**ASTERRA home assignment - iceberg classification**

**Elisha Goldstein**

The data is 2-bands pixel data which we treat as image data, thus the straightforward attitude will be to build a CNN. After reading the abstract referred in the instructions, it seems like a shallow network will work, which can make sense, since each image is relatively small (2\*75\*75). The number of convolution kernels and their sizes can be determined by optimization.

In addition to the 2 bands we have the ‘incidence angle’, which is maybe not information derived from the observed objects, but we should use it as it is part of the data. We can either use it as a normalization factor, as described the dark pixel values (close to 0) are the ocean areas, so the darker the ocean is (most of the image) thus we can normalize the values, something like - X\*(inc\_angle/X\_darkest\_pixel).

Another approach is to insert the incidence angle value to the CNN and let the network learn by itself. We think of three ways to do so. The first is to inject the inc\_anfle value to the FC after flattening the convolutions. However, the value here will get small weight in comparison to the pixel data (it will be only one node against the flattened convolutions). The second way is also to add the inc\_angle to the FC layer, but as an additional bias term to all the nodes. This way it will have the same amount of effect like the pixel data.

The third way is to add the inc\_angle as a new third channel to bands. The reason to do so is because the inc\_angle is tightened to the pixel's information, and it should be used as raw data in the same manner. This approach is computational expensive, however, we think this way the incidence angle gets the best representation of its value.

Normalization - according to the supplementary abstract, the SAR data can be right skewed (which is true in a certain amount). Therefore we normalize each image with log transformation and divide by maximum value as described in the paper. Notice that the band's values vary from negative to positive values, thus we can’t directly use the log function. So to each image we first add a constant so the pixel distribution will start at 1 (we keep the distribution). The constant is the min value + 1 (this is done only if there are values less than 1). We don’t mind shifting the distribution since we normalize it just after so the original values are replaced in anycase.

Hyper parameters and augmentations can be optimized separately.