - Tried Hough Lines but since we dilate so much, not needed

- Cheap medianBlur
    - Bilateral blur would have been best but would need to convert to floats and back
- Threshold with inverted option since we are interested in 'black' values

Improvements

- As mentioned, based this off "Anomaly_PyCV_test_task\Test_aerial_imagery\20171218T000000_HIRAMS_PLN_msavi_\20171218T000000_HIRAMS_PLN_ccci_gray.he.tif"
    - Pipeline parameters seem okay for other images, but not optimal
    - If had more time, would have experimented with finding optimum parameters for more images (i.e. I would have made it more adaptive)
- Geo-jsons are simple rectangles just to demonstrate ability to export to geo-json
    - Others aren't too hard to implement (https://docs.opencv.org/3.4.2/dd/d49/tutorial_py_contour_features.html)
- Can work on a machine learning solution if requested and given training data

Feedback

Please don't hesitate to send feedback to my recruiter, who will pass it on to me.

I can implement more features or undertake more tasks if it will help with my application.