

Using object oriented image analysis to  
classify ice, water, and harbor seals in  
Johns Hopkins Inlet in Southeast Alaska.

Jay Bean  
GEOS 422  
December 2013

## Table of Contents

- I. Introduction
- II. Objectives
- III. Study Area
- IV. Data Input
- V. Method
- VI. Process
- VII. Discussion
- VIII. Advantages
- IX. Limitations
- X. Conclusion
- XI. References

### I. Introduction

Harbor seals use glacial ice in the glacial inlets and bays throughout Southeast Alaska as a platform to rest on and hunt off of. Being able to accurately classify and quantify the amounts of ice in these habitats will help researchers better understand the relationship between harbor seal population and glacial ice presence.

For my term project I worked to develop a rule set in the software program eCognition. This object oriented identification software allowed for me to classify glacial ice, water, and harbor seals. Using an object oriented image analysis approach allows for greater accuracy when it comes to classifying than a straight pixel-based image analysis. This is due to the fact that the software takes a pixels relationship to its neighboring pixels into account when determining if the pixel is its own object or part of a larger one. The pixel isn't put into an arbitrarily designated bin just because of its DN value.

Some road blocks that past work on this project has had includes the program splitting large objects like icebergs into multiple objects due to debris strewn about the surface. Also objects like rocks share many of the same pixel layers values as the harbor seals so the program is inclined to assign them to similar classes. To combat this misclassification I had to find unique characteristics for each major class.

### II. Objectives

Devise a ruleset or series of rulesets that allow for accurate identification and classification of sea ice, water, and possibly harbor seals. These rules will take into consideration attributes such as object texture, RGB layer values, and geometry. After doing the initial classification I will attempt to sub-classify the classes based unique characteristics within their class. An example would be dirty ice and bright ice or large ice blocks and small ones.

### III. Study Area

My study area is located in Johns Hopkins Inlet in Glacier Bay National Park in the Southeast region of the state of Alaska. The inlet begins at the terminus of the

Johns Hopkins tidewater glacier and is approximately 1 mile wide and 9 miles long (NPS). The inlet ends in northwest portion of Glacier Bay. This area is home to many marine mammals, including otters and harbor seals.

#### IV. Data Input

For this project I will be using aerial imagery captured by the National Park Service in 2007. The sensor for this project was a simple Nikon DSLR camera that covered the red green and blue spectrums. This imagery has a 4cm spatial resolution and 8-bit radiometric resolution. Instead of working with a full survey of the inlet I tried to choose around 5 or so images that were representative of the whole inlet. That is, some contained a lot of ice that varied in cleanliness as well as size plus varying amounts of open water. Other images that I chose had minimal ice content and debris and were used to help me learn the ropes of how to classify the image objects.

#### V. Method

In this program there were two major steps to the image classification process, the initial segmentation and then the classification and sub-classifications. I didn't have to do any pretreatment to the data set so I was able to jump right into the segmentation process.

When segmenting an image into image objects the program looks at how pixels relate to one another. If a group of pixels meet the parameters I have set then it will create an image object out of them. All of these "image objects" are added to a new "level," which is basically a layer that is laid over the image. There is a lot of plugging and playing involved with this process. For this project I used a multi-resolution segmentation followed by a "spectral difference merge". The spectral difference merge joins image objects together if their DN values fall within the range that I specify. In this case I used a spectral difference of 5. This combination of segmentation algorithms does a better job of accurately identifying objects than the checkerboard and contrast-split segmentation did. It also cut down substantially on the amount of image objects that there are in the image, which helps cut down on processing time required to run the ruleset.

The classification part of the process was one of the more time consuming parts of the project. Using the image object feature viewer window I spent quite a bit of time finding characteristics that were unique to the major classes that I was trying to identify in the images. These features included the textures, mean layer values (color values), and geometry (asymmetry, length/width,...). The easiest way to do this was to grab the "low hanging fruit" in the images such as open water or bright ice and classify them out first. After this initial round of classification I began to classify out debris and darker ice based of characteristics that I found to be unique to them. Texture played a pretty big role in identifying these objects, it was much more effective than simply using the DN values like I did on the clean ice and open water.

