Deep learning approach to automatically classify seafloor type

Jean-Michel Amath Sarr, Timothée Brochier

Institut de Recherche pour le Développement FRANCE

LUX MEA LEX SIGILLUM UNIVERSITATIS DAKARENSIS

Outline

Background and Motivation

- Data acquisition
- Context
- Errors

Methodology

- Preprocessing
- Bottom Classification 1
- Bottom Classification 2
- Full Data Flow System
- Machine Learning Design Process
- Learning Under Covariate Shift

Results

- Training Accuracy
- Test Accuracy

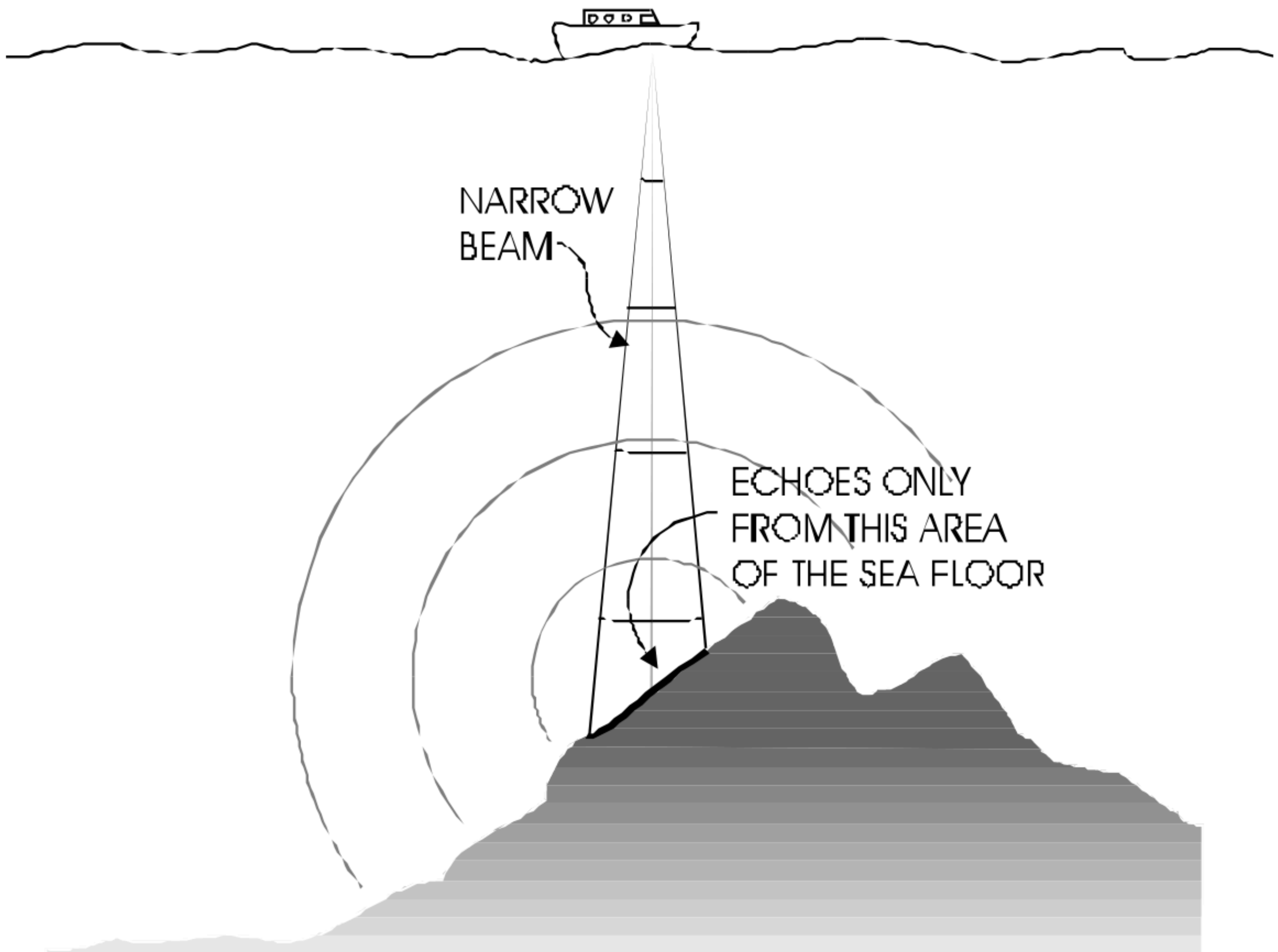
References

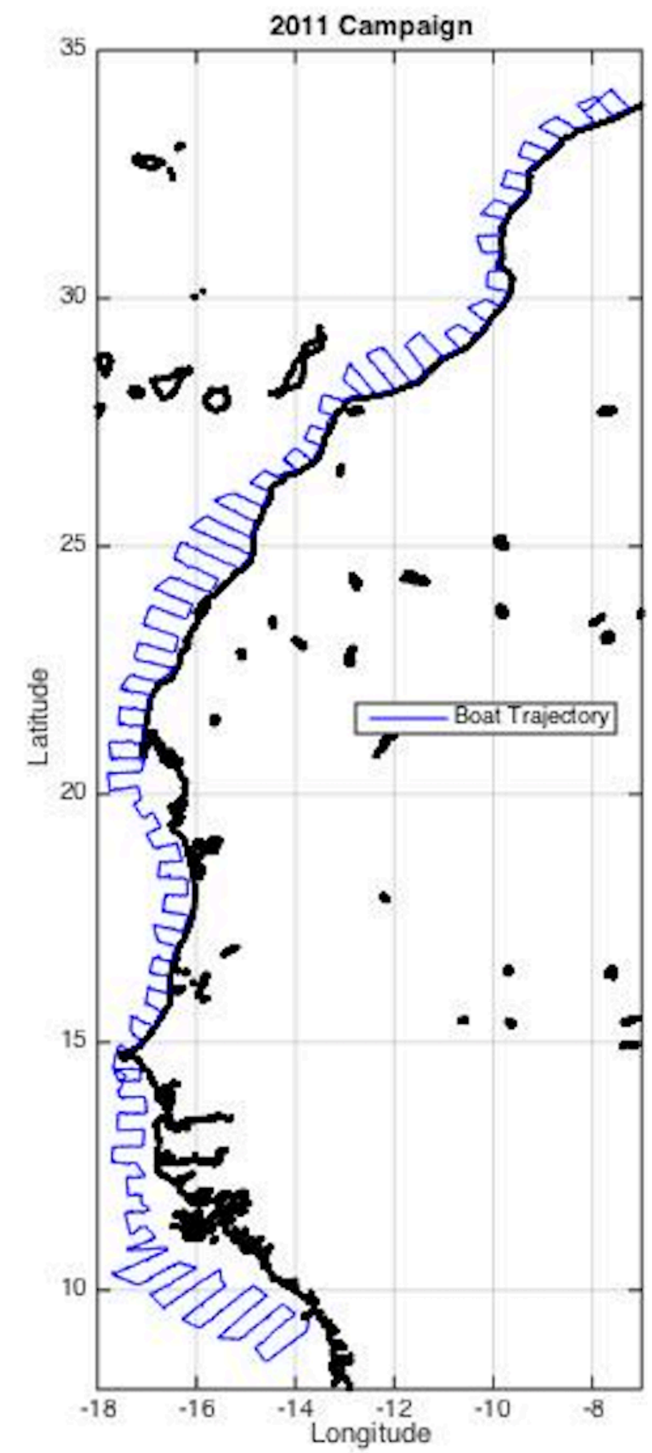
- Bibliography
- Tools

Background and Motivation

Data acquisition

