Machine Learning Engineer Nanodegree

Capstone Proposal

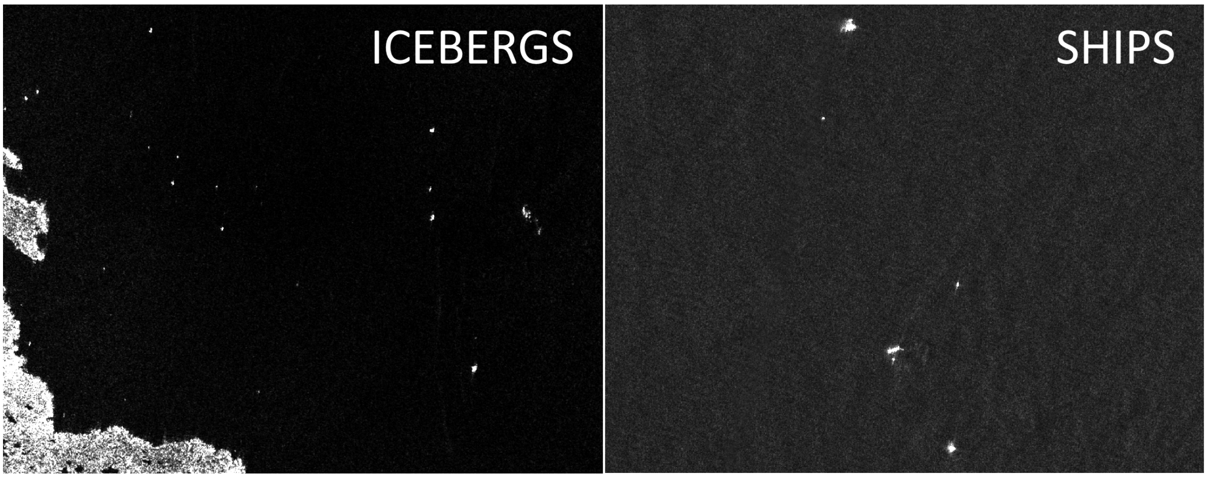
Satyanarayan Bhanja
October 28th, 2017

Proposal

Domain Background

The remote sensing systems used to detect icebergs are housed on satellites over 600 kilometers above the Earth. The Sentinel-1 satellite constellation is used to monitor Land and Ocean. Orbiting 14 times a day, the satellite captures images of the Earth's surface at a given location, at a given instant in time. The C-Band radar operates at a frequency that "sees" through darkness, rain, cloud and even fog. Since it emits it's own energy source it can capture images day or night.
Satellite radar works in much the same way as blips on a ship or aircraft radar. It bounces a signal off an object and records the echo, then that data is translated into an image. An object will appear as a bright spot because it reflects more radar energy than its surroundings, but strong echoes can come from anything solid - land, islands, sea ice, as well as icebergs and ships. The energy reflected back to the radar is referred to as backscatter.

When the radar detects a object, it can't tell an iceberg from a ship or any other solid object. The object needs to be analyzed for certain characteristics - shape, size and brightness - to find that out. The area surrounding the object, in this case ocean, can also be analyzed or modeled. Many things affect the backscatter of the ocean or background area. High winds will generate a brighter background. Conversely, low winds will generate a darker background.



Problem Statement

Drifting icebergs present threats to navigation and activities in areas such as offshore of the East Coast of Canada.
Currently, many institutions and companies use aerial reconnaissance and shore-based support to monitor environmental conditions and assess risks from icebergs. However, in remote areas with particularly harsh weather, these methods are not feasible, and the only viable monitoring option is via satellite.

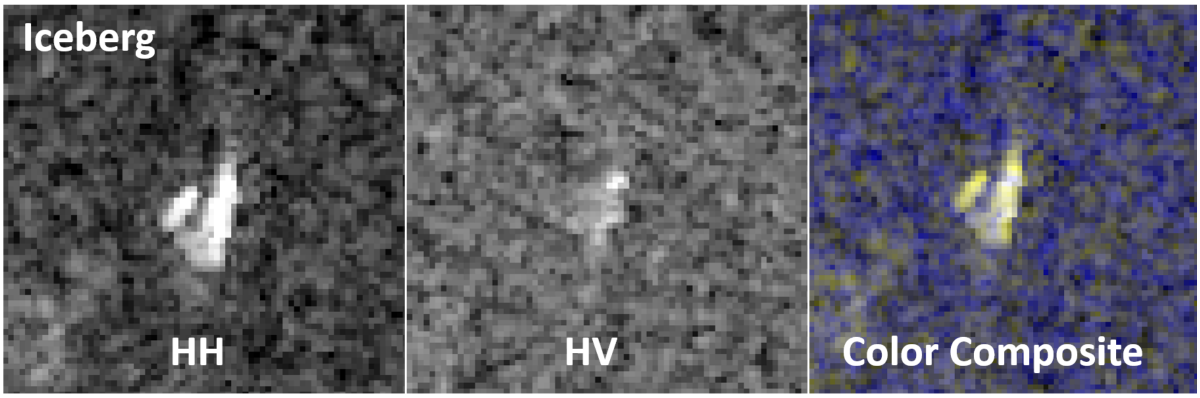
Problem is to predict whether an image contains a ship or an iceberg.