The sub-classification process involves creating another new "level" within the image. In this new level I can assign sub-classes to the objects based off of

criteria such as hue, area, or shape. For this project I did a basic sub-classification and created two new sub-classes, large ice ( $\text{area} > 5\text{m}^2$ ) and brash ice ( $\text{area} < 5\text{m}^2$ ).

### VI. Process

For my final ruleset I came up with a 25 step classification system that used multiple parameters in each step to isolate and classify the image objects. Each major process has a title that accurately describes what the child steps below it are doing. For the first ice classification I focused on hue and texture; the clean ice had bright DN values so it was easy to classify out. I then used a dissimilarity texture criteria and hue to classify out some of the dirtier ice. For the water identification I used the shape index and homogeneity texture criteria to get rid of the larger, more jagged shaped water bodies. These large image objects had fairly uniform texture so the homogeneity texture setting worked on getting them to classify out. Next I used set standard deviation from the mean and a minimum layer value parameter to classify out the dirty ice blocks that didn't get classified out earlier. After that I did another round of water classification that used area and mean layer values to get rid of the remaining open water. The last image object classification involved grabbing the rest of the unclassified image objects that represented debris or seals and classifying them in with the ice class. Lastly I created a new image object level and re-classified the ice objects based on size, orange represented objects above  $5\text{m}^2$  and dark red represented those that were smaller than  $5\text{m}^2$ .