Data are acquired with an echo sounder. The vessel sends out pulses of various frequency's acoustic waves in the water, those waves are reflected back to the source when they meet diverse organisms (fish, plankton, etc) or more generally solid objects. We call echogram (echo more informally) the corresponding signal.

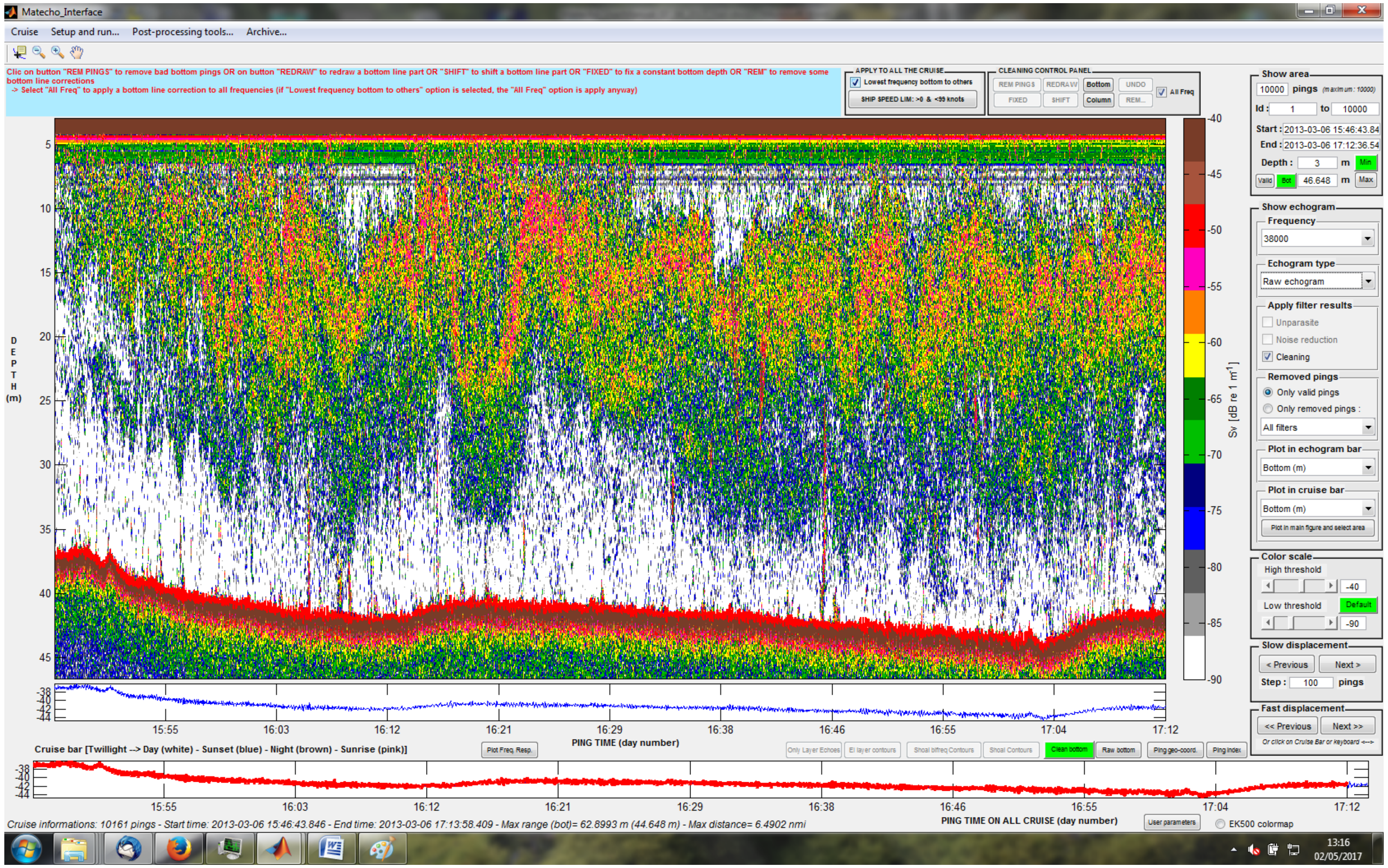




Background and Motivation

Context

- Echo-integration is a method for fish stock assessment, it plays a critical role to derive fishery management policies in order to prevent overfishing.
- Nowadays the task is done in a semi automatic way, and is very time consuming.
- When errors are made it can significantly hurt the stock evaluation which make the campaign meaningless

