# CS 229: Project Milestone

**Atharva Parulekar(atharva@stanford.edu)**
**Jun Li(junli530@stanford.edu)**
**Dhruv Samant(dhruvsam@stanford.edu)**

November 22, 2017

Dept. of Computer Science
Stanford University

# Contents

# 1. Motivation

Drifting icebergs present threats to navigation and activities in areas such as offshore of the East Coast of Canada. In this project, we build an algorithm that automatically identifies whether a remotely sensed target is a ship or iceberg. This project is inspired by the kaggle challenge posed by Startoil [3], an international energy company who have acquired satellite images by working closely with companies like C-CORE. Accurate detection of threatening icebergs will help drive the costs down, for maintaining safe working conditions.
Satellite radar bounces a signal off an object, records the echo and translates the data into an image. The object appears as a bright spot since it reflects more radar energy called backscatter than its surroundings. When the radar detects an object, it needs to be analyzed for specific characteristics- shape, size and brightness to classify it as an iceberg, or a ship or any other solid object. The Sentinel-1 satellite constellation, with remote sensing systems at over 600 km above the earth, transmits and receives energy in the horizontal and vertical plane, generating a dual-polarization image. In this project, we use raw images with two channels HH (transmit/receive horizontally) and HV (transmit horizontally and receive vertically).

# 2. Methods

- Principle Component Analysis with Support Vector Machines
- Computing statistics using thresholding and other operations.
- Convolutional Neural Networks (CNNs) Since the data is in the form of images, we implement CNNs that implement parameter sharing and sparse connections.

# 3. Experiments

## Data Visualization

We make a collage of the data with corresponding band maps shown side by side. We first zero centre the data by translating it such that the lowest element in the array is 0. We now extract some critical statistics and plot histograms using them.

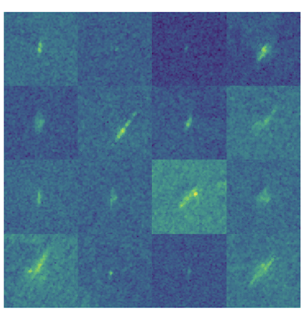


Figure 1: Iceberg HH

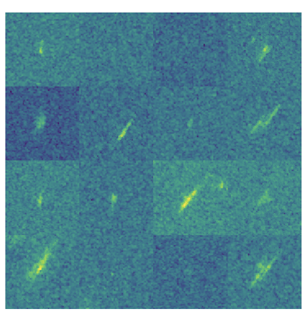


Figure 2: Iceberg HV

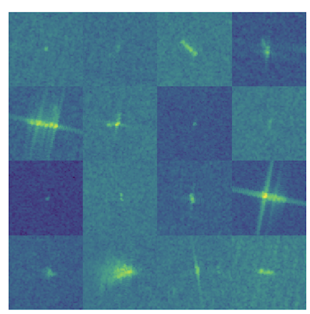


Figure 3: Ships HH

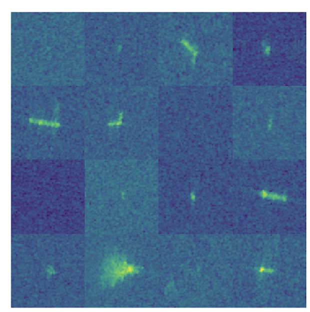
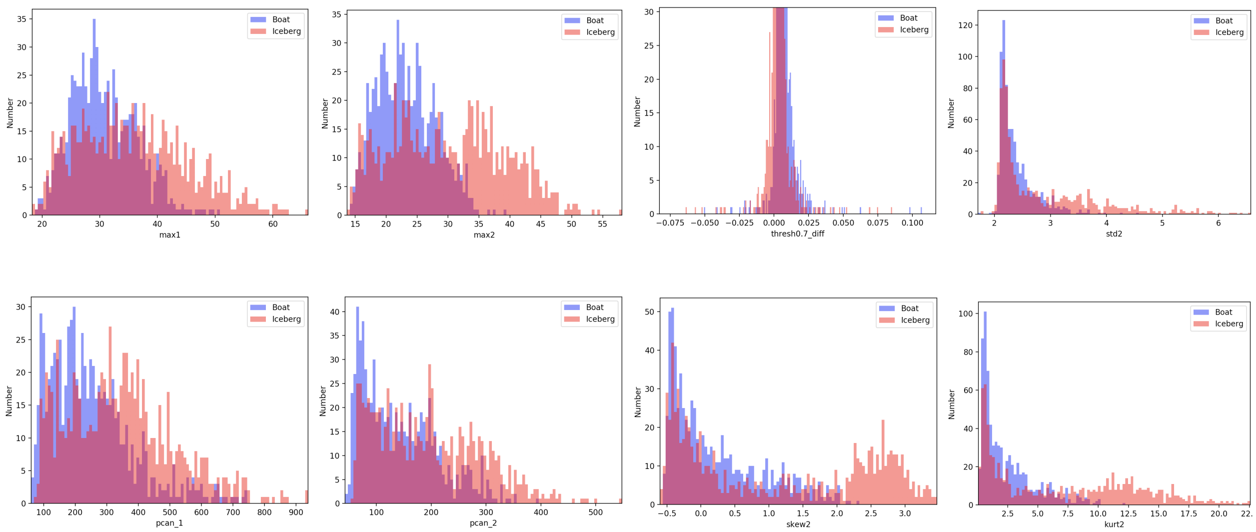


Figure 4: Ships HV

The following statistics have been used to compute histograms to visualize how the data separation occurs.

