CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

# Deep Learning for Deep Water

## Robust classification of ship wakes with expert in the loop

Master's thesis in Computer science and engineering

IGOR RYAZANOV

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

# Deep Learning for Deep Water

Robust classification of ship wakes with expert in the loop

IGOR RYAZANOV

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

Deep Learning for Deep Water
Robust classification of ship wakes with expert in the loop
IGOR RYAZANOV

© IGOR RYAZANOV, 2020.

Supervisor: Debabrota Basu, Department of Computer Science and Engineering
Examiner: Alexander Schliep, Department of Computer Science and Engineering

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

Deep Learning for Deep Water
Robust classification of ship wakes with expert in the loop
IGOR RYAZANOV
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

# Abstract

This work examines the applicability of the deep learning models to pattern recognition in acoustic ocean data. The features of the dataset include noise, data scarcity and the lack of labeled samples. A deep learning model is proposed for the task of automatic wake detection. It takes advantage of the availability of an expert in the marine science domain while using data generation and robustness techniques to enhance performance. The model shows encouraging results, although its performance decreases with heavily unbalanced data and the introduction of noise.

Keywords: machine learning, deep learning, pattern recognition, acoustic data analysis, shipping data, data augmentation, noise robustness, classification with data imbalance, expert-in-the-loop framework.

# Acknowledgements

I would like to thank the people who made working on this thesis a wonderful experience: my supervisor, Debabrota Basu, who guided and supported me over the past several months, Alexander Schliep, who invited me in the project and provided a lot of helpful feedback during its course, Amanda Nylund and Ida-Maja Hassellöv, the experts in maritime science who proposed the research idea and were always there to help with their domain knowledge.

Igor Ryazanov, Gothenburg, June 2020

# Contents

# List of Figures

# List of Tables

# 1
# Introduction

The ocean is the largest ecosystem on Earth. Over the years, it has been affected in numerous ways by humanity. The threats include climate change, pollution, uncontrolled resource extraction, and many others. Preserving oceans and ocean life is a global challenge, which is present, for example, on the list of sustainability goals in the United Nations 2030 Agenda for Sustainable Development.

Some of the most affected areas of the ocean are the ones surrounding the shipping lanes. Intense shipping changes both the chemical and physical characteristics of ocean water. One of the sources of disturbance is ship wakes, traces of waves and bubbles left behind passing vessels. Sometimes they can stretch for kilometers. The wakes are known to affect the surface of the ocean, but the turbulence can reach deeper layers as well. This is supported, for example, by the fact that the water temperature in the wakes is significantly lower than in the surrounding area; cool water is brought up to the surface from deeper sea [8].

Research on the ship-induced water mixing is being conducted at the Department of Mechanics and Maritime Sciences of the Chalmers University of Technology. For its purposes, a four-week dataset of measurements was collected by a five-beam Acoustic Current Doppler Profiler (ACDP). The ACDP produces an echogram, in which bubbles, waves, fish schools, and other objects are visible. A one-hour echogram is shown in Figure 1.1.

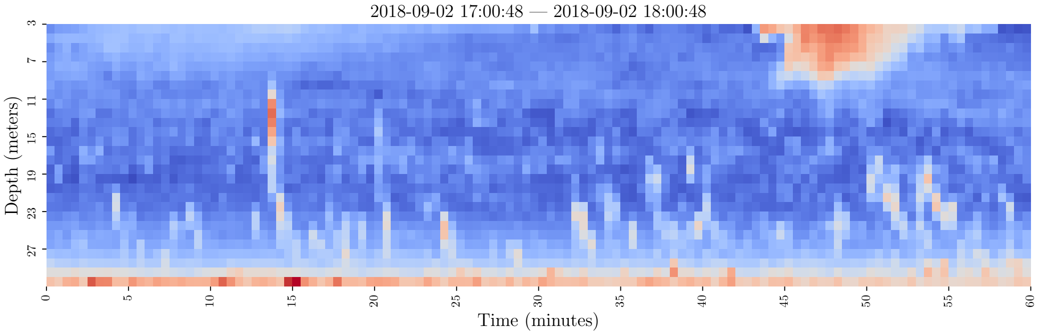

**Figure 1.1:** One hour (17:00 to 18:00 on the 2nd of September 2018) of observations collected by the central beam plotted. The color shows the intensity of the signal. For objects such as bubbles and plankton, the signal is stronger and plotted in red. Blue means a weaker signal and no detectable objects. A ship wake is visible at the timestamp 17:43.

For the echogram to be further investigated in the research, detected objects need to be recognized and labeled. Manual labeling of the data presents serious difficulties, and the main purpose of this project is to provide a tool for identifying ship wakes in the acoustic data. Machine learning algorithms are a good instrument to solve this problem as they are known to perform well on object recognition tasks. Wake detection can be treated as a classification problem. If the data is divided into frames of fixed size, they can be labeled as either containing wakes or empty. A machine learning model is then trained on the available data to recognize wakes. This model can later be used for the incoming unlabeled data, which is broken down into frames of the same size.

The main goal of this project was to create a tool that is capable of solving the task for future observations using a very limited set of labeled data. The challenges came from the lack of data and its quality: recorded data is noisy because of the wakes, wind, plankton, and other objects. During the development process, the expert-in-the-loop approach was taken. The number of labels in the available dataset is limited to the few wakes identified by an expert, and the development is largely based on the expert's initial evaluation. The labels were used to generate more samples, which were, in turn, approved by the expert, utilizing a feedback loop. A deep learning model was then trained on the populated dataset. To achieve better robustness, two possible modifications of the machine learning model were implemented: example reweighting and randomized smoothing. While the results for the default and reweighting was encouraging, the randomized smoothing algorithm did not show the expected improvement. The performance of the three models, as well as the potential reasons behind the achieved results, are discussed further in sections 4 and 5 of this paper.

# 2

# Background

This chapter explains the reasoning behind the choice of methods, as well as some approaches that have been considered but were eventually discarded. Firstly, the application of machine learning to related tasks is discussed explaining the general strategy. Secondly, the machine learning methods applied in this project are described. Finally, several approaches to address the lack and the quality of data are shown.

## 2.1 Application of machine learning to marine science

The problem of underwater wake recognition is to the best of our knowledge novel. Moreover, applying machine learning to the data collected from deep-sea devices is not very common in general. One of the reasons is that collecting a large amount of reliable data has been problematic until recently.

### 2.1.1 Related work

Research has been done on the automatic detection of the wakes in 2D-images of the ocean surface [3]. However, this approach does not consider the depth of the area affected by the turbulence.

Brautaset et al. recently applied machine learning techniques to a closely related problem — fish schools acoustic classification [1]. The data used in the paper is collected by an acoustic device and is very similar to the observations collected for the ship wake research project. Visual representations provided in the paper are also very close to what can be seen in Figure 1.1. The authors apply a Convolutional Neural Network (CNN) model to their problem and achieve good results for isolated regions in the echogram. This is the only known related paper that investigated a problem that could be directly mapped to the task of this project. The results achieved in it show that the convolutional neural network-based approach is applicable to similar tasks.

However, there is a difference between the two problems in the nature of target objects. Fish schools are relatively compact and produce similar traces in the echogram regardless of their relative position to the ACDP beams. Ship wakes, on the other hand, can affect the surface of the ocean for kilometers, and the echogram of the wake can change significantly depending on the distance to the ship, beam angle, and weather conditions.

### 2.1.2 Expert-in-the-loop

Expert-in-the-loop is a framework for AI training where an expert oversees the training and gives additional feedback based on the results after a training iteration. It can be beneficial in terms of both achieving better results and speeding up the training. However, in the traditional machine learning approach, the participation of a human expert is usually limited to preparing the training task (e.g., labeling). This is because large datasets are typically very expensive to create and require the work of many annotators.