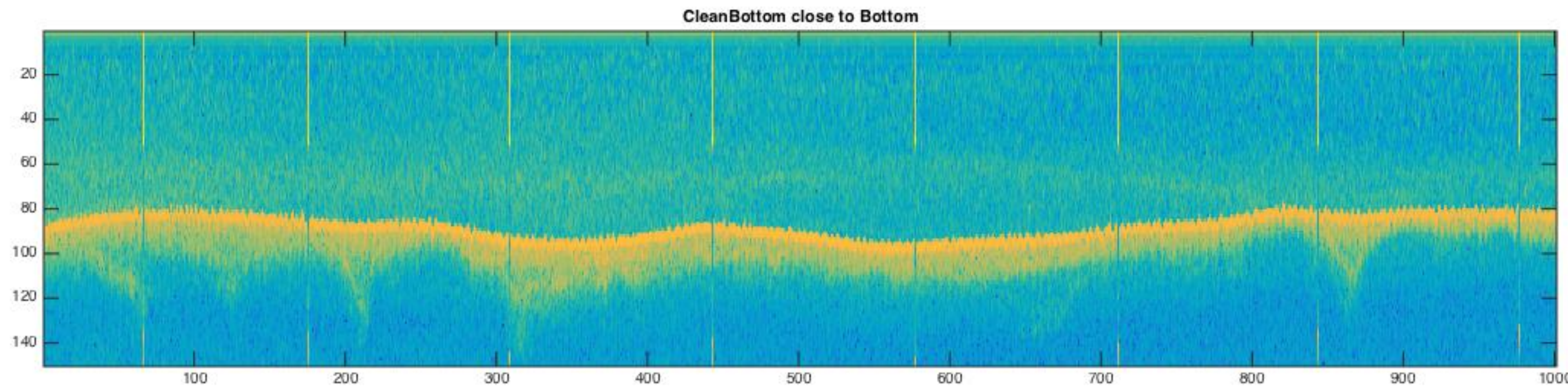


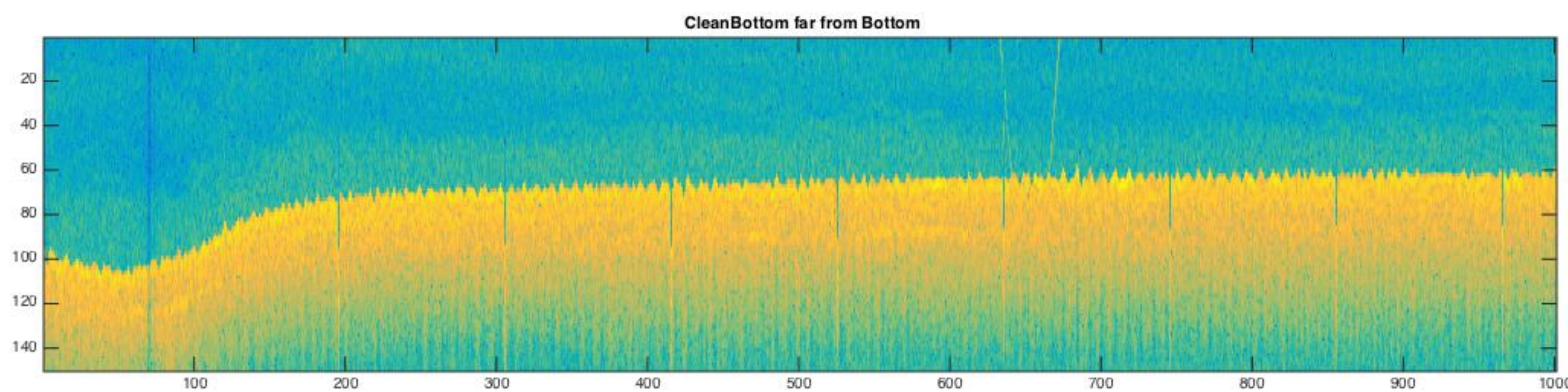
Background and Motivation

Errors

Often the errors made by automatic treatment of the echogram are due to their **inability to accurately predict the bottom depth** when **high density of fish are present close to the seabed**. Hence human experts are expected to inspect the whole echogram to correct the bottom depth estimation of the algorithm.

GOALS: Classify the full echogram in **two classes: strong bottom** and **unclear bottom**

