# Capstone proposal

## Domain Background

Icebergs presents threats to the ships navigation and various offshore activities. Especially, it as actual problem for the area offshore to Newfoundland and Labrador known as Iceberg Alley. The primary iceberg detection method for now is aerial reconnaissance using vessel-based monitoring data. Also, data received though satellites are widely being integrated now onto the monitoring systems greatly reduce monitoring cost. Additionally, Synthetic Aperture Radar (SAR) satellites can still monitor in various weather conditions such as clouds and fog.

However, manual visual classification of SAR images to identify iceberg is very time-consuming process. So, C‑CORE company (<https://www.c-core.ca/>) has developed a computer vision system that analyzes SAR data to automatically detect and classify icebergs and vessels. Now it challenges ML community to build effective classification algorithm for their detection system [1]:

## Problem Statement

The goal of the project is to build an algorithm which can reliably classify data to identify either it is iceberg or ship, based on given Synthetic Aperture Radar data. Also, the results is clearly measurable as it is important to have classifier with higher accuracy (ideally 100%).

Also, analysis and classification SAR data is interesting. Even if it seems like standard image classification task it has some important differences which makes it challengeable to use pre-trained neural networks with transfer learning for the image classification such as VGG [2] or Inception [3]:

* SAR data is not a three-channels regular image
* Radar detected shapes are different than visually detected shapes.
* Data set has additional incidence angle parameter of which the image was taken. So, it is additional interesting task to figure out how to include this into the model and how beneficial this parameter is.

However, as SAR data is still like image data – the potential solution is to use custom CNN architecture which fits SAR data best.

## Datasets and Inputs

CORE provided dataset of 5000 satellite SAR images containing either a ship or an iceberg.  SAR bounces a signal off an object and records the echo, then that data is translated into an image. Two channels of image are provided: HH (transmit/receive horizontally) and HV (transmit horizontally and receive vertically) [4].

There are two datasets available:

* Training dataset with n items for the model training and validation.
* Test dataset is used for the model evaluation. The result for test dataset is displayed in the Kaggle leaderboard.

(approx. 2-3 paragraphs)

I

n this section, the dataset(s) and/or input(s) being considered for the project should be thoroughly described, such as how they relate to the problem and why they should be used. Information such as how the dataset or input is (was) obtained, and the characteristics of the dataset or input, should be included with relevant references and citations as necessary It should be clear how the dataset(s) or input(s) will be used in the project and whether their use is appropriate given the context of the problem.

## Solution Statement

(approx. 1 paragraph)

An algorithm should be designed to

In this section, clearly describe a solution to the problem. The solution should be applicable to the project domain and appropriate for the dataset(s) or input(s) given. Additionally, describe the solution thoroughly such that it is clear that the solution is quantifiable (the solution can be expressed in mathematical or logical terms) , measurable (the solution can be measured by some metric and clearly observed), and replicable (the solution can be reproduced and occurs more than once).

## Benchmark Model

We will be using two benchmark models. As it is Kaggle competition the score of the best models is publicly available – so we can compare our model performance with the Kaggle leaderboard to check how well our model performs. The goal for my first Kaggle project it to create model which can be in the first 30% results.

As no model architecture is publicly available for the top Kaggle models it would be great to have some real model to compare with which has both model architecture defined and log loss score. The good candidate is simple CNN implemented with Tensorflow with the log loss = 0.27. So, we can see if we can achieve better with the proposed algorithm.

## Evaluation Metrics

As this problem is a standard binary classification problem and while considering problem domain it is obvious that accuracy of the prediction is an essential metric. It is just calculated as a fraction of correctly classified images over database size.

Additionally, we are going to use Log loss [3] as additional evaluation metric. Firstly, because it is official evaluation metric for the Kaggle completion and its value available for the benchmark model and can be compared. Secondly, it provides additional information compared to accuracy: how “sure” model is about its prediction. Lower log loss corresponds to models which more solid prediction boundary even if accuracies of the models are close.

## Project Design

(approx. 1 page)

It

As it is binary classifier

The first step which might be beneficial

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

## References

[1] Kaggle competition: <https://www.kaggle.com/c/statoil-iceberg-classifier-challenge>

[2] VGG: <http://www.robots.ox.ac.uk/~vgg/research/very_deep/>

[3] Inception: <https://arxiv.org/pdf/1409.4842.pdf>

[4] Log loss: <http://www.exegetic.biz/blog/2015/12/making-sense-logarithmic-loss/>

[5] Benchmark model: <https://www.kaggle.com/camnugent/convolutional-neural-net-tensorflow-lb-0-27>

[6] Kaggle leaderboard: <https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/leaderboard>