In the case of this project, the dataset is relatively small, and the data is very domain-specific. The expert who performed the initial labeling is also involved in the project, so it is possible to introduce intermediate expert evaluation in some form.

## 2.2 Machine learning: classification

Classification is a machine learning problem where the goal is to identify a new observation as a part of one of the predefined categories. The prediction is based on the training dataset of observations for which the categories are known. Classification problems can be solved by a large variety of algorithms ranging from decision trees to neural networks.

The task of this thesis project can be approached as a binary classification problem with two categories: time-frames with ship wakes and without them. Then a successfully trained predictive model should be able to identify wakes when given a previously unseen set of observations.

The formulation of the main thesis task in terms of classification allows the use of a range of Machine Learning classification algorithms. The nature of the available data, however, suggests using models based on CNN. CNNs are known to be effective in image recognition and finding patterns. If the acoustic data is treated in the form of time frames, the task maps directly to pattern recognition, where the target pattern is the trace of bubbles behind the ship.

### 2.2.1 Convolutional neural network

Neural networks are a class of machine learning models that consist of multiple interconnected nodes also called neurons. Nodes perform simple processing of incoming information and pass it further down the network. Nodes in neural networks are commonly organized in layers, which can be connected in various ways. Neural Networks are trained using backpropagation, which calculates the gradient of the loss function with respect to the weights of the model for each input-output and updates the weights accordingly.

The most common type of neural network, used in image analysis is CNN. The main feature of CNNs is convolutional layers. Each neuron in such a layer processes only the information from the receptive field corresponding to this neuron. It can be perceived as a sliding window moving over a 2-dimensional image and checking if certain patterns of the size of the window are present. Typical CNN architecture involves stacking several convolutional layers with other types of layers, such as pooling (aggregating information from neighboring neurons) and fully-connected layers. One example of a well-known and successful deep neural network for image recognition is VGG [11].

### 2.2.2 Residual neural network

A more recent CNN architecture which is often used as a benchmark is Residual Neural Network or ResNet [7]. ResNet solves the vanishing gradient problem, which affects the training of very large neural networks with gradient-based methods. In large networks, the gradient can become vanishingly small, and the weights can stop changing at all during training. To address this, ResNet introduces skip-connections. In simpler architectures, layers are always connected sequentially, while in ResNet some layers can be skipped during the first stages of training. For instance, if there are three layers in the network: $A$, $B$, $C$, and they are connected in order $A$ to $B$ to $C$, then the direct connection from $A$ allows to skip $B$ during initial training (Figure 2.1). This architecture not only allows addresses the vanishing gradient problem but also greatly improves training time. Since variations of ResNet are close to state-of-art results, it makes it a good architecture to apply to pattern recognition tasks such as the one in this thesis project.

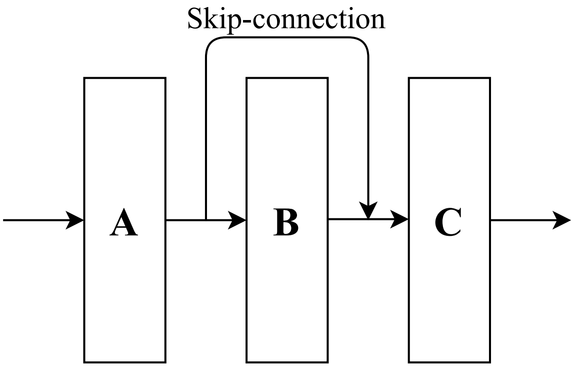

**Figure 2.1:** An example of a skipped layer in a Residual Neural Network. There is an additional connection between layers $A$ and $C$ that allows skipping layer $B$ when propagating.

## 2.3 Approaches for scarce data

One of the main challenges of this thesis project is the lack of labeled data and, more specifically, the lack of labeled positive wake examples. Typically, to perform supervised machine learning, algorithms require a considerable number of labeled samples. This is especially important for a neural network model, as they include many parameters to adjust. There are several possible approaches to address the lack of data.

### 2.3.1 Data augmentation

Data augmentation is a common strategy when the training dataset is not large enough, and more samples are needed. To augment a dataset, original samples are changed in a minor way while keeping the designated output value. Even simple augmentation is known to improve the performance of machine learning models. For example, image datasets can be augmented with rotated, cropped, or mirrored samples. Data augmentation is also a step that can be overseen by the domain expert. If generated of modified samples have to be approved as plausible by the expert, it reduces the chance of introducing errors during the augmentation.

The simple rotation type of augmentation might not be applicable to stationary profiler data, though, primarily because wakes have a fixed position relative at the top of the frame. There are, however, more complex approaches, which can also serve as a means of regularization and increasing robustness.

#### 2.3.1.1 Probabilistic models

New samples can be generated using statistical methods. They preserve the patterns in the original samples while adding diversity to the dataset.

A Gaussian Mixture Model (GMM) is a simple probabilistic model that fits the data to a convex combination of several Gaussian distributions. It can be an efficient sample generation or clustering tool when the data structure is not very complex. The optimal number of components can be determined using relative model quality estimators, such as Akaike Information Criterion (AIC). AIC penalizes a large number of parameters to fit and rewards the goodness of fit using the likelihood function, thus, finding a balanced number of distributions.

GMMs are not very efficient when applied to high-dimensional data such as images, so in order to apply them to acoustic scanner data, some dimensionality reduction is needed. Compression should also be reversible, so the generated samples can be restored.

Principal Component Analysis (PCA) is a method that transforms the data into a new coordinate system. The greatest sample variance in the data lies on the first component, the second-largest lies on the second component, and so on. The ratio of explained variance can regulate the number of dimensions after compression.

While PCA is efficient, it does not specifically try to preserve the patterns existing in the data. This can be addressed by applying a more meaningful transformation first. CNNs perform well in preserving image patterns, and they can also be used for efficient compression within an autoencoder. Autoencoder is a special type of unsupervised neural network that consists of two components: an encoder, which compresses the input and a decoder that reverts the process (Figure 2.2). The training process of an autoencoder consists of attempting to reconstruct a set of samples. A trained CNN-based autoencoder can create a compressed representation of image-like data while preserving patterns such as the shape of bubble traces felt by ships.

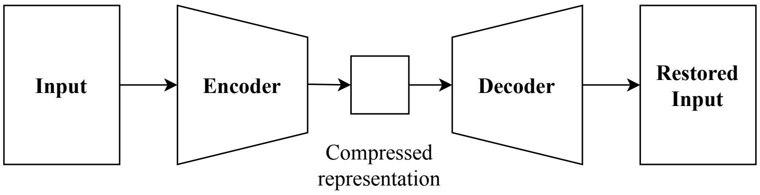

**Figure 2.2:** The architecture of autoencoder.

#### 2.3.1.2 Generative adversarial networks

The use of neural networks allows a more sophisticated approach to data generation with Generative Adversarial Networks (GANs) [6]. GANs can produce high-quality artificial samples. Their training involves two networks competing: the generative network creates new samples and the discriminative evaluates them. Applying GANs shows improvement over standard augmentation techniques on tasks similar to wake detection [5]. However, the main drawback of using this type of networks is that they require a much larger number of samples to be trained with. This means that until more labeled wake-related data is retrieved, GANs are not applicable.

### 2.3.2 Transfer learning

Transfer learning is an approach that uses information learned while solving a problem in one domain in solving a similar problem in a different domain. One example of a popular source model is VGG [11], which is trained on ImageNet [4] — a very large dataset with 1,000 categories of images. It can be tuned to perform a variety of similar image recognition tasks even outside the 1,000 categories. Transfer learning is typically used when large amounts of data are impossible to process or unobtainable, which is the case of this project.