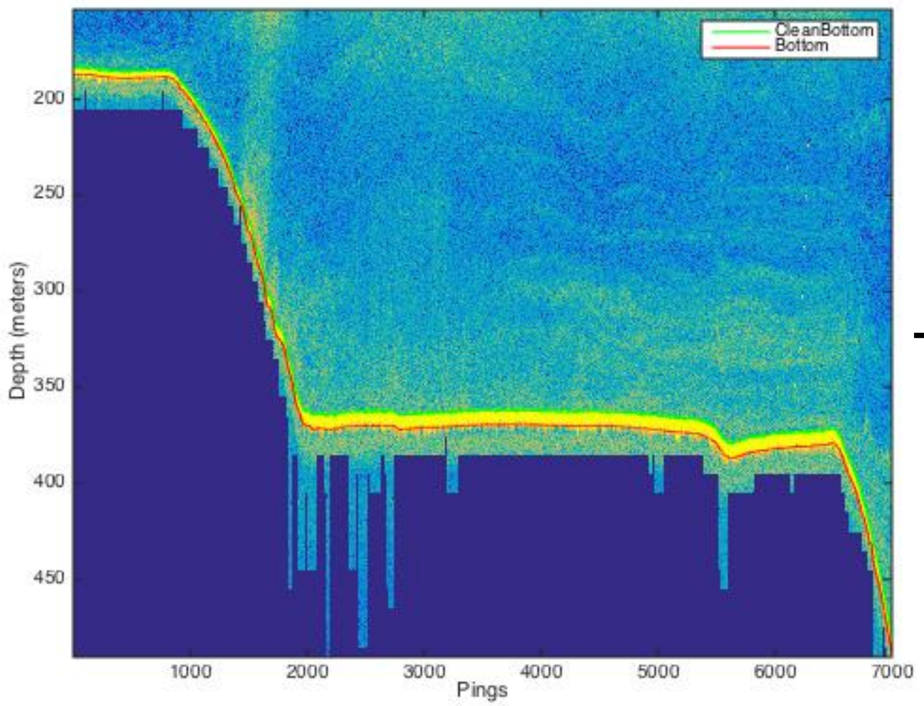




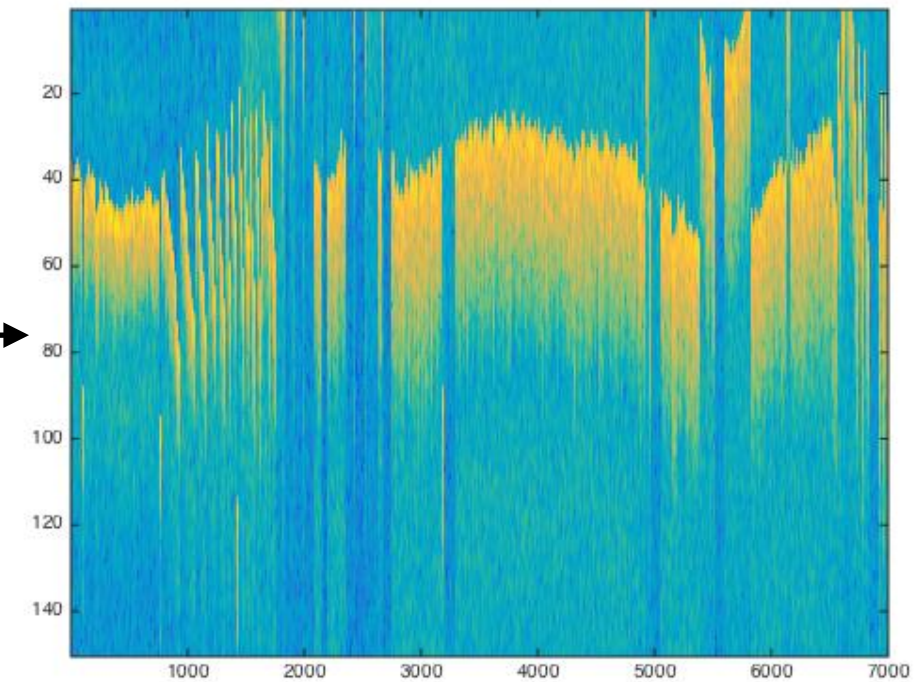
Methodology

Preprocessing

Full Echogram



35m of Echogram



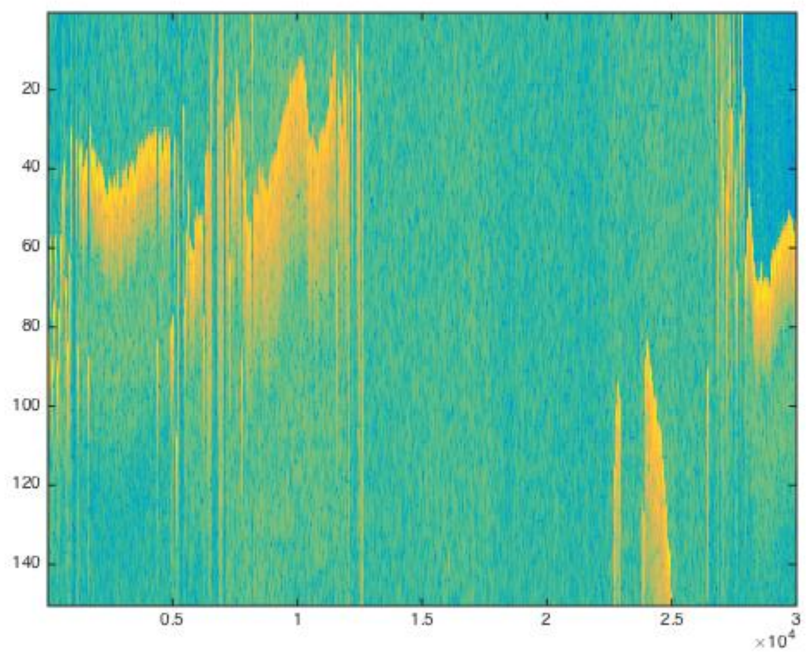
We only select the parts of the full echogram 35 meters above Nans, because:

- there is **too many cells with Nans** and it **destabilize the learning process** even if we replace them with sounded numerical values.
- The full dataset is **BIG: 2,661,003 rows and 2,581 columns**, so with 35 meters of echogram we **reduce the size down to 150 columns**.