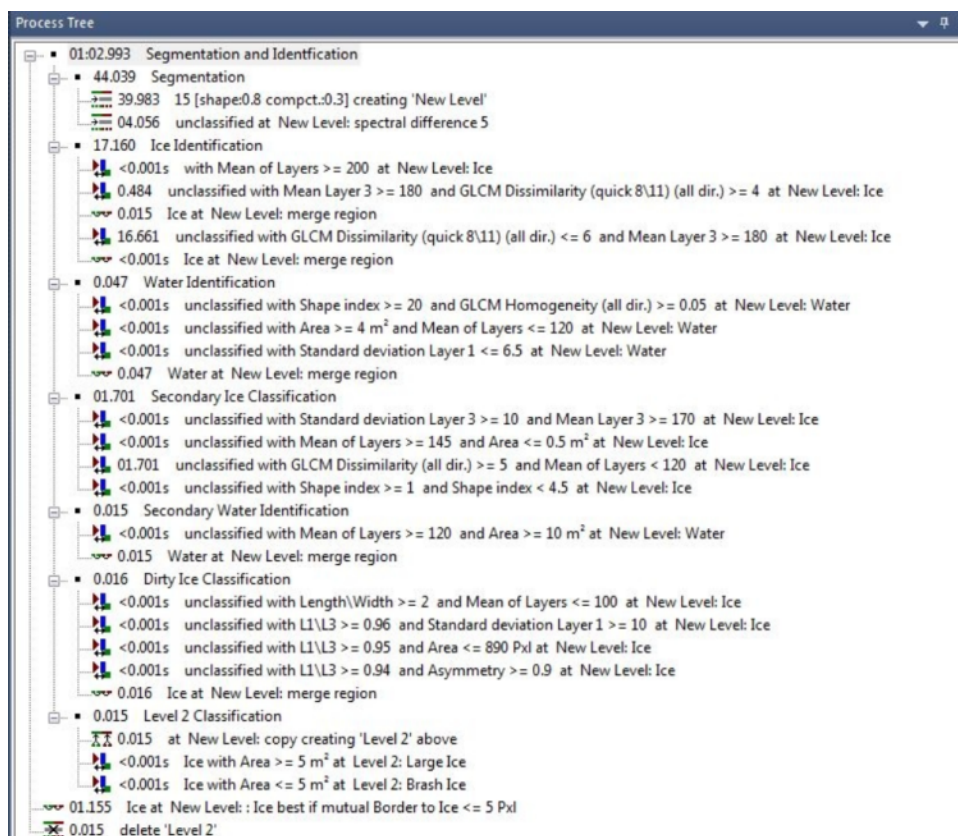
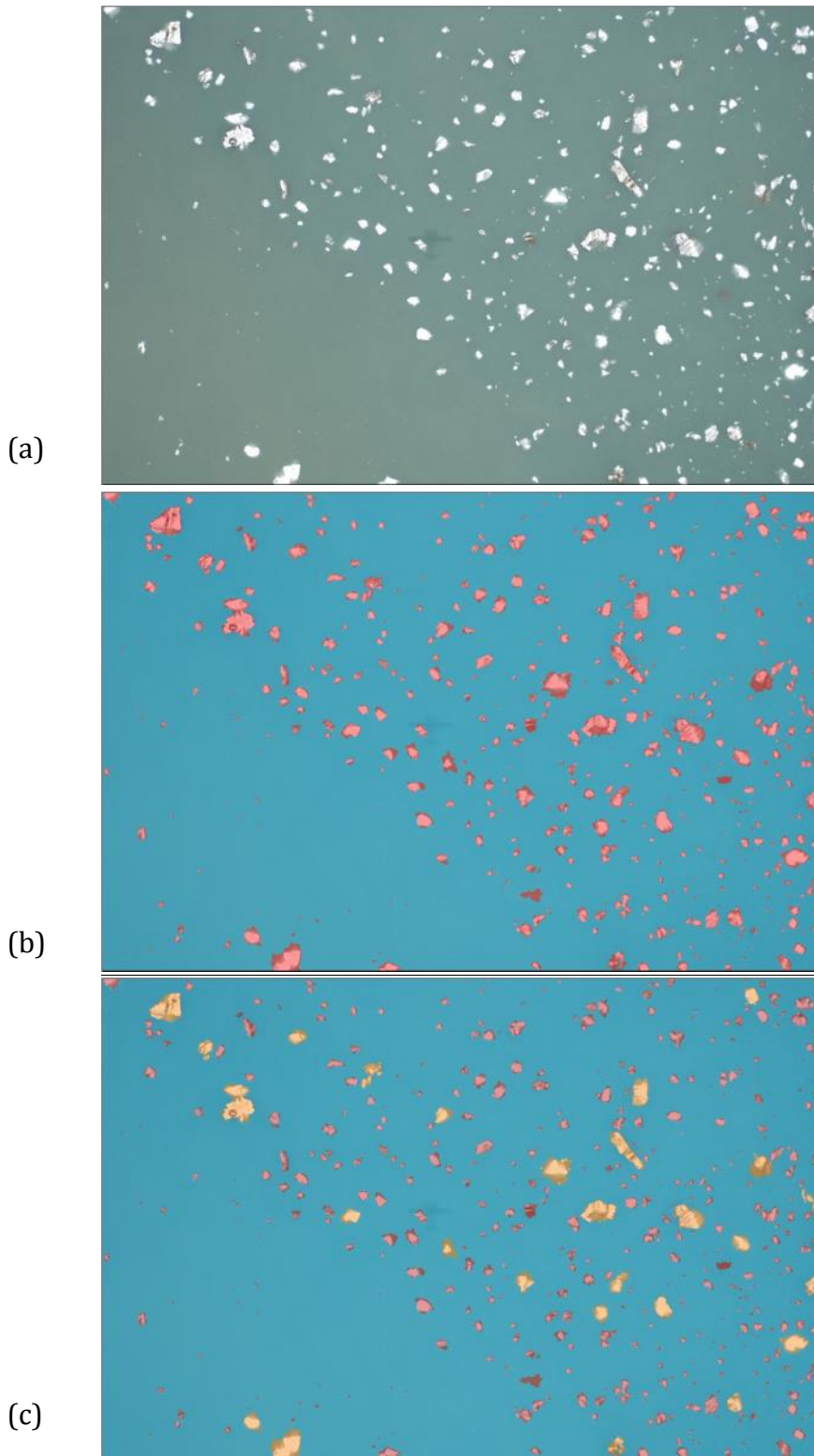