Transfer learning is known to perform well on image data in general and also shows good results in image segmentation of scanned data [12]. The main difficulty is finding a source dataset which is close enough to the target acoustic sonar data to make transfer possible. During the work on this thesis, several marine sets of data were considered, but ultimately it was decided that the format is too different. The reasons for this include:

- Most acoustic datasets, even the ones collected with similar devices, output the observations with much lower resolution than in the target data.
- Acoustic scanners often target only the upper layers of the sea down to the depth of several meters. The research project, this thesis is a part of, investigates the role shipping plays in water mixing, and the zone of interest might be as deep as 25–30 meters.
- The scanners set to monitor currents and sea fauna are usually mounted outside the shipping lanes to avoid interference, so their data does not contain any wakes or similar signatures to detect.
- Depending on the tasks, profilers can collect data in burst mode, meaning they perform many scans in a short time frame and then go into sleep mode. These time frames are not large enough to capture wakes and other larger patterns.

### 2.3.3 Unbalanced training

Classification algorithms typically require a balanced set of training data, even if target datasets are heavily unbalanced. This is because loss function is computed over the whole dataset, and models tend to ignore underrepresented categories. If having a balanced set is impossible, weights, over- and undersampling, data augmentation, and specific architectures are applied, though their efficiency can be limited.

While the general lack of data is a problem in this project, an even bigger issue is the lack of labeled wakes. A somewhat reasonable number of diverse negative samples can be selected from the month of observations. The number of wakes is very small even compared to this set, though. To some extent, it can be addressed by augmentation methods described above. However, their potential for generation is limited by a very small size of the original set of wakes. Another approach to improve performance without generating more wake samples or performing transfer is modifying the learning algorithm.

#### 2.3.3.1 Reweighting examples

One strategy of addressing unbalanced data is identifying more or less valuable samples within the training dataset and assigning corresponding weights to them. Typically these weights are initialized offline, once during the training process, but there is an alternative approach proposed by Ren et al. [9]. It involves using a small subset of perfectly labeled clean data as a hyper-validation set. At each step of training, the gradient of the hyper-validation loss with respect to the weights of samples in the current mini-batch is computed and the weights are updated accordingly. The reweighting model can be implemented for most deep learning architectures and showed good performance both for unbalanced datasets and datasets with noisy labels.

Perhaps the most important feature that makes this method applicable to the wake detection problem is that the size of the hyper-validation set can be as small as 10 samples while keeping a high level of performance. It also aligns well with the expert-in-the-loop framework. Since only a small number of clean examples are needed, they can be hand-picked or approved by the expert.

## 2.4 Noise robustness

Sea data, due to its origin, can be very noisy. The noise comes in different forms. The seemingly uniformly distributed noise is produced by plankton that comes closer to the surface at night and in the morning (Figure 2.3). Wind and waves produce more structured noise that sometimes can be similar to or cover completely traces of passing vessels.

Since the quantity of data is relatively small, there is a high chance that a deep learning model can overfit to this noise.

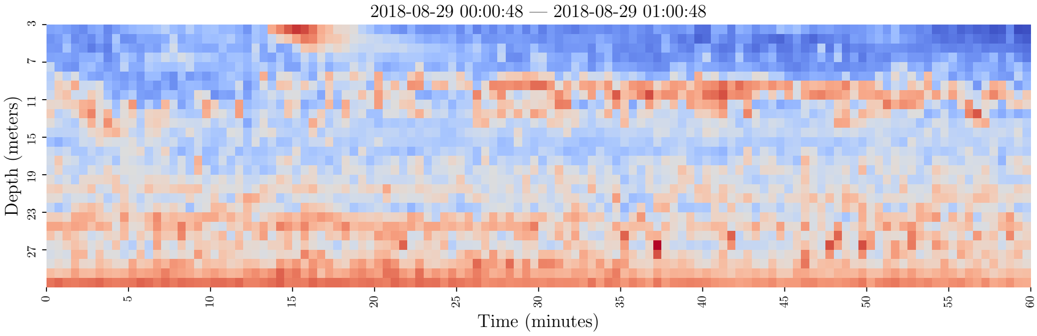

**Figure 2.3:** One hour (00:00 to 03:00 on the 28th of August 2018) of observations collected by the central beam plotted. The data is noisy because of the zooplankton that moves to the surface at night. A ship wake is visible at the timestamp of 00:13

### 2.4.1 Unstructured noise robustness

Since the noise from plankton resembles random noise, it might prove beneficial to improve the model's robustness to random noise. It can be done in several ways. For example, if the dataset was larger, it would have been feasible to augment it with noisy samples so the model can train on them.

#### 2.4.1.1 Randomized smoothing

Smoothing with noise insertions is a method used to increase adversarial robustness. It can guarantee robustness to random noise that changes the sample within a certain small neighborhood [2, 10].

The base version of the method [2] involves making additional evaluations for each new incoming sample. A sample is augmented with Gaussian noise to create noisy samples. A pretrained classifier makes predictions for each of the noisy samples, and the most commonly predicted class is selected as a prediction for the original input (Figure 2.4. Depending on parameters, the algorithm can also abstain if two or more classes show similar support.

The algorithm needs the base classifier to be trained on data with Gaussian noise in it and be able to classify it mostly correctly. Under the assumption that sea noise can be considered Gaussian, this applies to the wake dataset and classifier models trained on it.

### 2.4.2 Label noise robustness

Manual labeling can also produce noise in the labels. For the dataset used in this project, the number of ships passing close to the mounted scanner was not very large, and the time frame is limited to one month. With very few potential wakes to look for, it can be assumed that the labeling is perfect or near-perfect. However, even some of these wakes are marked as ambiguous or unclear by the expert. Further experiments are also planned, and they can be conducted in the proximity of a busier shipping lane for a longer period of time. For the larger body of data, the probability of having noisy labels is higher. It means that if a new set of observations

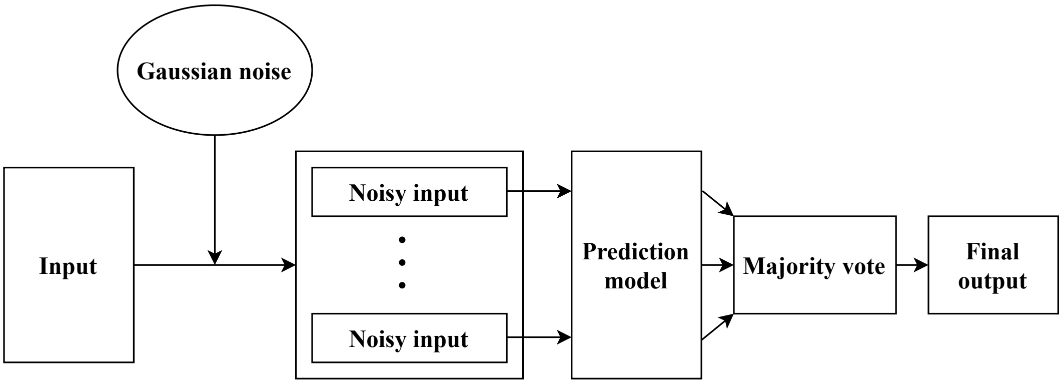

**Figure 2.4:** Randomized smoothing algorithm.

is to be added, it should be labeled either by a human expert or by an algorithm, such as proposed in this thesis project. Automatic detection always leaves room for mistakes, and for an expert, it might be an unreasonable workload to go through the list of all passing ships. Thus, some level of noise in labels is to be expected, and a predictive algorithm should be able to handle it. This can be considered another reason to implement the sample reweighting algorithm [9], as it shows good performance with noisy labeling as well as with unbalanced datasets.