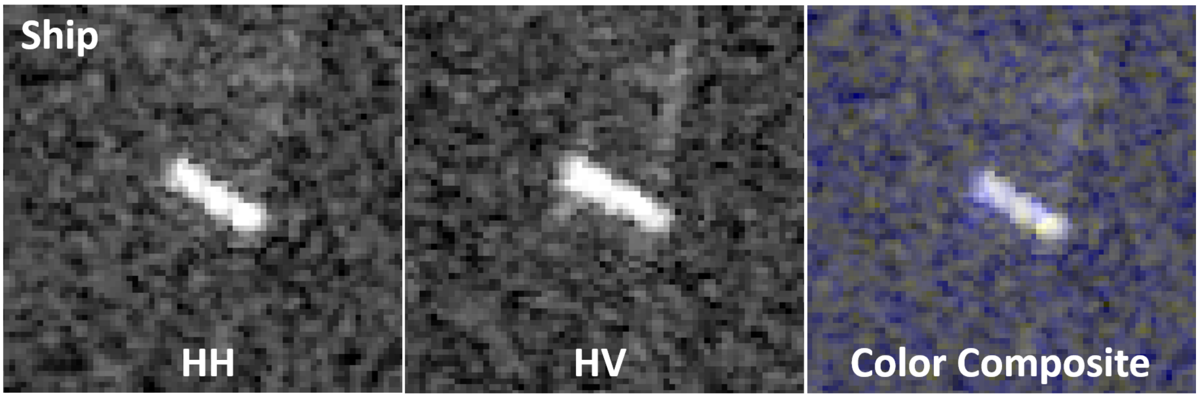
Datasets and Inputs

Datasets: train.json, test.json
The dataset is from kaggle competition https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/ (https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/)

The data (train.json, test.json) is presented in json format. The files consist of a list of images, and for each image, you can find the following fields:

- id - the id of the image
- band_1, band_2 - the flattened image data. Each band has 75x75 pixel values in the list, so the list has 5625 elements. Note that these values are not the normal non-negative integers in image files since they have physical meanings - these are float numbers with unit being dB. Band 1 and Band 2 are signals characterized by radar backscatter produced from different polarizations at a particular incidence angle. The polarizations correspond to HH (transmit/receive horizontally) and HV (transmit horizontally and receive vertically). More background on the satellite imagery can be found here.
- inc_angle - the incidence angle of which the image was taken. Note that this field has missing data marked as "na", and those images with "na" incidence angles are all in the training data to prevent leakage.
- is_iceberg - the target variable, set to 1 if it is an iceberg, and 0 if it is a ship. This field only exists in train.json.





Solution Statement

This is a image classification challenge i.e. to classify the images ship or iceberg.
The classifier model will be a CNN keras model with many layers like conv2D, maxpooling, dropout etc.

Benchmark Model

For this dataset,the labels are provided by human experts and geographic knowledge on the target, which is a manual task. The goal of this project is to build a ML model to classify the images automatically.

Evaluation Metrics

The avaluation metric to be used is **accuracy** to quantify the performance of both the benchmark model and the solution model.
Accuracy = (TP + TN)/(TP + TN + FP + FN)
TP = true positive
TN = true negative
FP = false positive
FN = false negative

Project Design

The project workflow as below,

- Read the datasets and remove NAs
- pre-process the data.
- train-test split the data into 75:25 for validation.
- Use CNN with Keras to classify the images.
- - The layers are conv2d, maxpooling, droput etc.
- Use keras fit() method to train the model.
- Then validate the model, on the validation set.
- Predict on the test data set and upload in kaggle to check public score.
- The solution file should be of format,
- - id - the id of the image
- - is_iceberg - your predicted probability that this image is iceberg.