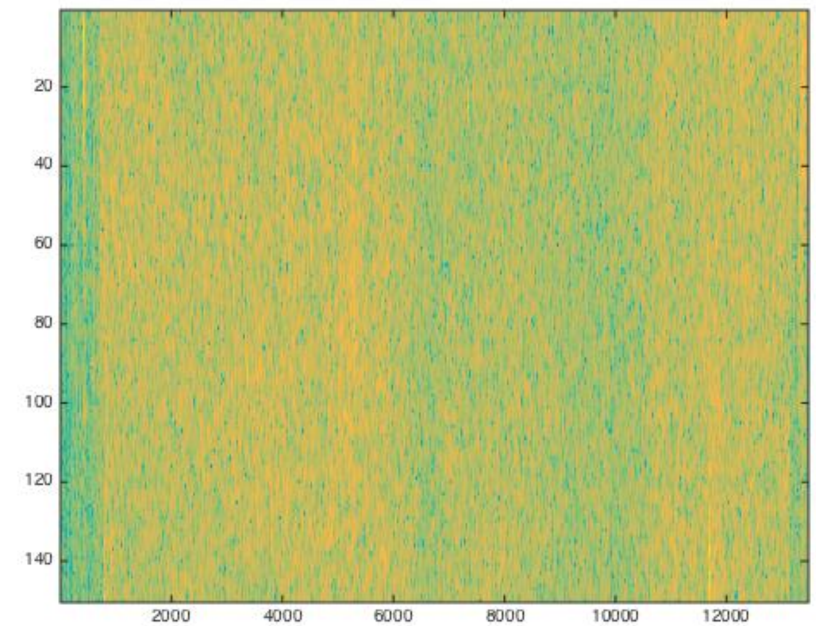
Methodology

Bottom Classification 1

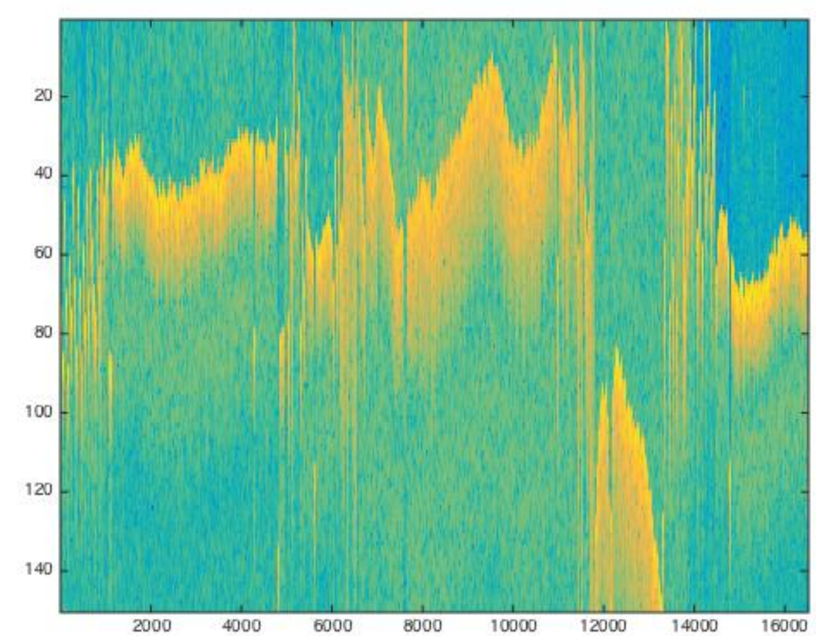
35m of Echogram



Bottom not present



Bottom present

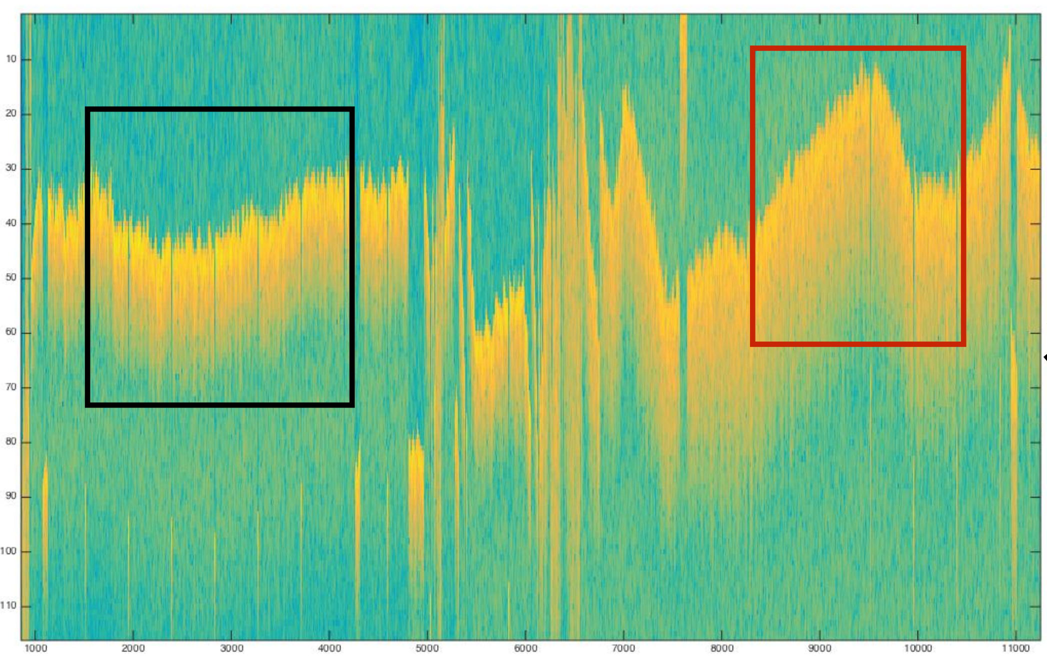


Echograms are first classified based on the presence or absence of signal representing the bottom.

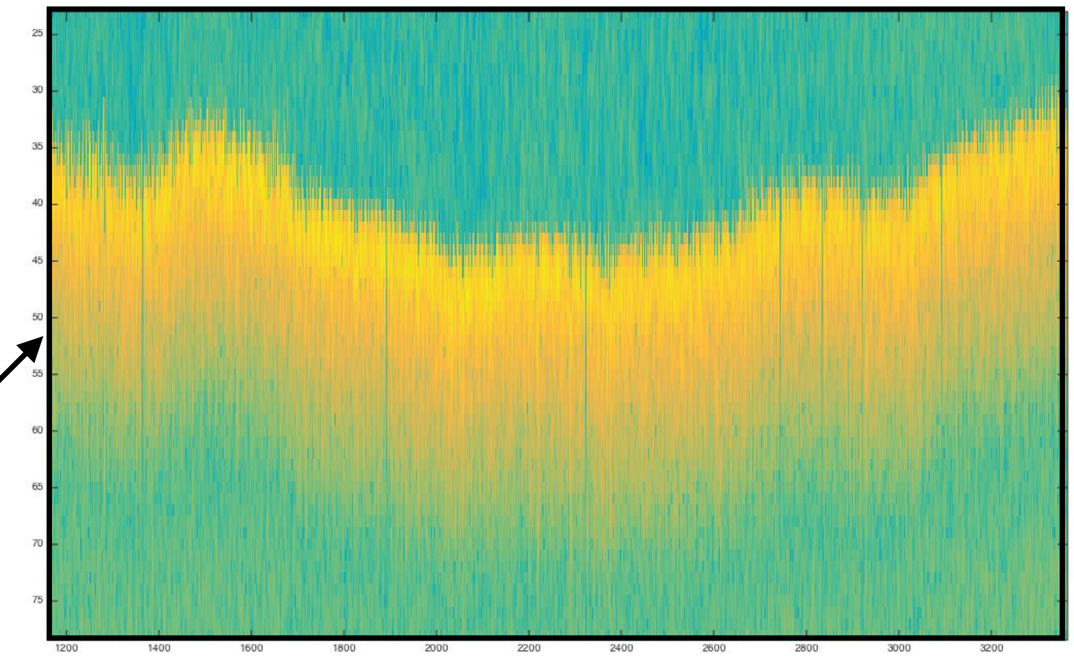
Methodology

Bottom Classification 2

Data are classified based on the Bottom found automatically and the Bottom corrected by the expert. There is two types of echogram