# 3
# Methodology

In this chapter, a brief overview of the model development is given, which is followed by the description of the original dataset and data augmentation. The training process and the baseline predictive model are discussed as well. In order to improve performance on unbalanced and noisy data, two additional models are implemented: the example reweighting model and the randomized smoothing model. The details of the evaluation process are also presented.

## 3.1 Problem formulation

The overall goal of this thesis project is to develop a machine learning-based tool that allows automatic wake detection. Having such an instrument is essential for studying available observations and conducting further research. Applying machine learning to this type of problem is fitting: automatic recognition can greatly reduce the time required to mark each wake, even if it produces some false-positive results. This can be formulated in terms of classification problem: The tool must solve a binary classification task, where the two classes can be labeled as 'wake' and 'background'. More formally, if $X = \{x_1, x_2, ..., x_n\}$ is a set of objects and $Y = \{y_1, y_2, ..., y_n\}$ is a set of classes, and $f$ is a classifier, which maps $X$ into $Y : f(x_i) = y_j$ . In case of wake recognition $X$ is the set of frames and $Y = \{\text{'}\textit{wake}\text{'}, \text{'}\textit{background}\text{'}\}$.

The main domain- and data-specific characteristics of the problem include labeled data imbalance, noise in the data, and potentially in labels. The expert responsible for the initial data labeling is involved in the project, and can, therefore, evaluate intermediate model performance.

The development of the model can be structured as follows:

- Collection of the data from the acoustic scanner;
- Initial data labeling by the domain expert;
- Data augmentation and its approval by the expert;
- Base deep learning model training;
- Base model's performance evaluation;
- Implementation of the algorithms to make the model robust to data imbalance and noise with an additional input from the expert;
- Final model evaluation.

The first two steps are out of the scope of this thesis, while the rest of the process is covered in the following sections of this paper. This simplified pipeline is plotted in Figure 3.1.

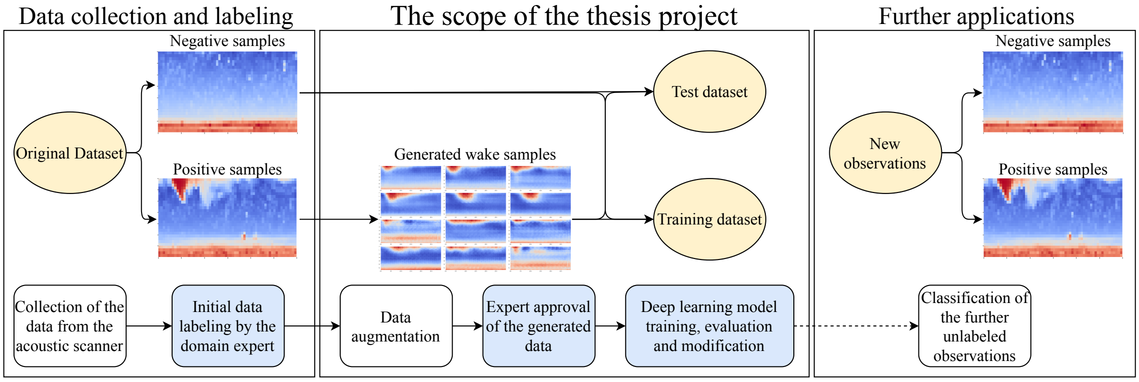
Data collection and labeling
The scope of the thesis project
Further applications
Negative samples
Original Dataset
Positive samples
Collection of the data from the acoustic scanner
Initial data labeling by the domain expert
Generated wake samples
Test dataset
Training dataset
Data augmentation
Expert approval of the generated data
Deep learning model training, evaluation and modification
New observations
Negative samples
Positive samples
Classification of the further unlabeled observations

**Figure 3.1:** Pipeline of the project showing all major steps from data collection to the final model evaluation. Potential further applications are also included in the diagram. Steps shaded in blue indicate the involvement of the domain expert.

## 3.2 Data description and preparation

The data for this project is collected by an Acoustic Current Doppler Profiler mounted on a buoy located approximately 30 meters deep. The profiler takes echosounder measurements of the waters above with an interval of 1 meter. It uses five beams: four are slanted at a 25° angle, and the fifth one is vertical. Because of the wave interference, the data from the upper 3-meter layer is unreliable. The dataset covering 4 weeks (28 August to 25 September 2018) is available. It comes in the form of 5 time series, 1 from each beam, with observations for 79,023 timestamps in total. For each timestamp, 28 data points are given, corresponding to depth levels from 3.5 meters to 30.5 meters with 1-meter interval. One of the slanted beams (beam 2) was malfunctioning, so its data is corrupted. The scanner records data every 30 seconds. Signal data was normalized, so all the values fall between 0 and 1. The nighttime data contains very high levels of noise, and several nighttime wakes are marked as ambiguous by the expert. Because of this, for training and testing purposes daytime negative samples are used.

### 3.2.1 Data representation and visualization

This type of data can be visualized in the form of an echogram, such as in Figures 1.1 and 2.3. The echograms display the intensity of the reflected signal; Wakes and other objects are visually recognizable in this representation. 165 wakes in total are identified by an expert and marked on echograms. Publicly available ship tracking data was checked during initial labeling, so it is confirmed that all the wakes that had a recognizable trace of bubbles were detected. This allows the assumption that all the data outside the marked wake time frames can be used as negative samples.

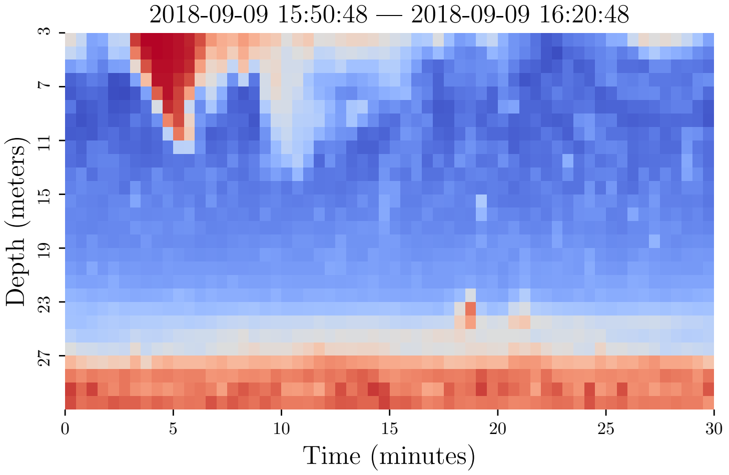

**Figure 3.2:** A 30-minute frame of observations starting around 15:50 on the 9th of September 2018. It displays a clearly visible wake in the top-left corner.

In order to perform binary classification, the echogram is split into fixed-size frames. The length of the time frame is 60 data points or 30 minutes, and the size of one such sample is 4 × 28 × 60 datapoints. The first dimension corresponds to the number of functional beams. One example of such a frame is plotted in Figure 3.2. The wakes in the dataset range from large to barely noticeable, as can be seen in Figure 3.3.