Figure 1: Ruleset that I came up with and used on my data



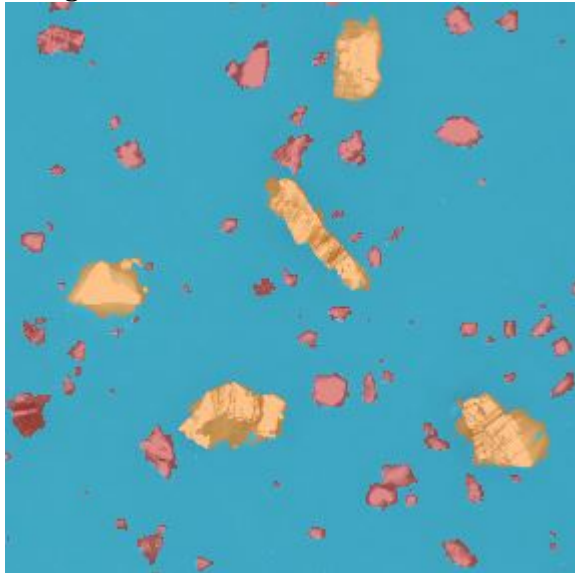
**Figure 2.** Three images that show the unclassified image (a), the first level classification (b), and the second level classification (c). The orange objects in (c) are ice blocks that are larger than  $5\text{m}^2$  and the red ones are blocks that are smaller than  $5\text{m}^2$ .

## VII. Discussion

Overall the ruleset that I used did a good job of identifying ice chunks covered with debris as single objects. It struggled a bit when it came to identifying the areas of high ice concentration due to the fact that it merged any objects that were in contact with one another into the same object. The merging was necessary to consolidate multiple image objects that made up one actual object into a single large object. The images with less brash ice tended to classify a lot more accurately than the images with a lot of it.

Also I didn't have any consistent success with identifying the seals. If I wrote the ruleset so that I could identify them in one image something completely different may get classified as seals in another. One possible solution would be to do a third level of segmentation and classification on the ice classes to try and find something unique about them.

One result that surprised me was that the aircraft's shadow didn't show up in the end product. It showed up prominently in the open water images yet the program still didn't incorrectly classify it as ice or debris, it merged it into the surrounding water.



**Figure 3.** A close-up of the correctly classified dirty ice blocks. Orange blocks are greater than 5m<sup>2</sup>, red are smaller. The image object level is partially transparent so the actual ice block is viewable below it.

## VIII. Advantages

This method provides greater accuracy than simply pixel-based classification because it doesn't just look at DN values; it looks at how well each pixel relates to their neighbors. If there is a single dark pixel in a group of lighter pixels the software may not classify it as its own object and will include it in an object with the surrounding pixels. This comes in handy when it comes to classifying the harbor seals in the image because of the variability of their colors; some strictly dark colors and others are a mix of light and dark. Traditional classification methods would classify all of those variances as separate objects.

Another benefit to this program is that if the images are georeferenced the software can export the classified image objects as shapefiles that can be used in GIS

software such as ArcMap. This is useful if someone is interested in trying to determine quantities of each class that are present in an area as well as their geographical position.

### IX. Limitations

I think that adding more spectral data to the project would have benefited the accuracy of this process quite a bit. eCognition allows for work with multiple bands at once which would have given me some attributes that I could base my classification off of. It also would have made identifying open water a lot easier because it would show up as an almost ideal blackbody.

Another limitation is that I don't yet have any programming skills so I can at least partially automate the process of loading and classifying images; I did all of it manually. Using this ruleset on thousands of images would require an exorbitant amount of time. Automating the process would allow the computer to load images, execute the ruleset, and export the data, greatly increasing the overall efficiency of the process.

### X. Conclusion

There were quite a few characteristics that all of the major classes shared so accurately classifying objects was a bit of a challenge. Some of these features include overlapping pixel values, similar geometry, and shared textures. I had to experiment quite a bit with the settings to find object features that were nearly exclusive to each class.

Overall though I was very pleased with how well the ruleset performed when it came to identifying the dirtier ice, which was one of my primary goals. It wasn't the greatest when it came to classifying high-density ice due to how it merged regions together but it was able to recognize patterns within the dense flow and outline the edges. I also wasn't able to classify the seals accurately, which was too bad as that was one of my initial goals.

XI. References

National Park Service (NPS). "An overview of selected glaciers in Glacier Bay." Feb. 2004 <<http://www.nps.gov/glba/naturescience/upload/Overview%20of%20Glacier%20Bay%20Glaciers2.pdf>>