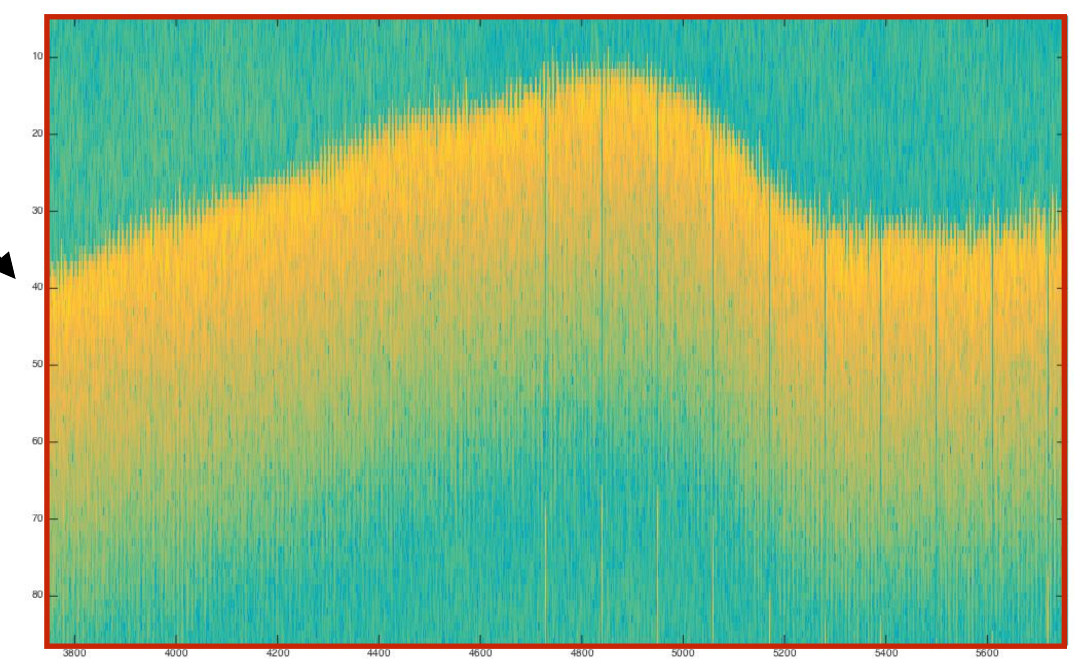


Strong Bottom



The bottom is found by the based algorithm

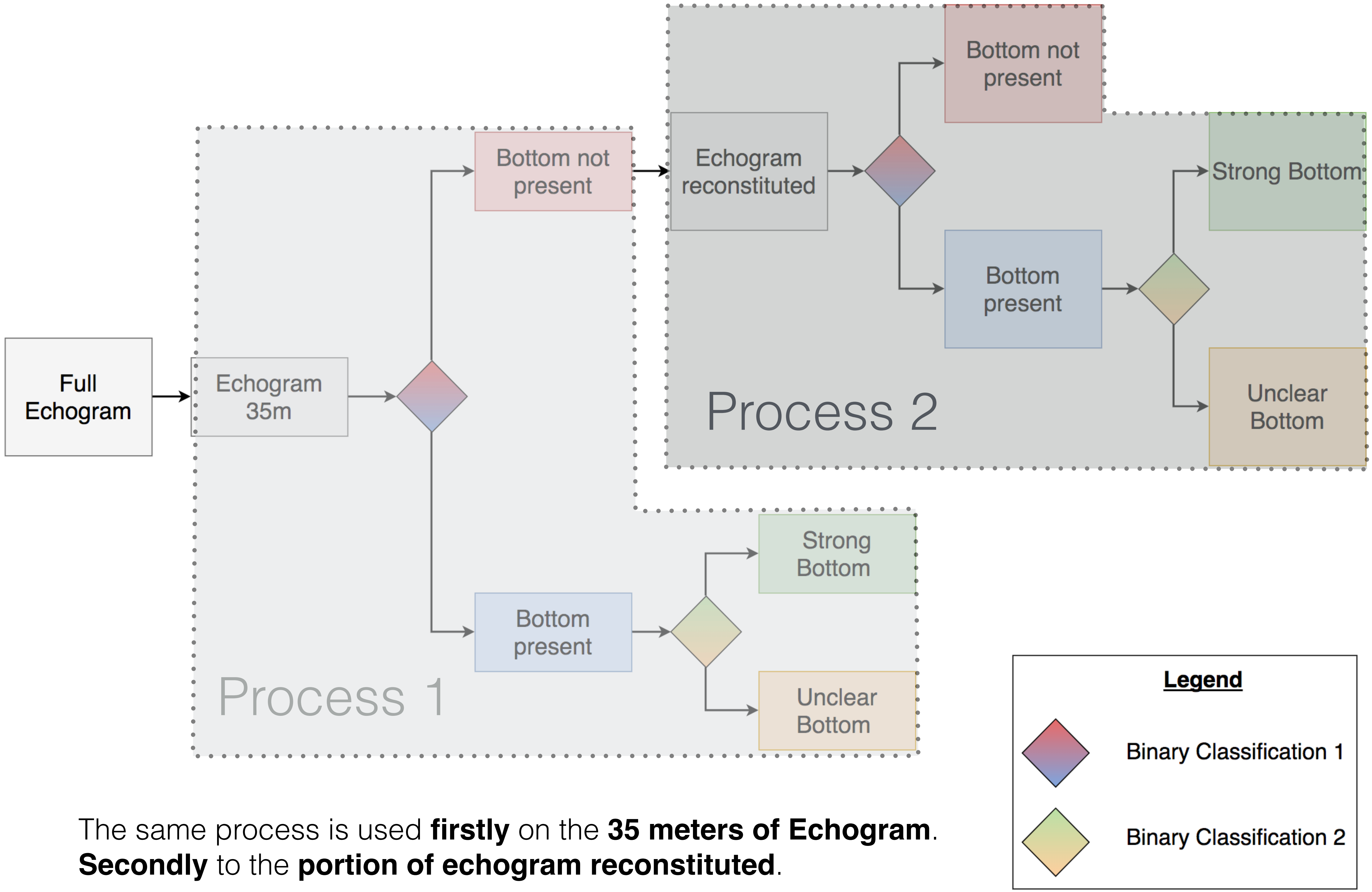
Unclear Bottom



The bottom found by the algorithm lag deeper than where the real bottom is.

Methodology

Full Data Flow System



The same process is used **firstly** on the **35 meters of Echogram**.
Secondly to the **portion of echogram reconstituted**.