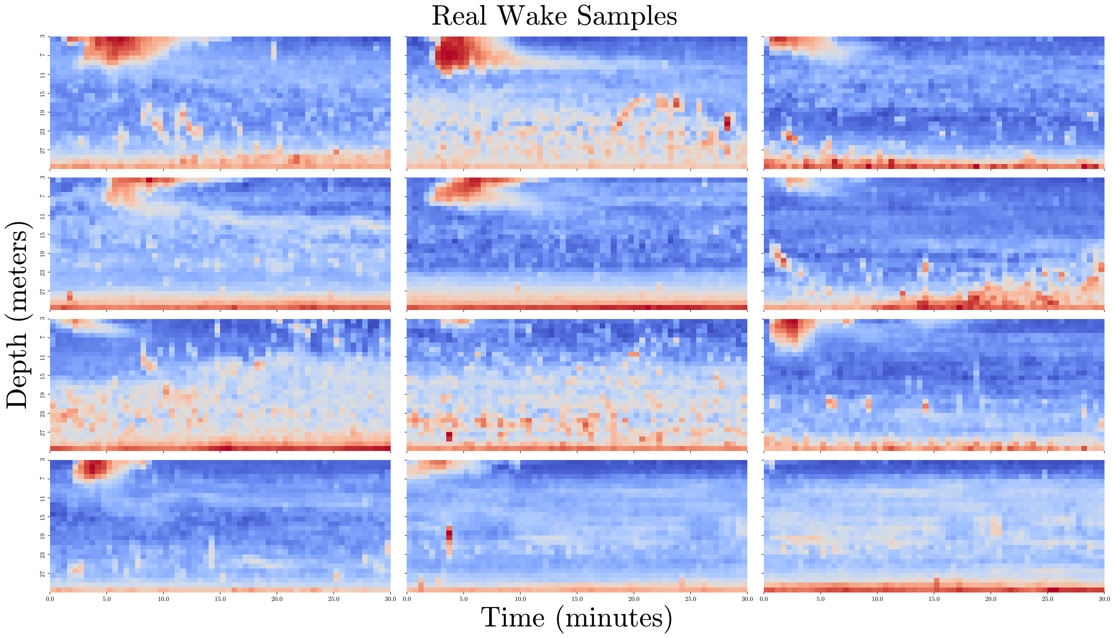

**Figure 3.3:** Examples of wakes labeled by the expert.

All of the plotted examples in this paper use the data from the central beam (beam 5) because it consistently shows stronger signals. The difference was confirmed by comparing Wasserstein distances between the data from different beams and an “empty” frame with a flat distribution of signals. Wasserstein distance computes the cost of transforming one distribution into another, where the cost is the size of the part that has to be changed. The Wasserstein distance from the flat distribution is consistently larger for the central beam than for the other three.

### 3.2.2 Set-aside dataset

Before proceeding with any experiments, a subset covering four days (22 September to 25 September 2018) was set aside for separate evaluation. The period includes 23 wakes of varying sizes and notably worse weather conditions, according to the expert. This subset is selected as a more difficult and completely independent additional evaluation task for the model. The remaining dataset containing 142 labeled wakes is used in the development of the predictive model.

## 3.3 Data augmentation

In order to successfully perform deep learning training, a sufficient number of positively and negatively labeled data samples is needed. The ship wake dataset is heavily unbalanced and noisy, so data augmentation is necessary to generate more wake samples and allow the use of neural networks.

Data augmentation can only be based on the labeled 142 wake frames. This number is clearly not large enough to apply deep learning models, such as generative adversarial networks (GANs) [5], so a more robust approach of fitting a Gaussian mixture model (GMM) is chosen.

### 3.3.1 Data compression

Since one frame containing a wake has relatively high dimensionality, a reduction has to be applied before fitting a GMM. To make the dimensionality reduction more meaningful and preserve the patterns of the wakes, a CNN-based approach is taken.

The first step of data compression was, therefore, training an autoencoder on all 142 positively labeled frames with wakes. A simple autoencoder model with three convolutional layers in both encoder and decoder was trained on these examples for 70 epochs. The autoencoder then was able to compress and restore frames while keeping the major patterns and canceling most of the noise. In Figure 3.4 the frame from Figure 3.2 is shown after passing the autoencoder. Most of the noise was removed, but the wake is preserved and clearly visible. After the training, the encoder is used to compress the same 142 samples.

After the frames are passed through the network, the dimensionality remains relatively high, and as the second compression step, PCA was applied. The model was set to keep 99.9% of the variance, and it reduced the sizes to only 48 dimensions each.

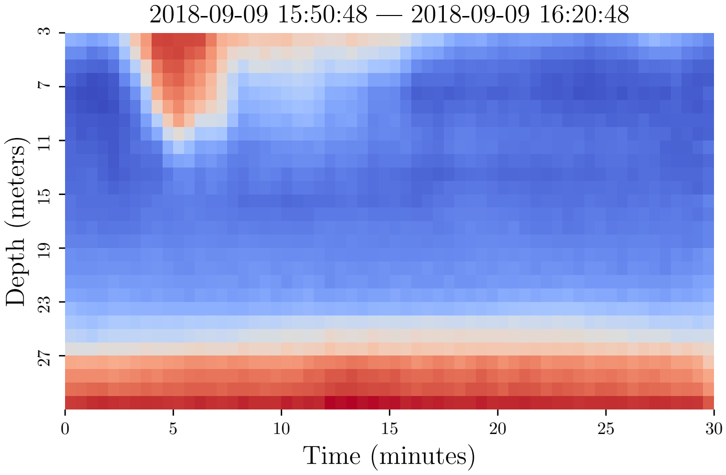

**Figure 3.4:** The frame plotted in Figure 3.2 after being passed through the trained autoencoder.

### 3.3.2 Sample generation

GMM model was fitted to the set of 142 compressed examples. The number of Gaussian components was set to 21 after computing Akaike Information Criterion for fitted GMMs with 1 to 60 components.

The GMM model allows the generation of new samples according to the learned distributions. For future experiments, a total of 1,000 48-dimensional samples were generated. The samples were passed through the inverse PCA model and then through the decoder, restoring them to the $4 \times 28 \times 60$ size.

Some of the generated wakes are shown in Figure 3.5. Synthetic data has noticeably less noise and much smoother than real samples (Figure 3.3), but all the defining features have been successfully preserved. The generated set was approved by the expert as containing plausible shapes that could only be wakes in real data. The expert agreed that while being smoother than the wakes in real observations, most of the generated data qualifies as a set of positive examples. As an additional step to confirm the quality of augmentation, pairwise Wasserstein distances were computed between a set of real wakes and a set of generated wakes. The average value was noticeably smaller than the average pairwise distance between the real wake frames and a random subset of frames.

## 3.4 Deep learning models

Four deep learning model were implemented and tested as a part of this thesis. The base model uses ResNet architecture. Since the structure of data is not very complex, and the resolution is small, a very deep network is not necessary. So a relatively small ResNet18 model was implemented. ResNet18 consists of 18 layers in total: 1 convolutional layer followed by 8 residual blocks with 2 convolutional layers each, and a fully connected output layer with sigmoid activation. The model treats the data from the four beams as four channels of the same image. A single input,

