Pedro Afonso
PARIS LODRON UNIVERSITY OF SALZBURG

October 25, 2025

«Divide and conquer.»
(This famous quote has never made more sense than in the context of OBIA.)

Abstract

Image segmentation has been called one of the pillars of Object-Based Image Analysis (OBIA). That being said, there are countless different ways to proceed with this segmentation, therefore it's always an interesting exercise to compare the results yielded by different approaches when segmenting the same image.

In this case, I compared the results of image segmentation done via the proprietary software "eCognition" and through Python programming with free libraries.

Keywords: Object-Based Image Analysis (OBIA), Image Segmentation.

1 Image Details

In order to complete this exercise, I was given several different images which could be used. After looking at each of them, I ended up picking Figure 1, showcasing a part of Eastern Salzburg, since it doesn't have anything too easy to segment like lakes and oceans, but it still has some noticeably different areas, like the railroads and the forest - in a way, the typical middle term of difficulty.

Figure 1: "The test subject" in true colors.

The image had a resolution of 1806 x 1396 pixels, and contained four layers: the three common bands of the visible spectrum (red, green and blue) and also a near-infrared (NIR) band.

After getting acquainted with my image of choice, I proceeded to the segmentation!

2 eCognition

For the eCognition segmentation, I used the "multi-resolution segmentation" tool we had already used in class. After playing with the color bands, I noticed two important things:

- the NIR band made the green hues stand out a lot more, as seen in Figure 2.
- if the NIR band was used as the green one, the real green band seemed fairly redundant.

With this in mind, I tinkered for a bit, until deciding to set all the layer weights to 3, except for the green band, which was left at 1, i.e. essentially making the green band three times less important than the others.

Figure 2: "The test subject" with the NIR band as a second green layer.

Regarding shape, it seemed to me that color was the most important, and several tests appeared to confirm my expectations, so I mostly used low values for the shape weight, as well as low values for compactness, which, in my mind, would better fit the long roads I wanted to represent. In the end, I settled with 0.1 and 0.2 for the shape and compactness weights respectively. Finally, my scale parameter was set to 500, to encompass the long railroads and the forest.

The result can be seen in Figure 3, which shows a good separation of green areas from the rest, even though the forest and the clearing are unfortunately each split in several segments. It also shows a good segregation of the railroad, but most buildings were included in the same segment as the adjacent street. Still, overall, it appears quite acceptable.

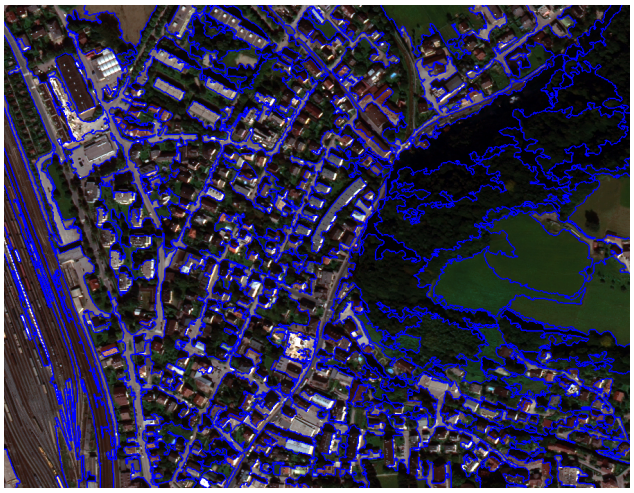

Figure 3: eCognition Results.

3 Python

For the Python-based segmentation, I took (a lot of) inspiration from the Jupyter Notebooks provided in the OBIA course. I started by normalizing the image values on all four layers, so that I could calculate the Normalized Difference Vegetation Index (NDVI), using:

$$\text{NDVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red})}$$

With it, I used Otsu's multi-threshold method included in the "scikit-image" package, but I didn't quite like the result, as the two threshold values were quite close to one another. To improve on it, I decided to take a more *ad hoc* approach and manually selected one of the thresholds as -0.06; then, I used Otsu's method for the remaining pixels, having obtained a second threshold of 0.28, both represented in the histogram of Figure 4.

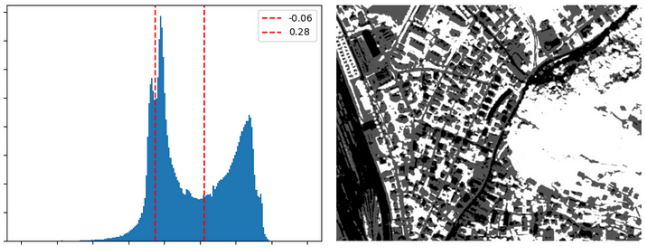

Figure 4: NDVI histogram and grey-scale representation.

I was quite happy with the final result, seen in Figure 5. It had more visual artifacts than the eCognition segmentation and aggregated some of the forest with the clearing, but it was more successful in separating buildings from the adjacent streets and also included the clearing and railroads as one large segment.

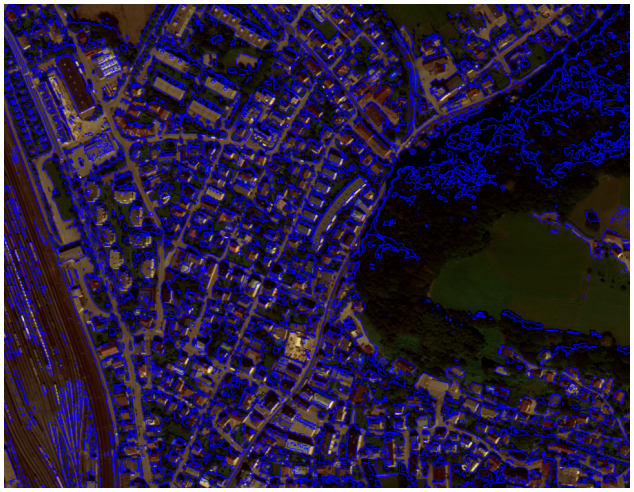

Figure 5: Python Results.