Methodology

Machine Learning Design Process

Echogram 35m

Fully Connected 3 hidden layers Neural Networks, with binary cross entropy loss were used.
The **number of weights, learning rate and l2-regularization parameters** were found using **Bayesian Optimisation**
for each:

- Binary Classification 1
- Binary Classification 2

Echogram Reconstituted

We used **1-d Convolutional Neural Networks** with **3 convolutional hidden layers** followed by **2 and 3 Fully Connected layers**, and a binary cross entropy loss. The **number of weights, learning rate and l2-regularization parameters** were found using **Bayesian Optimisation**
for each:

- Binary Classification 1
- Binary Classification 2

Learning data

2011
Campaign

Test data

2015
Campaign

Covariate Shift

In general we expect the training and test data to come from the same distribution. However due to differences in the vessel sensors settings, the **distribution of classes** from 2011 and 2015 was very **different** for the **Echogram Reconstituted.** So the **Process 2** needed to **learn under covariate shift**

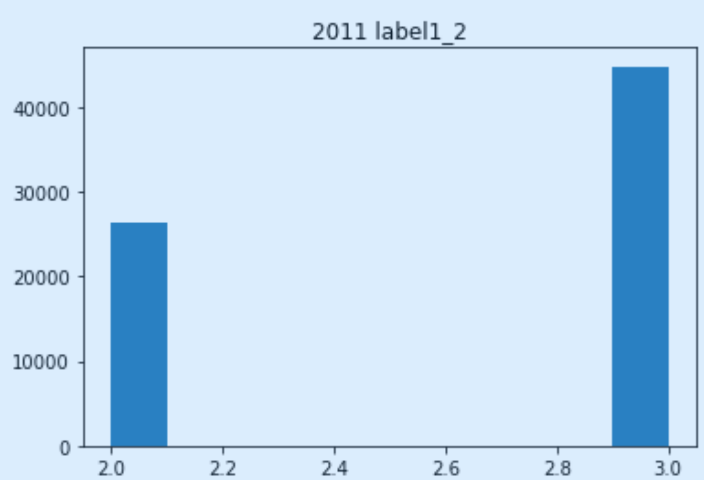
Methodology

Learning Under Covariate Shift

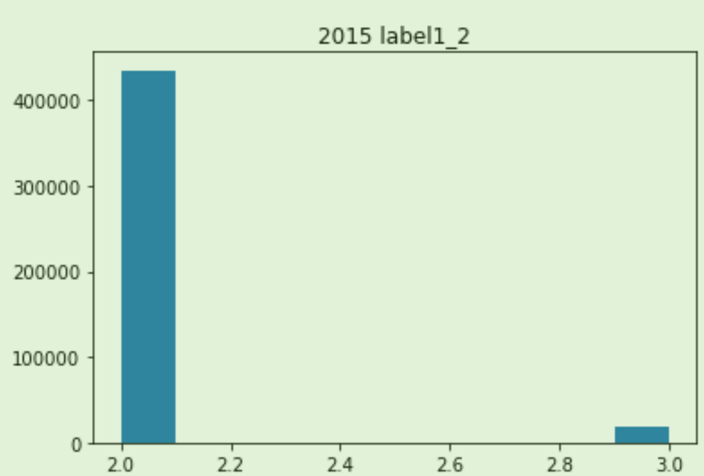
Process 2

Binary Classification 2

Training data distribution



Test data distribution



Let $T = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ be the training set, $x^{(i)} \in \mathbb{R}^d$, following $p_{train}(x)$ the probability density function generating the training data, $y^{(i)} \in \{-1, 1\}$ the label associated following a conditional distribution $p(y|x)$. $\hat{f}(x, \theta)$ the neural network class prediction for an input x, $\theta \in \Theta$ the parameter space, and $l(x, y, \hat{f}(x, \theta))$ the binary cross entropy loss function Then the **Empirical Risk Minimizer**:

$$\hat{\theta}_{ERM} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{1}^{n} l(x^{(i)}, y^{(i)}, \hat{f}(x^{(i)}, \theta))$$

is not **consistent under covariate shift**. i.e if we assume $p_{train}(x) \neq p_{test}(x)$ and that the conditional distribution are the same

$$\lim_{n \to \infty} \hat{\theta}_{ERM} \neq \theta^*$$

with θ^* the optimal estimator. Instead we introduce the **Importance Weighted Empirical Risk Minimizer** (IWERM)

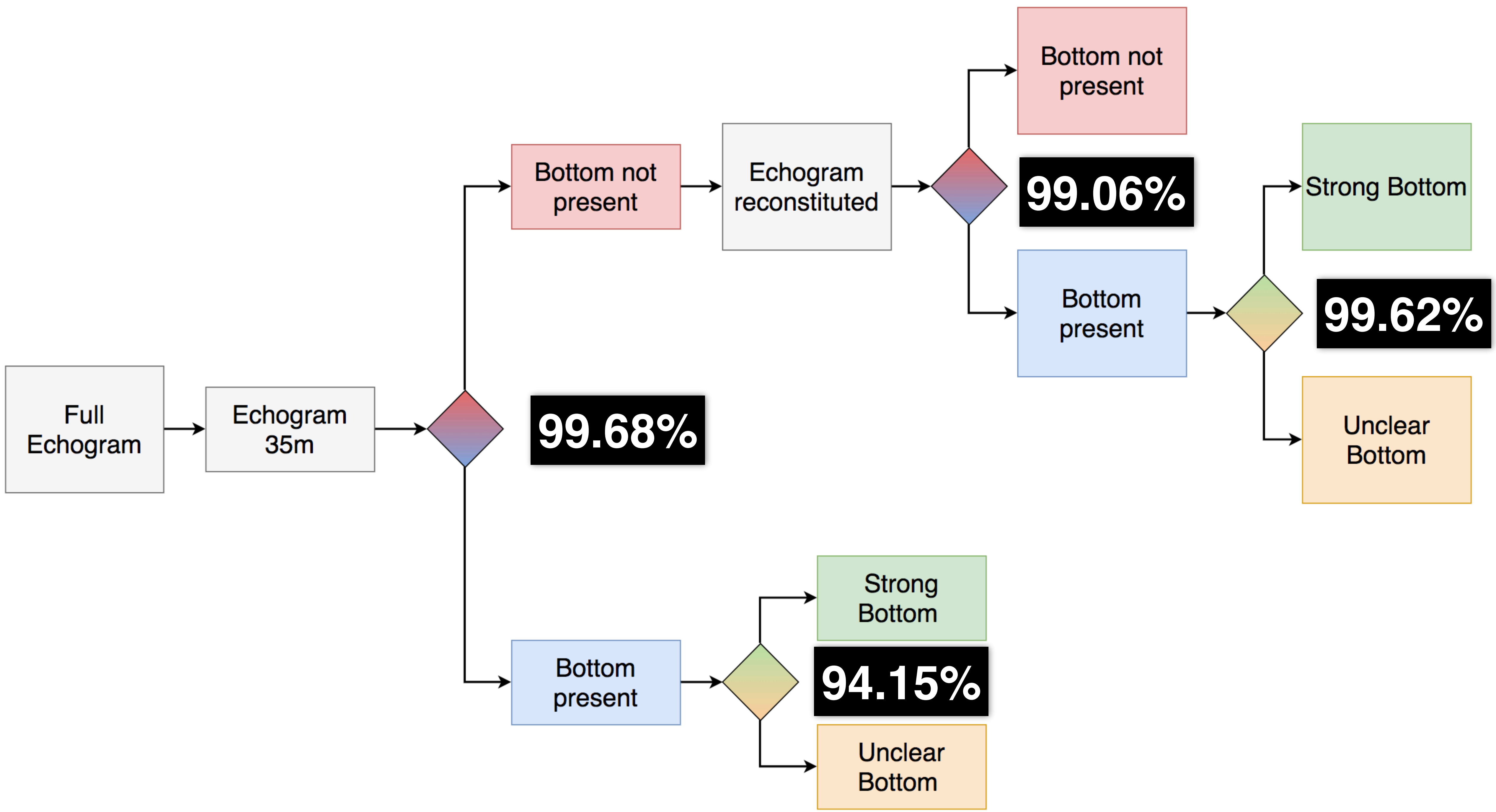
$$\hat{\theta}_{IWERM} = \arg\min_{\theta \in \Theta} \left[\frac{1}{n} \sum_{1}^{n} \frac{p_{test}(x^{(i)})}{p_{train}(x^{(i)})} l(x^{(i)}, y^{(i)}, \hat{f}(x^{(i)}, \theta)) \right]$$