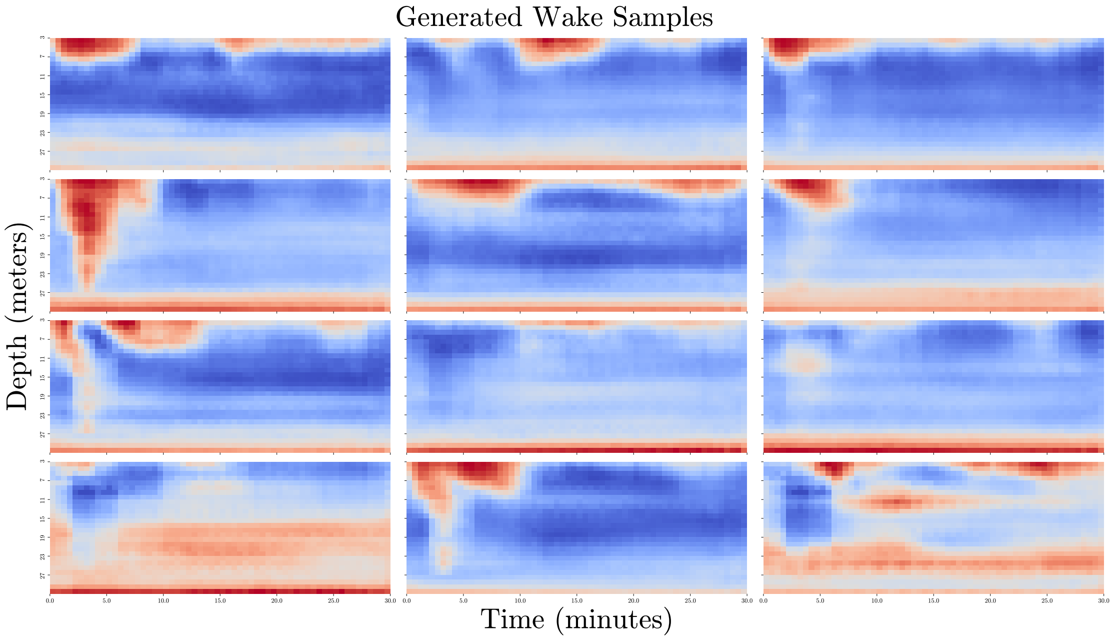

**Figure 3.5:** Wake frames generated using GMM and autoencoder.

therefore, has the shape of $4 \times 28 \times 60$. The output consists of two values, which can be interpreted as probabilities of the input belonging to 'wake' and 'background' classes.

For the second model, exactly the same ResNet18 architecture was reimplemented to allow computing gradients with respect to the weights of examples. The hyper-validation set is set to the minimal size of 10: 5 positive and 5 negative examples. The negative samples were hand-picked from the dataset by the expert as clean with confidence. The purpose of this model is to test its performance when the training data is unbalanced compared to the baseline.

The third model implements the randomized smoothing algorithm using the default ResNet as a default classifier. This model should show an improvement compared to the baseline if the hypothesis that the noise in the data is Gaussian is correct.

The fourth and the final model combines the two approaches and uses the example reweighting model as a default classifier.

## 3.5 Evaluation metrics

The evaluation of the models is based on the standard metrics for binary classification. Accuracy (the ratio of correct guesses) shows general performance well on balanced datasets.

Area Under Receiver Operating Characteristic Curve (AUC ROC) is a metric that measures how well the model distinguishes between classes with different thresholds to make a decision. Randomized smoothing uses voting between predictions for noisy variations of the same sample. Therefore, it is hard to calculate aggregated AUC ROC, so it is computed only for the baseline classifier and the example reweighting model.

False Negative Rate (FNR) shows the ratio wakes missed by the algorithm. Since the purpose of the model is to help to detect wakes in the datasets, this is probably the most meaningful metric.

In order to achieve stability in results, each experiment is performed 10 times. For each metric, the average and the standard deviation are calculated.

For all experiments, the test dataset includes 150 samples, 29 of which are randomly selected real wake frames, 46 are randomly selected generated wake frames, and 75 are randomly subsampled frames without wakes.

An additional experiment is performed using the imbalanced set-aside dataset covering four days with 23 known wakes. It is passed to the model sequentially in the form of overlapping frames. The purpose of this test is to show the performance with a completely independent set of observations.

# 4

# Experiments

This chapter presents the experimental results. The performance of three main models is evaluated: the baseline ResNet18 model, the example reweighting model, the randomized smoothing model. In all experiments, the main metrics are the accuracy score and FNR, since the test dataset is always balanced. Additionally, for the baseline experiment, the AUC ROC score is computed. Each experiment is performed 10 times to achieve stability in the results, and average values are used. Full result tables are provided in the appendix section. As an additional evaluation, the performance of the main model is shown on the set-aside subset of four days.

## 4.1 Baseline ResNet model

In order to perform the baseline experiment, the ResNet18 model is trained on a balanced dataset that includes: 500 negative samples, 113 positive samples from the real data, and 387 generated positive samples. The test dataset is also balanced and includes 75 randomly selected negative samples, 29 positive samples from the real data, and 46 generated positive samples. All samples are selected randomly from their respective sets.

| Training set ratio | Accuracy | AUC ROC | FNR |
|---|---|---|---|
| 500 positive / 500 negative samples | 93.40 ± 1.80 % | 0.97 ± 0.01 | 9.87 ± 3.28 % |

**Table 4.1:** The baseline experiment with the ResNet18 model. The test set contains 75 positive (29 real + 46 generated) and 75 negative samples. The results are the mean average over 10 experiments ± standard deviation.

The results shown in Table 4.1 can be interpreted in the following way: 93.4% of predictions are correct on average, but most of the wrong predictions come from false negatives. Roughly 10% of wakes The model has a high AUC ROC score, meaning it performs well in ranking samples by the likelihood of containing wakes.

## 4.2 Example reweighting model

The second evaluated model is the example reweighting model. Its main purpose is to allow training on unbalanced data while keeping test performance from quickly degrading. The reweighting model has exactly the same architecture as the baseline model but takes an additional step in training to adjust the example weights during training.

The experiment includes training the models on five training datasets with different class balance: 50%, 20%, 10%, 5%, and 2.5% of positive samples. The total size of the training dataset is the same as in the previous experiment — 1000 samples. The test dataset is kept balanced with 75 positive and 75 negative samples. The results are shown in Figure 4.1. With the decrease of positive samples in the training data, the quality of predictions of both models predictably drops. The example reweighting models shows noticeably better robustness to class imbalance.

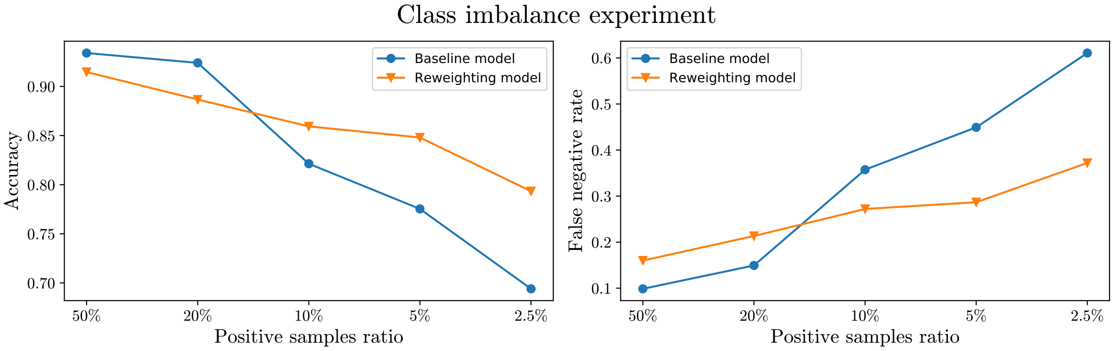

**Figure 4.1:** Class imbalance experiment with the ResNet18 model. Accuracy and false negative rate scores on the test dataset are plotted for the baseline and the reweighting model. The training sets have five different ratios of positive samples from 50% to 2.5%. The test set contains 75 positive (29 real + 46 generated) and 75 negative samples. The results are the mean average over 10 experiments.