1. Maximum activation element in band HH and band HV
2. Intensity thresholding differences at 0.7 of maximum intensity on bands HH and HV
3. Standard deviation of the elements in HH and HV

4. Norm of the PCA on band HH and HV
5. Kurtosis and Skewdness of the data in HV




## Principal Component analysis on Data

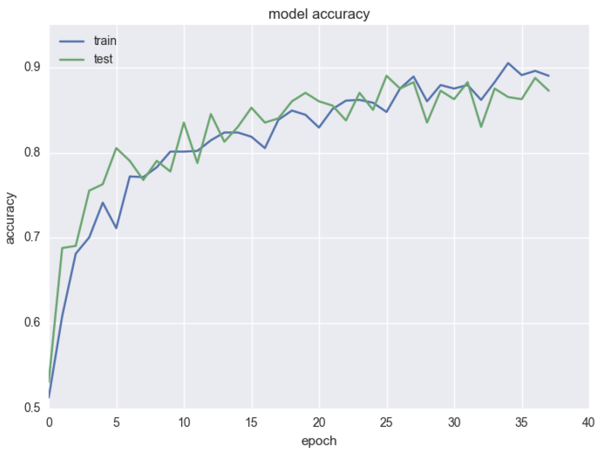
We first apply Principal component analysis on the data and train support vector machines on the PCA results. We compute the correlation of the first PCA component with the iceberg indicator random variable of the observations. We get the following correlation coefficients

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Band1 (PCA) correlation | -0.109 | -0.082 | 0.011 | 0.054 | -0.13 | -0.01 | 0.19 |
| Band2 (PCA) correlation | -0.13 | -0.22 | 0.09 | 0.029 | 0.059 | 0.11 | 0.08 |

## Convolution Neural Nets

We perform an experiment by running the following deep model on the data. We use a 0.2 dropout layer after every convolutional layer to avoid over fitting. The neural network has a fully connected layer at the end where we plan to feed inclination angle and other supplementary data.

| Architecture | | | | |
|---|---|---|---|---|
| Layer (type) | Filter / Neurons | Size (Depth) | Parameters | Output |
| 2D Convolution | 3 × 3 × 3 | 64 | 1,792 | 73 × 73 × 64 |
| Max pool | 2×2 | 0 | 0 | 36 × 36 × 64 |
| 2D Convolution | 3× 3 ×64 | 128 | 73,856 | 34 × 34 × 128 |
| Max pool | 2×2 | 0 | 0 | 17 × 17 × 128 |
| 2D Convolution | 3×3 × 128 | 128 | 1,47,584 | 15 × 15 × 128 |
| Max pool | 2×2 | 0 | 0 | 7×7 × 128 |
| 2D Convolution | 3×3 × 128 | 64 | 73,792 | 5×5 × 64 |
| Max pool | 2×2 | 0 | 0 | 2×2 × 64 |
| Flatten | 256 | 0 | 0 | 256 |
| Dense | 512 | 0 | 1,31,584 | 512 |
| Dense | 256 | 0 | 1,31,328 | 256 |
| Dense (Sigmoid) | 1 | 0 | 257 | 1 |
| Total | | | 560,193 | |





# 4. Future Steps

## Data Augmentation

We will use data augmentation to increase training set for our CNN training.[2]. Specifically we'll apply rotations in the multiple of 30 degrees, flips and shifts creating a data set 24 times as big.
Moreover, in order for the shape of images to be consistent and not creating any unusual boundaries after the rotations, we'll cut off the inscribed disk and rotate the disk area, while keeping the background the same. Since our algorithm focuses on the determining features between an iceberg and a boat, this augmentation must give us consistent results on a larger data set.

## Residual Neural Networks/Deeper Models

In order to increase the depth of our neural network while easing the training procedure, we will try residual neural networks to compare with [1]. This would potentially solve issues related to vanishing and exploding gradient and make our model more accurate after many iterations. Furthermore, we look to explore the inception architecture, combined with the resident idea of skip connections, to build even deeper neural nets.

## Exploiting symmetry

We observe that the HH bands in a ship form a symmetric pattern with rays emerging from the side of the ship. While the HV bands in a ship do not contain these rays. We also observed that the boats are more likely to possess a symmetric property, as opposed to icebergs. In the future we can take this feature into consideration, with the realization in mind that the centerpiece of a 75 by 75 image might not give as much information on symmetry as we expect.

## Using Filters

We also add some predefined filters like Sobel, Roberts and incorporate them in our res net models to check how performance is affected.

## Ensemble Models

Our final model will be an ensemble of deep neural nets which work in tandem and are trained separately on the augmented data. We then average out predictions to calculate the final validation.

# 5. Contribution

1. Atharva Parulekar: Data Visualisation, PCA, SVM, CNN models.
2. Dhruv Samant: CNN models, deeper models like residual networks.
3. Jun Li: Data Augmentation, Data Visualisation.

# References

[1] Kaiming Xiangyu Shaoqing Jian. "Deep Residual Learning for Image Recognition". In: *IEEE* (2015).

[2] Peter Corcoran Joseph Lemley Shabab Bazrafkan. "Smart Augmentation Learning an Optimal Data Augmentation Strategy". In: *IEEE* 5 (2017), pp. 5858–5869.

[3] Statoil/C-CORE. *Statoil/C-CORE Iceberg Classifier Challenge*. URL: `https://www.kaggle.com/c/statoil-iceberg-classifier-challenge`.