MACHINE LEARNING NANODEGREE
Capstone Proposal
Statoil: Iceberg Classifier

Youssef Mubarak
October 01, 2018

1. Project Domain Background

Statoil, an international energy company operating worldwide, has worked closely with companies like C-CORE which has been using satellite data for over 30 years and has built a computer vision based surveillance system. To keep operations safe and efficient, Statoil is interested in getting a fresh new perspective on how to use machine learning to more accurately detect and discriminate against threatening icebergs as early as possible.

Satellite imagery is to be used to identify icebergs and differentiate them form ships, this process need to be automated using machine learning techniques.

2. 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.

So, we're challenged to build an algorithm that automatically identifies if a remotely sensed target is a ship or iceberg from satellite imagery. Improvements made will help drive the costs down for maintaining safe working conditions.

Note: This problem was presented in a Kaggle competition and you can find it here.

3. Datasets and Inputs:

The dataset is obtained as satellite imagery and the labels were provided by human experts and geographic knowledge on the target. All the images are 75x75 images with two bands.

The data is presented in *json* format as *train.json* and *test.json*.
The training data size is 1604, where the testing data size is 8424.

Data fields

The files consist of a list of images, and for each image, you can find the following fields:

	band_1	band_2	id	inc_angle	is_iceberg
0	[-27.878360999999998, -27.15416, -28.668615, -...	[-27.154118, -29.537888, -31.0306, -32.190483,...	dfd5f913	43.9239	0
1	[-12.242375, -14.920304999999999, -14.920363, ...	[-31.506321, -27.984554, -26.645678, -23.76760...	e25388fd	38.1562	0
2	[-24.603676, -24.603714, -24.871029, -23.15277...	[-24.870956, -24.092632, -20.653963, -19.41104...	58b2aaa0	45.2859	1
3	[-22.454607, -23.082819, -23.998013, -23.99805...	[-27.889421, -27.519794, -27.165262, -29.10350...	4cfc3a18	43.8306	0
4	[-26.006956, -23.164886, -23.164886, -26.89116...	[-27.206915, -30.259186, -30.259186, -23.16495...	271f93f4	35.6256	0

- **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.

The training data has 53% ships and 47% Icebergs where the ships has only 98 more images, so the data can be considered balanced.

You can find the dataset here:
https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/data

4. Solution Statement

As the testing data is not labeled we will split the training data into 75% training and 25% testing. Mainly, we will make use of Convolutional neural networks (CNNs) to address the problem of classification. Then we will make iterative refinements and improvements to the model in order to increase its performance.

5. Benchmark Model

The benchmark model will be a simple CNN classifier. Then we will try to improve the model performance by adding more layers and increasing its complexity. We will also consider submitting the final solution to the Kaggle competition to show where it stands among the scores in the competition leaderboard.

6. Evaluation Metrics

As the data is close to balanced we will mainly use ***Accuracy*** as an evaluation metric. We will also make use of classification report that shows the *precession*, *recall* and *f1-score*, but will focus on ***f1-score***.
So these are the metrics used to quantify the performance of both the benchmark and the solution models.

7. Project Design

1. *Data exploration*:
    - This includes reading, displaying data distribution and other visualization.
2. Data preprocessing:

    Includes but not limited to:
    - Dealing with missing values of data fields.
    - Combining the image bands to reconstruct the image from the bands values.
3. Implementation:
    - Starting by dimensionality reduction and using different classification models like: *Random Forest*, *K-NN*, and *Logistic Regression*.
    - Making use of Convolutional neural networks (CNNs) to address the problem of classification and building a simple CNN as a benchmark model and comparing its result to the previous models.
    - Increasing the complexity of the CNN model to improve its performance.
    - Applying data augmentation and seeing its effect on the model performance.
    - Applying transfer learning technique to see how the new model performs against previous models.
    - Finally, as the training data is relatively small compared to the unlabeled testing data we will apply a semi-supervised learning technique; pseudo-labeling to increase the training data and make use of the large unlabeled testing data. Then we pick the best obtained model and retrain it on the new data and analyze its performance***.***