## 4.3 Randomized smoothing model

The purpose of the randomized smoothing model is to improve robustness to Gaussian noise in the data. In order to test the model, the noise was introduced into the test samples of the same structure as in the previous experiments. The intensity of the noise is regulated by the standard deviation parameter $\sigma$. There are two standard deviation parameters to be adjusted: for the smoothing process and the test data corruption. First, the smoothing $\sigma$ values for the experiments were selected by running the smoothing model with clean test data. The performance is shown in Table 4.2. It drops noticeably when smoothing is applied. Three benchmark $\sigma$ values are selected for further experiments: 0.1, 0.05 and 0.03.

| Smoothing $\sigma$ | Test accuracy |
|---|---|
| $\sigma = 0.300$ | 55.38 ± 2.15% |
| $\sigma = 0.250$ | 59.43 ± 2.85% |
| $\sigma = 0.200$ | 65.86 ± 3.44% |
| $\sigma = 0.150$ | 72.62 ± 3.73% |
| $\sigma = 0.100$ | 79.67 ± 3.22% |
| $\sigma = 0.085$ | 81.29 ± 3.30% |
| $\sigma = 0.065$ | 85.57 ± 2.54% |
| $\sigma = 0.050$ | 89.71 ± 2.02% |
| $\sigma = 0.030$ | 93.10 ± 0.85% |
| $\sigma = 0.010$ | 94.14 ± 1.44% |
| $\sigma = 0.000$ | 93.40 ± 1.80% |

**Table 4.2:** The experiment with the randomized smoothing model to select the appropriate standard deviation. No additional noise added to the test data. The classifier makes 500 predictions in the neighborhood of each sample to make a decision. The dataset structure is the same as in experiment 1 (Table 5.1) Highlighted rows indicate smoothing $\sigma$ selected for further experiments.

The second parameter is the intensity of noise introduced into the test data. Noise $\sigma$ was arbitrarily set to values: 0.25, 0.2, 0.15, 0.1, 0.05, 0.01, 0. For each combination of parameters (3 smoothing $\sigma$ values and no smoothing), 10 experiments were performed. The training and test dataset structure is the same as in the baseline experiment, the only difference being Gaussian noise added to the test set. The results of the experiment are shown in Figure 4.2. The results drop similarly when stronger smoothing is applied. This means that the model's robustness is not increased and, most likely, that the assumption of Gaussian noise in the sea data is wrong. Since the reweighting experiment did not show any improvements in noise robustness, the experiment with the combined randomized smoothing model was folded.

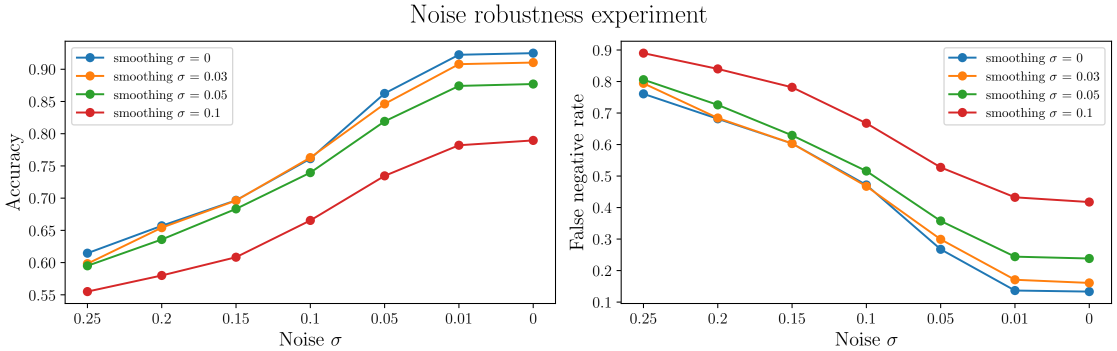

**Figure 4.2:** The noise robustness experiment with the ResNet18 model. Accuracy and false negative rate scores on the test dataset are plotted for the baseline and three randomized smoothing models for different levels of noise in the test data. The test set contains 75 positive (29 real + 46 generated) and 75 negative samples. The results are the mean average over 10 experiments.

## 4.4 Set-aside dataset experiment

As a secondary evaluation of the baseline model, an experiment is performed on the set-aside dataset. The whole dataset is passed through the ResNet18 model in the form of overlapping time frames of the $4 \times 28 \times 60$ size starting at each timestamp. Unlike the previous experiments, the whole set-aside is used, including extremely noisy nighttime data and bad weather in a part of this period. The 23rd of September was specifically marked by the expert as the one with the worst weather over the month.

There are 23 known wakes in this subset. For each wake, only the starting timestamp is known. It is assumed that the wake is visible in a 30-minute frame if the frame starts at most 25 minutes before the wake start or at most 5 minutes later. The total number of frames in this experiment is 9,780, out of which only 1,338 are expected to be classified as wakes due to overlaps.

For this setup. the mean accuracy score over 10 experiments is as low as 60.6%, while the mean false negative rate is at 38.06%, which is comparable to test performance with a significant level of noise. Overall, the results show a significant drop in performance compared to experiments on cleaner and smaller test sets.

# 5

# Discussion

In this chapter, the previously shown results are discussed in further detail. The relevance of the baseline model and its modifications is analyzed; possible reasons behind the performance are pointed out. Based on this analysis, alternative ways to improve the model are outlined.

## 5.1 ResNet model

The baseline ResNet18 model showed a consistently high level of performance for such a small dataset in terms of accuracy. It is noticeable, though, that most of the mistakes are made on positive samples, and that the false negative rate is relatively high, close to 10% on average. It means that when the model is given a new dataset, it can potentially miss 10% of wakes altogether. It is likely the model is experiencing difficulties in learning the differences between wakes and structured noise as sometimes they are nearly indistinguishable even for a human expert.

Even though human abilities to classify are limited, the participation of the expert turned out to be beneficial tor this project. It allowed to 'certify' data augmentation, and made it possible to make an educated choice of hyper-validation set for the example reweighting experiment.

The most apparent weakness of the model is its ability to handle noise. Both the experiment with added Gaussian noise and the experiment with a more noisy dataset from the last four days show a significant performance drop of the baseline model. It is possible that the problem is related to the small sizes of train datasets, and the model becomes sensitive to the smallest changes.

As shown by the class imbalance experiment, the default model also requires a relatively balanced training set. This limitation means that, for the unbalanced wake data, a large portion of negative samples cannot be used in training at all. Alternatively, a substantial augmentation is required, as it was done in this project. Augmentation has its own downsides, though, such as potential overfitting due to reusing the information from the same samples.

## 5.2 Example reweighting

The purpose of the example reweighting model is to allow training on unbalanced data, thus allowing using as much collected information as possible. Because it uses the hyper-validation set at every step of training to preserve robustness, it loses in performance to the baseline model on balanced and slightly disproportional training sets. However, when trained with only 10% of positive samples, accuracy for the baseline drops while the number of false negatives goes up. With the same ratio, the example reweighing model retains better performance closer to the values for the balanced set.

These figures are encouraging as they mean that using the reweighting technique model can make use of most of the unbalanced data. The improvements from adding a large number of negative samples to the test set can potentially compensate for the reduced ability to predict and lead to overall performance gains.

## 5.3 Randomized smoothing

The randomized smoothing model was outperformed by the baseline model in almost all the experiments. The likely explanation is that the assumption that the wake dataset already contains Gaussian noise is incorrect. Then smoothing with Gaussian cannot possibly lead to any improvements. The combined experiment with randomized smoothing and example reweighting in one model showed similar results and was not included in the main results section.

## 5.4 Potential improvements

The results achieved in this paper can potentially be improved directly by training the model on an unbalanced dataset while utilizing as many samples as possible, as was suggested above. There are, however, other methods not investigated in this paper that might lead to improvements. Firstly, transfer learning remains a viable option as long a suitable source dataset is available, even though it was not possible to access one within the time frame of this project. Secondly, noise in the data remains an issue as shown by the set-aside experiment. Randomized smoothing did not lead to improvements, but other means might improve robustness of the model even with the small dataset. For instance, certain types of structured noise (fish schools, waves, etc.) can be assigned to their own class. This way the model can learn too better distinguish them from wakes. To use this method a thorough data preparation would be needed as well as even greater expert involvement.