which is **consistent under covariate shift**

$$\lim_{n \to \infty} \hat{\theta}_{IWERM} = \theta^*$$

Results

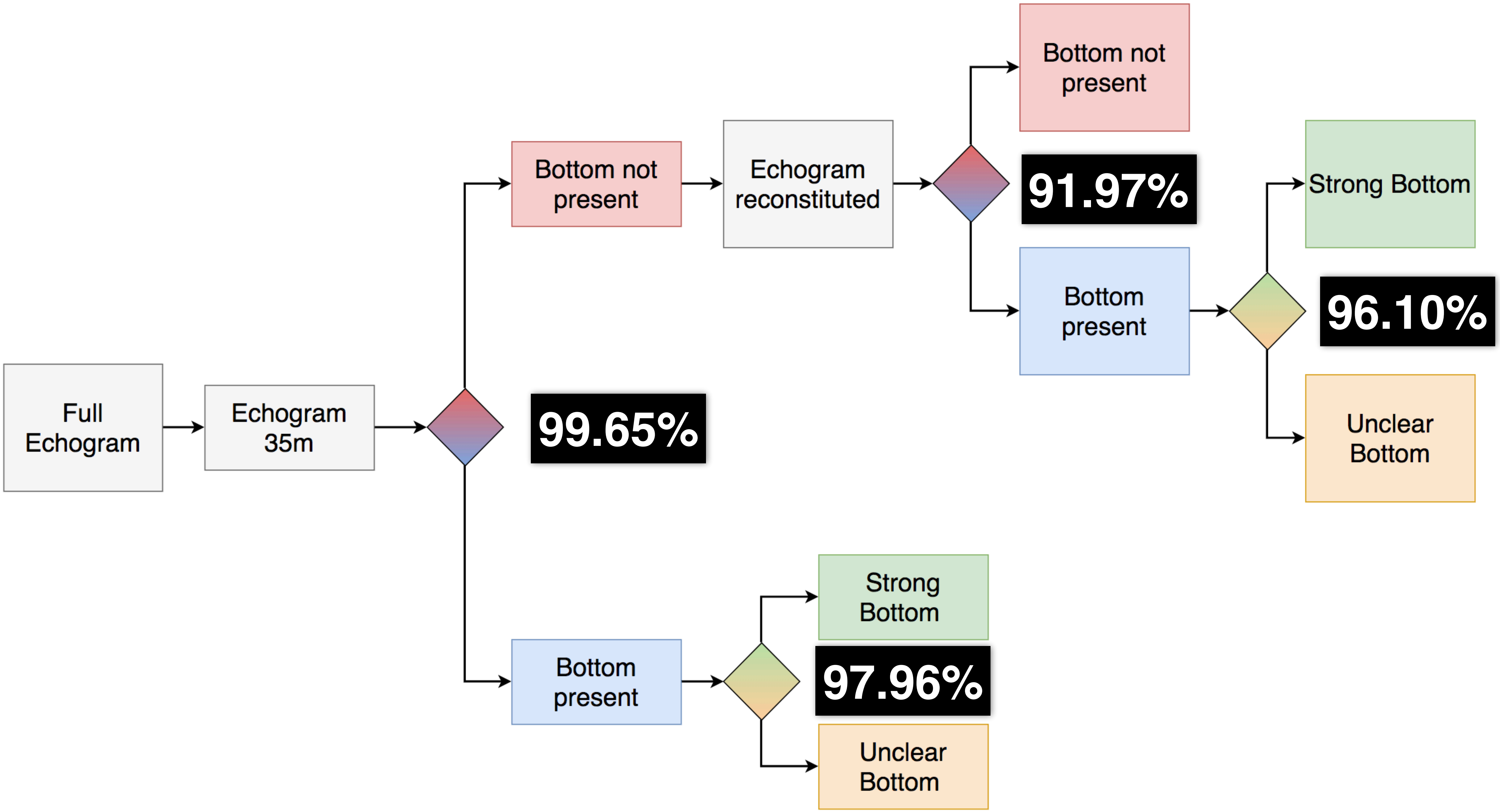
Training Accuracy



Total System Accuracy: 92.63%

Results

Test Accuracy



Total System Accuracy: 86.29%

References

Bibliography

- Verena M. Trenkel, Laurent Berger, Sébastien Bourguignon,et al., *Overview of recent progress in fisheries acoustics made by Ifremer with examples from the Bay of Biscay,* Aquat. Living Resour, EDP Sciences, IFREMER, IRD 2009

- Ian Goodfellow and Yoshua Bengio and Aaron Courville, *Deep Learning,* 2016, MIT Press

- Jasper Snoek, Hugo Larochelle, *Practical Bayesian Optimization of Machine Learning Algorithms,* ARXIV, 2012

- Masashi Sugiyama, Matthias Krauledat, Klaus-Robert Muller, *Covariate Shift Adaptation by Importance Weighted Cross Validation,* Journal of Machine Learning Research, 2007

Tools

- Python 3: Keras, GyOpt, Numpy, Scipy,
- Matlab
- Floydhub