# 6 Conclusion

The studies on humanity's impact on the ocean are crucial for preserving marine life and resources. An important aspect of marine science is data collection and processing. Implementing advanced analytical methods, such as machine learning, on this stage of research, can be highly beneficial. In this thesis project, a deep learning-based tool was developed for ship wake detection in the acoustic scanner data.

The baseline model performed adequately well as can be seen from the test results. However, the experiments exposed several weaknesses. As can be expected of a deep learning model, it does not display high accuracy when trained on a small and unbalanced dataset. In order to address this problem, the dataset was populated with the generated samples. In addition, the example reweighting technique was applied. While decreasing overall performance, it displayed better robustness to class imbalance, which can potentially allow using more information in training. The second and potentially more serious issue is the high level of noise in the marine data. The randomized smoothing technique was implemented to enhance noise robustness, but the experiment did not show any improvements over the baseline model. Even though this particular weakness remains, the results demonstrate that machine learning solutions are viable for these types of tasks. In order to process noisy data, a different approach is needed, and it could be a possible direction for further work within this project.

Potential applications of the developed model are not limited to finding wakes. It detects patterns in the acoustic data, and patterns can represent other objects, such as fish schools, sea mammals, or water mixing zones. Studying acoustic data with machine learning tools can provide insights into underwater processes other than ship-induced water mixing, that is the subject of the project this thesis is part of. More generalized recognition of the underwater objects of a different nature can, therefore, become a promising research topic.

# Bibliography

[1] Olav Brautaset, Anders Ueland Waldeland, Espen Johnsen, Ketil Malde, Line Eikvil, Arnt-Børre Salberg, and Nils Olav Handegard. Acoustic classification in multifrequency echosounder data using deep convolutional neural networks. *ICES Journal of Marine Science*, 2020.

[2] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[3] Anthony C Copeland, Gopalan Ravichandran, and Mohan M Trivedi. Localized radon transform-based detection of ship wakes in sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 33(1):35–45, 1995.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] R Peltzer, W Garrett, and P Smith. A remote sensing study of a surface ship wake. In *OCEANS'85-Ocean Engineering and the Environment*, pages 277–286. IEEE, 1985.

[9] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

[10] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *arXiv preprint arXiv:1906.04584*, 2019.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] Annegreet Van Opbroek, M Arfan Ikram, Meike W Vernooij, and Marleen De Bruijne. Transfer learning improves supervised image segmentation across imaging protocols. *IEEE transactions on medical imaging*, 34(5):1018–1030, 2014.

# A

# Appendix

| Training set ratio | Accuracy | AUC ROC | FNR |
|---|---|---|---|
| 500 positive / 500 negative samples | 93.40 ± 1.80 % | 0.97 ± 0.01 | 9.87 ± 3.28 % |
| 200 positive / 800 negative samples | 92.40 ± 2.65 % | 0.96 ± 0.02 | 14.93 ± 5.33 % |
| 100 positive / 900 negative samples | 82.13 ± 1.33 % | 0.94 ± 0.02 | 35.73 ± 2.65 % |
| 50 positive / 950 negative samples | 77.53 ± 4.56 % | 0.93 ± 0.03 | 44.93 ± 9.12 % |
| 25 positive / 975 negative samples | 69.40 ± 2.64 % | 0.87 ± 0.04 | 61.07 ± 5.16 % |

**Table A.1:** Class imbalance experiment with the default model. The training sets have five different ratios of positive samples from 50% to 2.5%.The test set contains 75 positive (29 real + 46 generated) and 75 negative samples. The results are the mean average over 10 experiments ± standard deviation.

| Training set ratio | Accuracy | AUC ROC | FNR |
|---|---|---|---|
| 500 positive / 500 negative samples | 91.47 ± 2.36 % | 0.97 ± 0.01 | 16.00 ± 5.06 % |
| 200 positive / 800 negative samples | 88.67 ± 2.70 % | 0.96 ± 0.02 | 21.33 ± 6.08 % |
| 100 positive / 900 negative samples | 85.93 ± 4.03 % | 0.94 ± 0.02 | 27.20 ± 8.05 % |
| 50 positive / 950 negative samples | 84.80 ± 3.84 % | 0.92 ± 0.02 | 28.67 ± 8.67 % |
| 25 positive / 975 negative samples | 79.33 ± 6.42 % | 0.84 ± 0.08 | 37.20 ± 14.85 % |

**Table A.2:** Class imbalance experiment with the reweighting model. The training sets have five different ratios of positive samples from 50% to 2.5%.The test set contains 75 positive (29 real + 46 generated) and 75 negative samples. The results are the mean average over 10 experiments ± standard deviation.

| | Randomized smoothing model test results | | | |
|---|---|---|---|---|
| | Smoothing $\sigma = 0.1$ | | Smoothing $\sigma = 0.05$ | |
| Noise $\sigma$ | Accuracy | FNR | Accuracy | FNR |
| $\sigma = 0.25$ | 55.50 ± 3.65 % | 89.00 ± 7.30 % | 59.50 ± 4.07 % | 80.58 ± 8.51 % |
| $\sigma = 0.2$ | 58.00 ± 4.04 % | 84.00 ± 8.08 % | 63.58 ± 4.29 % | 72.58 ± 8.55 % |
| $\sigma = 0.15$ | 60.83 ± 4.19 % | 78.17 ± 8.52 % | 68.33 ± 4.51 % | 62.92 ± 9.05 % |
| $\sigma = 0.1$ | 66.54 ± 4.21 % | 66.75 ± 8.56 % | 73.96 ± 3.14 % | 51.58 ± 5.96 % |
| $\sigma = 0.05$ | 73.46 ± 3.56 % | 52.75 ± 7.36 % | 81.92 ± 2.63 % | 35.75 ± 5.38 % |
| $\sigma = 0.01$ | 78.21 ± 2.39 % | 43.25 ± 5.18 % | 87.42 ± 3.29 % | 24.42 ± 6.35 % |
| $\sigma = 0$ | 78.96 ± 2.58 % | 41.75 ± 5.51 % | 87.71 ± 3.55 % | 23.83 ± 6.81 % |

| | Smoothing $\sigma = 0.03$ | | No Smoothing | |
|---|---|---|---|---|
| Noise $\sigma$ | Accuracy | FNR | Accuracy | FNR |
| $\sigma = 0.25$ | 59.83 ± 4.40 % | 79.42 ± 9.21 % | 61.46 ± 5.44 % | 76.08 ± 11.49 % |
| $\sigma = 0.2$ | 65.42 ± 4.22 % | 68.42 ± 8.66 % | 65.71 ± 4.90 % | 68.17 ± 10.02 % |
| $\sigma = 0.15$ | 69.62 ± 4.31 % | 60.33 ± 8.94 % | 69.67 ± 4.91 % | 60.33 ± 9.92 % |
| $\sigma = 0.1$ | 76.29 ± 2.99 % | 46.75 ± 6.28 % | 76.12 ± 3.79 % | 47.17 ± 7.83 % |
| $\sigma = 0.05$ | 84.62 ± 2.45 % | 29.92 ± 4.76 % | 86.25 ± 2.37 % | 26.75 ± 4.95 % |
| $\sigma = 0.01$ | 90.79 ± 3.16 % | 17.08 ± 5.89 % | 92.25 ± 1.75 % | 13.67 ± 3.93 % |
| $\sigma = 0$ | 91.04 ± 2.60 % | 16.08 ± 5.11 % | 92.50 ± 1.85 % | 13.33 ± 3.89 % |

**Table A.3:** The noise robustness experiment with the ResNet18 model. Accuracy and false negative rate scores on the test dataset are plotted for the baseline and three randomized smoothing models for different levels of noise in the test data. The test set contains 75 positive (29 real + 46 generated) and 75 negative samples. The results are the mean average over 10 experiments.)